PDP11/23

DCF11-AA DIAG
CJKDBB0

AH-F140B-MC

FICHE 1 OF 2

SEP 1979
digital
MADE IN USA

PDP11/23

DCF11-AA DIAG

CJKDBB0

AH-F140B-MC

FICHE 2 OF 2

SEP 1979

digital

MADE IN USA

```
000000                          .REPT 0
```

IDENTIFICATION

PRODUCT CODE: AC-F141B-MC

PRODUCT NAME: CJKDBB0 DCF11-AA CPU DIAG

DATE: JUN-79

MAINTAINER: DIAGNOSTIC ENGINEERING

CONTENTS

1.0 GENERAL INFORMATION

1.1 HISTORY:

THIS PROGRAM IS A COMBINED VERSION OF THE THREE BASIC
11/34 DIAGNOSTIC PROGRAMS WITH MODIFICATIONS AND
ENHANCEMENTS MADE TO ACCOUNT FOR THE DIFFERENCES
BETWEEN THE TWO PROCESSORS.

1.2 PROGRAM DESCRIPTION:

THIS PROGRAM CONTAINS THREE PARTS: CPU, TRAP AND
EIS TESTS. IN THE FIRST AND SECOND PARTS, THE PROGRAM
WILL HALT ON ERROR. IN PART THREE, EIS TEST,
WHEN AN ERROR IS DETECTED, THE ERROR PC AND ERROR
NUMBER WILL BE TYPED, THEN THE PROGRAM WILL
CONTINUE EXECUTION.
LOOP ON ERROR IS PROVIDED BY MANUALLY MODIFYING
SOME APPROPRIATE MEMORY LOCATIONS. SEE THE LISTING
OF THAT TEST FOR DETAILS AND INSTRUCTIONS.

THIS PROGRAM ASSUMES SOME OPTIONS (FOR EIS TEST ONLY),
THEY ARE:
1. ENABLE ERROR PRINTOUTS, 2.CONTINUE EXECUTION ON ERROR.

1.3 ABSTRACT

PART ONE:

CPU TEST, THIS IS THE FIRST PART OF THE MAIN PROGRAM.
THIS TEST CHECK OUT THE BASIC PDP-11 INSTRUCTIONS IN EVERY
ADDRESSING MODES WITH VARIOUS TYPES OF DATA PATTERNS.

PART TWO:

TRAP TEST, THIS IS THE SECOND PART OF THE MAIN PROGRAM.
THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE
TRAPS. ALSO TESTED ARE TRAP OVERFLOW CONDITIONS,
ODDITIES OF REGISTER 6, INTERRUPTS , THE RESET AND WAIT
INSTRUCTIONS. THIS PROGRAM CHECKS THAT ON ALL TRAP
OPERATIONS REGISTER 6 IS DECREMENTED THE CORRECT AMOUNT,
THAT THE CORRECT PC IS SAVED ON THE STACK, THAT THE OLD
CONDITION CODES AND PRIORITY ARE PLACED ON THE STACK AND
THAT THE NEW STATUS AND CONDITION CODES ARE CORRECT.
BOTH THE "TRAP" AND "EMT" TRAP INSTRUCTIONS ARE TESTED
TO SEE THAT ALL COMBINATIONS WILL TRAP. CHECKED ALSO
IS THAT ALL RESERVED INSTRUCTIONS WILL TRAP. THE TRACE BIT
IS CHECKED TO SEE IF IT CAUSES A TRAP.
THE RTI AND RTT INSTRUCTIONS ARE CHECKED. STACK OVERFLOW IS
ALSO CHECKED FOR ALL THE TRAP INSTRUCTIONS.

```
                        SPECIAL CHECKS ARE MADE TO SEE IF BUS
                        ERROR TRAPS OCCUR ON NON-EXISTENT MEMORY.
                        ALL INSTRUCTIONS THAT ARE RESERVED SHOULD TRAP TO LOCATION 10,
                        AND THE PC THAT POINTS TO THE TRAPPING INSTRUCTION
                        SHOULD BE PLACED ON THE STACK.


                    PART THREE:

                        THIS PROGRAM TESTS THE EXTENDED INSTRUCTION SET
                        <ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT LEAST
                        ONCE WITH EACH INSTRUCTION.
                        THIS PROGRAM TESTS ALL THE EIS INSTRUCTIONS OF THE 11/34
                        FOR ASH AND ASHC INSTRUCTIONS EVERY EVEN PASS IS EXECUTED
                        WITH DESTINATION MODE 0 FOR ALL REGISTERS AND EVERY ODD PASS
                        WITH DESTINATION MODE OF 67. THE DIAGNOSTIC DOES NOT MAKE A
                        PASS WITH T BIT SET.


            2.0     HARDWARE REQUIREMENT
                    --------------------

                        A PROCESSOR WITH DCF11-AA CHIP SET, A MINIMUN OF 16K OF
                        MEMORY AND A CONSOLE TERMINAL.  IF PROGRAM IS RUNNING
                        UNDER APT OR ACT, THE CONSOLE TERMINAL IS NOT NECESSARY.


            3.0     RELATED DOCUMENTS AND STANDARDS:

                        ACT11/XXDP PROGRAMMING SPECIFICATION
                        STANDARD APT SYSTEM TO A PDP11 DIAGNOSTIC INTERFACE
                        PDP11 MAINDEC SYSMAC PACKAGE
                        KDF11-A MODULE SPECIFICATION


            4.0     STARTING PROCEDURES
                    -------------------

                        THE PROGRAM IS STARTED BY LOADING ADDRESS 200.
                        THE RESTART ADDRESS IS 1024.
                        PROGRAM IDENTIFICATION WILL BE TYPED AFTER THE FIRST
                        PASS OF THE WHOLE PROGRAM.


            5.0     TRAPCATCHER ABSTRACTS
                    ---------------------

                        THIS IS A SERIES OF INSTRUCTIONS DESIGNED TO DETECT AND
                        ISOLATE UNEXPECTED TRAPS AND INTERRUPTS, THAT OCCUR IN THE
                        TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

                        THE PRINCIPLE OF THIS ROUTINE IS: THE VECTOR ENTRANCE
                        ADDRESS POINTS TO THE NEXT SEQUENTIAL WORD WHICH WILL CON-
                        TAIN A HALT (000000) (THIS LOCATION IS ALSO THE STATUS
```

```
                        WORD FOR THAT VECTOR ENTRANCE. BUT THIS WILL HAVE NO EFFECT
                        ON IT ALSO BEING THE NEXT INSTRUCTION).

                        IF A HALT OCCURS IN THE TRAP OR INTERRUPT VECTOR AREA,
                        REGISTER SIX SHOULD BE EXAMINED TO DETERMINE ITS CONTENTS,
                        THEN USE REGISTER SIX CONTENTS AS AN ADDRESS TO DETERMINE
                        WHERE THE PROGRAM WAS. WHEN THE INTERRUPT OR
                        TRAP OCCURRED; MEMORY AS SPECIFIED BY R6 CONTAINS THE
                        PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE THE
                        TRAP OCCURRED.
                        THE CONTENTS OF LOCATION '$TESTN'(304) CONTAINS
                        THE TEST NUMBER THAT IT WAS DOING BEFORE IT
                        TRAPPED.


6.1     ERROR HANDLING IN PART ONE AND PART TWO
        ---------------------------------------

                        IN PARTS ONE AND TWO, ALL ERRORS WILL CAUSE A HALT.

                        THE PROGRAM CHECKS TO SEE THAT THE PC. DOESN'T JUMP ERRATICALLY
                        WITHIN THE TESTS BY USING A SEQUENCE COUNT CALLED '$TESTN'.

                EXAMPLE

                TSTA:   INC     (R2)            ;INCREMENT THE TEST NUMBER
                        CMP     #A,(R2)         ;COMPARE FOR THE RIGHT TEST
                        BNE     TSTA+1-10       ;IF NOT CORRECT BRANCH TO A HALT


                        * R2 CONTAINS THE ADDRESS OF $TESTN (304).
                          A IS THE CURRENT TEST NUMBER.


                        IF AN ERROR IS DETECTED, THE PROGRAM WILL HALT
                        IT COULD BE BECAUSE OF TWO REASONS.
                        A)      WRONG TEST NUMBER (SEQUENCE ERROR)
                        B)      ERROR IN THE PRESENT TEST.

                        THE TEST SEQUENCE COUNT ''TESTN'' SHOULD BE CHECKED FIRST
                        TO SEE IF IT MATCHES THE PRESENT TEST.
                        IF IT DOESN'T MATCH ; THEN THE CONTENTS OF THIS LOCATION
                        TELL YOU WHICH TEST IT WAS DOING BEFORE IT HALTED.


6.2     ERROR HANDLING IN PART THREE
        ----------------------------

                        IN PART THREE, ANY ERROR, INCLUDES SEQUENCE CHECK ERROR WILL
                        CAUSE THE ERROR MESSAGE TO BE TYPED.  THE PROGRAM WILL
                        CONTINUE EXECUTION AFTER TYPE OUT.

                        THE ERROR REPORTING FORMAT IS AS FOLLOWS:

                        ERROR.  PC AND ERROR # ARE:
                         PC #
```

ERROR #

7.0 SWITCH SETTINGS (APPLICABLE ONLY TO PART THREE).

SINCE NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE SWITCH REGISTER. THE INITIAL CONTENT OF LOC. 176 IS 000000, THE USER MAY PRE-SET THIS LOCATION BEFORE STARTING THE PROGRAM.

BIT #	OCTAL VALUE	FUNCTION
15	100000	HALT ON ERROR
13	020000	INHIBIT ERROR PRINTOUT

ALSO, WITHIN THE APT TABLE, AN 8 BIT BYTE $ENVM [LOCATION 321] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE $ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN LOCATION 320.

8.0 EXECUTION TIMES

THE RUN TIME FOR A SINGLE RUN (THE FIRST PASS) IS ONE SECOND.
AFTER THE FIRST PASS, THE PROGRAM WILL ITERATE EVERY 15 TIMES BEFORE THE END OF PASS MESSAGE IS TYPED AGAIN.
THE RUN TIME FOR EACH ADDITIONAL END OF PASS MESSAGE TYPED IS APPROXIMATELY 15 SECONDS.

9.0 ROUTINES ABSTRACT

9.1 HALT ROUTINE (APPLICABLE ONLY TO PART THREE).

THIS ROUTINE IS CALLED VIA A JSR INSTRUCTION EACH TIME AN ERROR IS SEEN AND AN ERROR MESSAGE IS THEN TYPED OUT UNLESS IT IS SUPRESSED BY THE SWITCHES.
THE COMMENTS BESIDE THE CALL TO THE HALT SUBROUTINE TELLS WHAT WAS BEING TESTED AND WHAT WAS EXPECTED. ALL PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION $ENVM IS HIGH.
WHILE RUNNING UNDER AP* THE DIAGNOSTIC WILL NOT SUPPORT SPOOLING OF CONSOLE OUTPUTS.

9.2 POWER FAIL ROUTINE

```
                                                                IF A POWER FAIL OCCURS (FOLLOWED BY A POWER UP), THE
                                                                MESSAGE 'POWER FAIL'' IS TYPED OUT AND THE PROGRAM WILL
                                                                RESTART EXECUTION AT 'RESTRT''.


                                                        .ENDR
```

```
                                ;PROGRAMMER: KIN C. LEE

                                        .TITLE  CJKDB-B  DCF11-AA CPU DIAG.
                                        .ENABLE ABS
                                        .NLIST  CND,MC,MD
                                        .LIST   ME
                        000240          SCOPE=NOP
                        000007          R7=%7
                        000006          R6=%6
                        177776          PS=177776
                        177560          TKS=177560
                        177562          TKB=177562
                        177564          TPS=177564
                        177566          TPB=177566
                        140000          USRM=140000
                        030000          PUSRM=30000
                                .SBTTL  ACT11 HOOKS

                                ;*******************************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000400          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  060472                          $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
                        000300          .-300
                                .SBTTL  APT MAILBOX-ETABLE

                                ;*******************************************************************************
                                .EVEN
000300                  $MAIL:                                  ;;APT MAILBOX
000300  000000          $MSGTY: .WORD   AMSGTY                  ;;MESSAGE TYPE CODE
000302  000000          $FATAL: .WORD   AFATAL                  ;;FATAL ERROR NUMBER
000304  000000          $TESTN: .WORD   ATESTN                  ;;TEST NUMBER
000306  000000          $PASS:  .WORD   APASS                   ;;PASS COUNT
000310  000000          $DEVCT: .WORD   ADEVCT                  ;;DEVICE COUNT
000312  000000          $UNIT:  .WORD   AUNIT                   ;;I/O UNIT NUMBER
000314  000000          $MSGAD: .WORD   AMSGAD                  ;;MESSAGE ADDRESS
000316  000000          $MSGLG: .WORD   AMSGLG                  ;;MESSAGE LENGTH
000320                  $ETABLE:                                ;;APT ENVIRONMENT TABLE
000320      000         $ENV:   .BYTE   AENV                    ;;ENVIRONMENT BYTE
000321      000         $ENVM:  .BYTE   AENVM                   ;;ENVIRONMENT MODE BITS
000322  000000          $SWREG: .WORD   ASWREG                  ;;APT SWITCH REGISTER
000324  000000          $USWR:  .WORD   AUSWR                   ;;USER SWITCHES
000326  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
000330                  $ETEND:
                                .MEXIT
```

```
                                .SBTTL  APT PARAMETER BLOCK

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

                                $APTHD:
                                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
                                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
                                $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
                                $PASTM: .WORD   20      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
                                $UNITM: .WORD   5       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
                                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                                ;*****************************************************************************
                                ;SOME POINTERS TO CPU TRAP HANDLERS
                                ;*****************************************************************************
                                        .=4
                                        TO4
                                        0
                                        TO10
                                        0
                                        TO14
                                        .-30
                                        TO30
                                        0
                                        TO34
                                        0
                                        .=114
                                        TO114
                                        0
                                        .=244
                                        TO244
                                        0
                                        TO250
                                        0

                                .=172
                                LPADR:          0       ;LOOP ADDRESS (EIS TEST)
                                DISPREG:        0       ;SOFTWARE DISPLAY REGISTER
                                SWREG:          0       ;SOFTWARE SWITCH REGISTER

                                ;*****************************************************************************
                                ;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
                                ;*****************************************************************************
                                        .-370
                                        0,0,0,0,0,0
```

```
                                        1,1,-1
                                ;*****************************************************************************************
                                ;SET UP STARTING ADDRESS
                                $ERROR=$FATAL
                                $TSTNM=$TESTN
                                .=1000
001000  000000                  STBOT:  .WORD 0                         ;STACK POINTER

                                .=200
000200  000167  000576                  JMP     START
000204  012706  001000                  MOV     #STBOT,R6               ;SET STACK POINTER
000210  012702  000304                  MOV     #$TESTN,R2              ;SET MAILBOX POINTER
000214  000137                          JMP     @(PC)+                  ;JUMP TO SUBTEST
000216  000000                          0                               ;ADDR. OF SUBTEST GOES HERE

                                .=1002
                                .SBTTL **STARTING OF CPU TEST **
001002  012737  060716  000024  START:  MOV     #PWRDN,@#24             ;SET UP FOR POWER FAIL
001010  012737  000000  000306          MOV     #0,@#$PASS      ;CLEAR PASS COUNT
001016  012737  000016  062304          MOV     #16.,@#PASSPT   ;SET PRINT COUNTER
001024  012706  001000          RESTRT: MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
001030  012702  000304                  MOV     #$TESTN,R2      ;SET UP POINTER TO MESSAGE TYPE
001034  012737  000000  000304          MOV     #0,@#$TSTNM     ;CLEAR TEST NUMBER
001042  012737  000000  000302          MOV     #0,@#$ERROR     ;CLEAR ERROR NUMBER
001050  012737  000000  000300          MOV     #0,@#$MSGTY     ;CLEAR MESSAGE TYPE(FOR APT)
                                ;*****************************************************************************************
                                ;TEST 1 CHECK BRANCHES ON Z BIT
                                ;*****************************************************************************************
001056  005212                  TS1:    INC     (R2)            ;UPDATE TEST NUMBER
001060  022712  000001                  CMP     #1,(R2)         ;SEQUENCE ERROR?
001064  001024                          BNE     TS2-10  ;BR TO ERROR HALT ON SEQ ERROR
001066  000257                          CCC                     ;CLEAR ALL CONDITION CODES
001070  001401                          BEQ     BRA1            ;SHOULD BRANCH
001072  000404                          BR      BRA2            ;BAD BRANCH OF Z-BIT
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           BRANCH INSTRUCTION AND         <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           FOLLOWING W/ 774               <====
001074                          BRA1:
001074  012742  000001                  MOV     #1,-(R2)        ;MOVE TO MAILBOX #  *******  1  *******
001100  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001102  000000                          HALT                    ;SHOULD HAVE BRANCHED: Z=0
001104                          BRA2:
001104  001004                          BNE     BRA3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
001106  012742  000002                  MOV     #2,-(R2)        ;MOVE TO MAILBOX #  *******  2  *******
001112  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001114  000000                          HALT                    ;
001116  000264                  BRA3:   SEZ
001120  001001                          BNE     BRA4
001122  000404                          BR      BRA5
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           BRANCH INSTRUCTION AND         <====
```

```
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           FOLLOWING W/ 760               <====
001124                                          BRA4:
001124  012742  000003                                  MOV     #3,-(R2)        ;MOVE TO MAILBOX #  *******  3  *******
001130  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001132  000000                                          HALT                    ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
001134                                          BRA5:
001134  001404                                          BEQ     TS2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <===-
                                                        ;           REPLACE THE MOVE INSTRUCTION   <==-_
                                                        ;           WHICH FOLLOWS W/ 753           <====
001136  012742  000004                                  MOV     #4,-(R2)        ;MOVE TO MAILBOX #  *******  4  *******
001142  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001144  000000                                          HALT                    ;SHOULD HAVE BRANCHED ON Z=1
                                                        ;  OR SEQUENCE ERROR

                                        ;***************************************************************************************
                                        .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.
                                        ;***************************************************************************************
                                        ;TEST 2 TEST OF ZEROES IN THE DATA PATH
                                        ;***************************************************************************************
001146  005212                          TS2:    INC     (R2)            ;UPDATE TEST NUMBER
001150  022712  000002                          CMP     #2,(R2)         ;SEQUENCE ERROR?
001154  001006                                  BNE     TS3-10  ;BR TO ERROR HALT ON SEQ ERROR
001156  012737  000000  000000                  MOV     #0,@#0          ;MOVE ZEROES THRU ADDRESS LINES, DATA
                                                                        ;LINES AND INTERNAL PATHS
001164  005737  000000                          TST     @#0             ;SUCCESSFUL?
001170  001404                                  BEQ     TS3
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
001172  012742  000005                                  MOV     #5,-(R2)        ;MOVE TO MAILBOX #  *******  5  *******
001176  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001200  000000                                          HALT                    ;DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR

                                        ;***************************************************************************************
                                        ;TEST 3 TEST OF PATTERN 125252 IN DATA PATH
                                        ;***************************************************************************************
001202  005212                          TS3:    INC     (R2)            ;UPDATE TEST NUMBER
001204  022712  000003                          CMP     #3,(R2)         ;SEQUENCE ERROR?
001210  001007                                  BNE     TS4-10  ;BR TO ERROR HALT ON SEQ ERROR
001212  012737  125252  000000                  MOV     #125252,@#0     ;MOVE ALTERNATING ONES AND ZEROES
                                                                        ;THRU DATA PATHS
001220  022737  125252  000000                  CMP     #125252,@#0     ;SUCCESSFUL
```

```
001226  001404                          BEQ     TS4
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 770           <====
001230  012742  000006                  MOV     #6,-(R2)        ;MOVE TO MAILBOX #  *******  6  *******
001234  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001236  000000                          HALT                    ;DATA INCORRECT
                                        ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 4 TEST OF PATTERN 052525 IN DATA PATH
                                ;*******************************************************************************
001240  005212                  TS4:    INC     (R2)            ;UPDATE TEST NUMBER
001242  022712  000004                  CMP     #4,(R2)         ;SEQUENCE ERROR?
001246  001007                          BNE     TS5-10  ;BR TO ERROR HALT ON SEQ ERROR
001250  012737  052525  000000          MOV     #052525,@#0     ;MOVE ALTERNATING ZEROES AND ONES
                                                                ;THRU DATA PATH
001256  022737  052525  000000          CMP     #052525,@#0     ;SUCCESSFUL?
001264  001404                          BEQ     TS5
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 770           <====
001266  012742  000007                  MOV     #7,-(R2)        ;MOVE TO MAILBOX #  *******  7  *******
001272  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001274  000000                          HALT                    ;DATA INCORRECT
                                        ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 5 TEST OF ALL ONES IN DATA PATH
                                ;*******************************************************************************
001276  005212                  TS5:    INC     (R2)            ;UPDATE TEST NUMBER
001300  022712  000005                  CMP     #5,(R2)         ;SEQUENCE ERROR?
001304  001007                          BNE     TS6-10  ;BR TO ERROR HALT ON SEQ ERROR
001306  012737  177777  000000          MOV     #177777,@#0     ;MOVE ONES THRU DATA PATH
001314  022737  177777  000000          CMP     #177777,@#0     ;SUCCESSFUL
001322  001404                          BEQ     TS6
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= -=
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <===-
                                        ;           WHICH FOLLOWS W/ 770           <= -=
001324  012742  000010                  MOV     #10,-(R2)       ;MOVE TO MAILBOX #  *******  10  *******
001330  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001332  000000                          HALT                    ;DATA INCORRECT
                                        ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                .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 6 SHIFT BIT 0 TO BIT 1
                                ;*************************************************************************************
001334  005212                TS6:    INC     (R2)            ;UPDATE TEST NUMBER
001336  022712  000006                CMP     #6,(R2)         ;SEQUENCE ERROR?
001342  001012                        BNE     TS7-10  ;BR TO ERROR HALT ON SEQ ERROR
001344  000241                        CLC                     ;CLEAR CARRY BIT
001346  012737  000001  000000        MOV     #1,@#0          ;LOAD A 1
001354  006137  000000                ROL     @#0             ;SHIFT LEFT
001360  022737  000002  000000        CMP     #2,@#0          ;SUCCESSFUL
001366  001404                        BEQ     TS7
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
001370  012742  000011                MOV     #11,-(R2)       ;MOVE TO MAILBOX #  *******  11  *******
001374  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001376  000000                        HALT                    ;BIT 1 NOT SET
                                                        ;  OR SEQUENCE ERROR
                                ;*************************************************************************************
                                ;TEST 7 SHIFT CARRY INTO BIT 0
                                ;*************************************************************************************
001400  005212                TS7:    INC     (R2)            ;UPDATE TEST NUMBER
001402  022712  000007                CMP     #7,(R2)         ;SEQUENCE ERROR?
001406  001017                        BNE     TS10-10         ;BR TO ERROR HALT ON SEQ ERROR
001410  012737  000000  000000        MOV     #0,@#0          ;CLEAR LOCATION
001416  000261                        SEC                     ;SET CARRY
001420  006137  000000                ROL     @#0             ;ROTATE CARRY BIT TO BIT 0
001424  103014                        BCC     TS10
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
001426  012742  000012                MOV     #12,-(R2)       ;MOVE TO MAILBOX #  *******  12  *******
001432  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001434  000000                        HALT                    ;CARRY CLEAR
                                                        ;  OR SEQUENCE ERROR
001436  022737  000001  000000        CMP     #1,@#0          ;BIT 0 SET
001444  001404                        BEQ     TS10
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
001446  012742  000013                MOV     #13,-(R2)       ;MOVE TO MAILBOX #  *******  13  *******
001452  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001454  000000                        HALT                    ;BIT 0 NOT SET
                                                        ;  OR SEQUENCE ERROR

                                ;*************************************************************************************
                                ;TEST 10        LEFT SHIFT FROM BIT 0 TO C-BIT
```

```
                                ;*****************************************************************************
001456  005212                  TS10:   INC     (R2)            ;UPDATE TEST NUMBER
001460  022712  000010                  CMP     #10,(R2)        ;SEQUENCE ERROR?
001464  001014                          BNE     TS11-10         ;BR TO ERROR HALT ON SEQ ERROR
001466  012737  000001  000000          MOV     #1,@#0          ;SET BIT 0
001474  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
001500  000241                          CLC                     ;CLEAR C-BIT
001502  005200                  SHL:    INC     R0              ;INCREMENT BIT COUNTER
001504  001404                          BEQ     SHLE            ;BR TO ERROR HALT IF BIT IS LOST
001506  006137  000000                  ROL     @#0             ;SHIFT LEFT ONE POSITION
001512  103373                          BCC     SHL             ;BRANCH IF C-BIT NOT SET
001514  001404                          BEQ     TS11
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
                                                                ;           CONDITIONAL BRANCH INST. AND   <
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
                                                                ;           WHICH FOLLOWS W/ 763           <
001516                  SHLE:
001516  012742  000014                  MOV     #14,-(R2)       ;MOVE TO MAILBOX #  *******  14  *******
001522  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001524  000000                          HALT                    ;LEFT SHIFTING LOGIC FAILED
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 11         SHIFT BIT 15 TO BIT 14
                                ;*****************************************************************************
001526  005212                  TS11:   INC     (R2)            ;UPDATE TEST NUMBER
001530  022712  000011                  CMP     #11,(R2)        ;SEQUENCE ERROR?
001534  001012                          BNE     TS12-10         ;BR TO ERROR HALT ON SEQ ERROR
001536  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
001544  000241                          CLC                     ;CLEAR CARRY
001546  006037  000000                  ROR     @#0             ;SHIFT BIT 15 TO BIT 14
001552  022737  040000  000000          CMP     #40000,@#0      ;SUCCESSFUL
001560  001404                          BEQ     TS12
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <====
001562  012742  000015                  MOV     #15,-(R2)       ;MOVE TO MAILBOX #  *******  15  *******
001566  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001570  000000                          HALT                    ;BIT 14 NOT SET
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 12         RIGHT SHIFT FROM BIT 15 TO C-BIT
                                ;*****************************************************************************
001572  005212                  TS12:   INC     (R2)            ;UPDATE TEST NUMBER
001574  022712  000012                  CMP     #12,(R2)        ;SEQUENCE ERROR?
001600  001014                          BNE     TS13-10         ;BR TO ERROR HALT ON SEQ ERROR
001602  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
001610  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
001614  000241                          CLC                     ;CLEAR C-BIT
001616  005200                  SHR:    INC     R0              ;INCREMENT BIT COUNTER
001620  001404                          BEQ     SHRE            ;BR TO ERROR HALT IF BIT IS LOST
001622  006037  000000                  ROR     @#0             ;ROTATE RIGHT ONE POSITION
001626  103373                          BCC     SHR             ;BRANCH IF C-BIT CLEAR
001630  001404                          BEQ     TS13
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
                                                        ;           CONDITIONAL BRANCH INST. AND   <
                                                        ;           REPLACE THE MOVE INSTRUCTION   <
                                                        ;           WHICH FOLLOWS W/ 763           <--
001632                          SHRE:
001632  012742  000016                  MOV     #16.,-(R2)      ;MOVE TO MAILBOX #  *******  16  *******
001636  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001640  000000                          HALT                    ;RIGHT SHIFT LOGIC FAILED
                                                                ;  OR SEQUENCE ERROR

                        ;*******************************************************************************
                        .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 13        TEST IF R0 CAN HOLD ALL ZEROES
                        ;*******************************************************************************
001642  005212          TS13:   INC     (R2)            ;UPDATE TEST NUMBER
001644  022712  000013          CMP     #13,(R2)        ;SEQUENCE ERROR?
001650  001004                  BNE     TS14-10         ;BR TO ERROR HALT ON SEQ ERROR

001652  012700  000000          MOV     #0,R0           ;MOVE ZEROES TO R0
001656  005700                  TST     R0              ;SUCCESSFUL?
001660  001404                  BEQ     TS14
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
                                                        ;           CONDITIONAL BRANCH INST. AND   <
                                                        ;           REPLACE THE MOVE INSTRUCTION   <
                                                        ;           WHICH FOLLOWS W/ 773           <
001662  012742  000017                  MOV     #17,-(R2)       ;MOVE TO MAILBOX #  *******  17  *******
001666  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001670  000000                          HALT                    ;R0 NOT 0
                                                                ;  OR SEQUENCE ERROR

                        ;*******************************************************************************
                        ;TEST 14        TEST IF R0 CAN HOLD ONES AND ZEROES
                        ;*******************************************************************************
001672  005212          TS14:   INC     (R2)            ;UPDATE TEST NUMBER
001674  022712  000014          CMP     #14,(R2)        ;SEQUENCE ERROR?
001700  001005                  BNE     TS15-10         ;BR TO ERROR HALT ON SEQ ERROR
001702  012700  125252          MOV     #125252,R0      ;MOVE ALTERNATING ONES AND ZEROES TO R0
```

```
001706  020027  125252                  CMP     R0,#125252      ;SUCCESSFUL?
001712  001404                          BEQ     TS15
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-- -
                                                                ;           CONDITIONAL BRANCH INST. AND   <- --
                                                                ;           REPLACE THE MOVE INSTRUCTION   < --
                                                                ;           WHICH FOLLOWS W/ 772           <
001714  012742  000020                  MOV     #20,-(R2)       ;MOVE TO MAILBOX #  *******  20  *******
001720  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001722  000000                          HALT                    ;R0 NOT 125252
                                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************************
                        ;TEST 15        TEST IF R0 CAN HOLD ZEROES AND ONES
                        ;*****************************************************************************************
001724  005212          TS15:   INC     (R2)            ;UPDATE TEST NUMBER
001726  022712  000015          CMP     #15,(R2)        ;SEQUENCE ERROR?
001732  001005                  BNE     TS16-10         ;BR TO ERROR HALT ON SEQ ERROR
001734  012700  052525          MOV     #052525,R0      ;MOVE ALTERNATING ZEROES AND ONES TO R0
001740  020027  052525          CMP     R0,#052525      ;SUCCESSFUL?
001744  001404                  BEQ     TS16
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <   -
                                                        ;           CONDITIONAL BRANCH INST. AND   <-
                                                        ;           REPLACE THE MOVE INSTRUCTION   <
                                                        ;           WHICH FOLLOWS W/ 772           < --
001746  012742  000021                  MOV     #21,-(R2)       ;MOVE TO MAILBOX #  *******  21  *******
001752  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001754  000000                          HALT                    ;R0 NOT 52525
                                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************************
                        ;TEST 16        TEST IF R0 CAN HOLD ALL ONES
                        ;*****************************************************************************************
001756  005212          TS16:   INC     (R2)            ;UPDATE TEST NUMBER
001760  022712  000016          CMP     #16,(R2)        ;SEQUENCE ERROR?
001764  001005                  BNE     TS17-10         ;BR TO ERROR HALT ON SEQ ERROR
001766  012700  177777          MOV     #177777,R0      ;MOVE ALL ONES TO R0
001772  020027  177777          CMP     R0,#177777      ;SUCCESSFUL?
001776  001404                  BEQ     TS17
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
002000  012742  000022                  MOV     #22,-(R2)       ;MOVE TO MAILBOX #  *******  22  *******
002004  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002006  000000                          HALT                    ;R0 NOT 177777
                                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************************
                        ;TEST 17        TEST IF R1 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************************
002010  005212          TS17:   INC     (R2)            ;UPDATE TEST NUMBER
002012  022712  000017          CMP     #17,(R2)        ;SEQUENCE ERROR?
002016  001012                  BNE     TS20-10         ;BR TO ERROR HALT ON SEQ ERROR
002020  012701  000001          MOV     #1,R1           ;SET BIT 0
002024  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002030  000241                  CLC                     ;CLEAR C-BIT
```

```
                                REG1:   INC     R0              ;INCREMENT BIT COUNTER
                                        BEQ     REG1E           ;BR TO ERROR HALT IF BIT IS LOST
                                        ROL     R1              ;ROTATE 1 POSITION
                                        BCC     REG1            ;ALL DONE
                                        BEQ     TS20
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===-
                                                                ;           CONDITIONAL BRANCH INST. AND   <---=
                                                                ;           REPLACE THE MOVE INSTRUCTION   < - -
                                                                ;           WHICH FOLLOWS W/ 765           <--=-
                                REG1E:
                                        MOV     #23,-(R2)       ;MOVE TO MAILBOX #  *******  23  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;FAILURE WITH R1
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 20         TEST IF R1 CAN HOLD A ZERO IN ALL BITS
                                ;*******************************************************************************
                                TS20:   INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #20,(R2)        ;SEQUENCE ERROR?
                                        BNE     TS21-10         ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #-2,R1          ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        SEC                     ;SET C-BIT
                                REG1A:  INC     R0              ;INCREMENT COUNTER
                                        BEQ     R1ERR           ;BR TO ERROR HALT IF COUNTER=0
                                        ROL     R1              ;ROTATE 1 POSITION
                                        BCS     REG1A           ;CONTINUE UNTIL C-BIT IS CLEAR
                                        CMP     #-1,R1          ;CHECK DATA IN R1
                                        BEQ     TS21
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
                                                                ;           CONDITIONAL BRANCH INST. AND   <===-
                                                                ;           REPLACE THE MOVE INSTRUCTION   <=- -
                                                                ;           WHICH FOLLOWS W/ 763           <====
                                R1ERR:
                                        MOV     #24,-(R2)       ;MOVE TO MAILBOX #  *******  24  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;FAILURE WITH R1
                                                                ;  OR SEQUENCE ERROR
                                ;*******************************************************************************
                                ;TEST 21         TEST IF R2 CAN HOLD A ONE IN ALL BITS
                                ;*******************************************************************************
                                TS21:   INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #21,(R2)        ;SEQUENCE ERROR?
                                        BNE     REG2A-14                ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #1,R2           ;SET BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        CLC                     ;CLEAR C-BIT
                                REG2:   INC     R0              ;INCREMENT BIT COUNTER
                                        BEQ     REG2A-14        ;BR TO ERROR HALT IF BIT IS LOST
                                        ROL     R2              ;ROTATE 1 POSITION
                                        BCC     REG2            ;ALL DONE
                                        BEQ     REG2A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           BRANCH INSTRUCTION AND         <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                        ;          FOLLOWING W/ 771              <====
002160  012702  000304                          MOV     #$TESTN,R2      ;RESTORE POINTER
002164  012742  000025                          MOV     #25,-(R2)       ;MOVE TO MAILBOX #  *******  25  *******
002170  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002172  000000                                  HALT                    ;FAILURE WITH R2
002174  012702  000304                  REG2A:  MOV     #$TESTN,R2              ;RESTORE POINTER

                                        ;*******************************************************************************
                                        ;TEST 22        TEST IF R2 CAN HOLD A ZERO IN ALL BITS
                                        ;*******************************************************************************
002200  005212                          TS22:   INC     (R2)            ;UPDATE TEST NUMBER
002202  022712  000022                          CMP     #22,(R2)        ;SEQUENCE ERROR?
002206  001020                                  BNE     TS23-10         ;BR TO ERROR HALT ON SEQ ERROR
002210  012702  177776                          MOV     #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
002214  012700  177757                          MOV     #-21,R0         ;SET BIT COUNTER
002220  000261                                  SEC                     ;SET C-BIT
002222  005200                          REG2B:  INC     R0              ;INCREMENT BIT COUNTER
002224  001407                                  BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
002226  006102                                  ROL     R2              ;ROTATE 1 POSITION
002230  103774                                  BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
002232  022702  177777                          CMP     #-1,R2          ;CHECK DATA IN R2
002236  001406                                  BEQ     REG2C
002240  012702  000304                          MOV     #$TESTN,R2      ;RESTORE POINTER
002244                                  R2ERR:
002244  012742  000026                          MOV     #26,-(R2)       ;MOVE TO MAILBOX #  *******  26  *******
002250  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002252  000000                                  HALT                    ;FAILURE WITH R2
002254  012702  000304                  REG2C:  MOV     #$TESTN,R2      ;RESTORE POINTER

                                        ;*******************************************************************************
                                        ;TEST 23        TEST IF R3 CAN HOLD A ONE IN ALL BITS
                                        ;*******************************************************************************
002260  005212                          TS23:   INC     (R2)            ;UPDATE TEST NUMBER
002262  022712  000023                          CMP     #23,(R2)        ;SEQUENCE ERROR?
002266  001012                                  BNE     TS24-10         ;BR TO ERROR HALT ON SEQ ERROR
002270  012703  000001                          MOV     #1,R3           ;SET BIT 0
002274  012700  177757                          MOV     #-21,R0         ;SET BIT COUNTER
002300  000241                                  CLC                     ;CLEAR C-BIT
002302  005200                          REG3:   INC     R0              ;INCREMENT BIT COUNTER
002304  001403                                  BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
002306  006103                                  ROL     R3              ;ROTATE 1 POSITION
002310  103374                                  BCC     REG3            ;ALL DONE
002312  001404                                  BEQ     TS24

                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <= =
                                                ;           CONDITIONAL BRANCH INST. AND    < - -
                                                ;           REPLACE THE MOVE INSTRUCTION    <-
                                                ;           WHICH FOLLOWS W/ 765            <--=-

002314                                  REG3E:
002314  012742  000027                          MOV     #27,-(R2)       ;MOVE TO MAILBOX #  *******  27  *******
002320  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002322  000000                                  HALT                    ;FAILURE WITH R3
                                                ;  OR SEQUENCE ERROR

                                        ;*******************************************************************************
                                        ;TEST 24        TEST IF R3 CAN HOLD A ZERO IN ALL BITS
                                        ;*******************************************************************************
```

```
                                TS24:   INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #24,(R2)        ;SEQUENCE ERROR?
                                        BNE     TS25-10         ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #-2,R3          ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        SEC                     ;SET C-BIT
                                REG3A:  INC     R0              ;INCREMENT BIT COUNTER
                                        BEQ     R3ERR           ;BR TO ERROR HALT IF COUNTER=0
                                        ROL     R3              ;ROTATE 1 POSITION
                                        BCS     REG3A           ;CONTINUE UNTIL C-BIT IS CLEAR
                                        CMP     #-1,R3          ;CHECK DATA
                                        BEQ     TS25
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 763           <====
                                R3ERR:
                                        MOV     #30.,-(R2)      ;MOVE TO MAILBOX #  *******  30  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;FAILURE WITH R3
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 25         TEST IF R4 CAN HOLD A ONE IN ALL BITS
                                ;*******************************************************************************
                                TS25:   INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #25,(R2)        ;SEQUENCE ERROR?
                                        BNE     TS26-10         ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #1,R4           ;SET BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        CLC                     ;CLEAR C-BIT
                                REG4:   INC     R0              ;INCREMENT BIT COUNTER
                                        BEQ     REG4E           ;BR TO ERROR HALT IF BIT IS LOST
                                        ROL     R4              ;ROTATE 1 POSITION
                                        BCC     REG4            ;ALL DONE
                                        BEQ     TS26
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <- ==
                                REG4E:
                                        MOV     #31,-(R2)       ;MOVE TO MAILBOX #  *******  31  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;FAILURE WITH R4
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 26         TEST IF R4 CAN HOLD A ZERO IN ALL BITS
                                ;*******************************************************************************
                                TS26:   INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #26,(R2)        ;SEQUENCE ERROR?
                                        BNE     TS27-10         ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #-2,R4          ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        SEC                     ;SET C-BIT
                                REG4A:  INC     R0              ;INCREMENT BIT COUNTER
```

```
 975  002464  001405                    BEQ     R4ERR           ;BR TO ERROR HALT IF COUNTER-0
 976  002466  006104                    ROL     R4              ;ROTATE 1 POSITION
 977  00247?  103774                    BCS     REG4A           ;CONTINUE UNTIL C-BIT IS CLEAR
 978  00247?  022704  177777            CMP     #-1,R4          ;CHECK DATA
 979  002476  001404                    BEQ     TS27
 980                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 981                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 982                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 983                                                            ;           WHICH FOLLOWS W/ 763           <====
 984  002500                    R4ERR:
 985  002500  012742  000032            MOV     #32,-(R2)       ;MOVE TO MAILBOX #  *******  32  *******
 986  002504  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 987  002506  000000                    HALT                    ;FAILURE WITH R4
 988                                                            ;  OR SEQUENCE ERROR
 989
 990                                                                    .
 991  ;*****************************************************************************************
 992  ;TEST 27         TEST IF R5 CAN HOLD A ONE IN ALL BITS
 993  ;*****************************************************************************************
 994  002510  005212            TS27:   INC     (R2)            ;UPDATE TEST NUMBER
 995  002512  022712  000027            CMP     #27,(R2)        ;SEQUENCE ERROR?
 996  002516  001012                    BNE     TS30-10         ;BR TO ERROR HALT ON SEQ ERROR
 997  002520  012705  000001            MOV     #1,R5           ;SET BIT 0
 998  002524  012700  177757            MOV     #-21,R0         ;SET BIT COUNTER
 999  002530  000241                    CLC                     ;CLEAR C-BIT
1000  002532  005200            REG5:   INC     R0              ;INCREMENT BIT COUNTER
1001  002534  001403                    BEQ     REG5E           ;BR TO ERROR HALT IF BIT IS LOST
1002  002536  006105                    ROL     R5              ;ROTATE 1 POSITION
1003  002540  103374                    BCC     REG5            ;ALL DONE
1004  002542  001404                    BEQ     TS30
1005                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1006                                                            ;           CONDITIONAL BRANCH INST. AND   <====
1007                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
1008                                                            ;           WHICH FOLLOWS W/ 765           <====
1009  002544                    REG5E:
1010  002544  012742  000033            MOV     #33,-(R2)       ;MOVE TO MAILBOX #  *******  33  *******
1011  002550  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1012  002552  000000                    HALT                    ;FAILURE WITH R5
1013                                                            ;  OR SEQUENCE ERROR
1014
1015  ;*****************************************************************************************
1016  ;TEST 30         TEST IF R5 CAN HOLD A ZERO IN ALL BITS
1017  ;*****************************************************************************************
1018  002554  005212            TS30:   INC     (R2)            ;UPDATE TEST NUMBER
1019  002556  022712  000030            CMP     #30,(R2)        ;SEQUENCE ERROR?
1020  002562  001014                    BNE     TS31-10         ;BR TO ERROR HALT ON SEQ ERROR
1021  002564  012705  177776            MOV     #-2,R5          ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
1022  002570  012700  177757            MOV     #-21,R0         ;SET BIT COUNTER
1023  002574  000261                    SEC                     ;SET C-BIT
1024  002576  005200            REG5A:  INC     R0              ;INCREMENT BIT COUNTER
1025  002600  001405                    BEQ     R5ERR           ;BR TO ERROR HALT IF COUNTER=0
1026  002602  006105                    ROL     R5              ;ROTATE 1 POSITION
1027  002604  103774                    BCS     REG5A           ;CONTINUE UNTIL C-BIT IS CLEAR
1028  002606  022705  177777            CMP     #-1,R5          ;CHECK DATA
1029  002612  001404                    BEQ     TS31
1030                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
                                        ;           CONDITIONAL BRANCH INST. AND   <-=--
                                        ;           REPLACE THE MOVE INSTRUCTION   <=- =
                                        ;           WHICH FOLLOWS W/ 763           <-=--
002614                          R5ERR:
002614  012742  000034                  MOV     #34,-(R2)       ;MOVE TO MAILBOX #  *******  34  *******
002620  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002622  000000                          HALT                    ;FAILURE WITH R5
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 31        TEST IF R6 CAN HOLD A ONE IN ALL BITS
                                ;*******************************************************************************
002624  005212                  TS31:   INC     (R2)            ;UPDATE TEST NUMBER
002626  022712  000031                  CMP     #31,(R2)        ;SEQUENCE ERROR?
002632  001012                          BNE     TS32-10         ;BR TO ERROR HALT ON SEQ ERROR
002634  012706  000001                  MOV     #1,R6           ;SET BIT 0
002640  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
002644  000241                          CLC                     ;CLEAR C-BIT
002646  005200                  REG6:   INC     R0              ;INCREMENT BIT COUNTER
002650  001403                          BEQ     REG6E           ;BR TO ERROR HALT IF BIT IS LOST
002652  006106                          ROL     R6              ;ROTATE 1 POSITION
002654  103374                          BCC     REG6            ;ALL DONE
002656  001404                          BEQ     TS32
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <-=--
                                        ;           WHICH FOLLOWS W/ 765           <====
002660                          REG6E:
002660  012742  000035                  MOV     #35,-(R2)       ;MOVE TO MAILBOX #  *******  35  *******
002664  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002666  000000                          HALT                    ;FAILURE WITH R6
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 32        TEST IF R6 CAN HOLD A ZERO IN ALL BITS
                                ;*******************************************************************************
002670  005212                  TS32:   INC     (R2)            ;UPDATE TEST NUMBER
002672  022712  000032                  CMP     #32,(R2)        ;SEQUENCE ERROR?
002676  001014                          BNE     TS33-10         ;BR TO ERROR HALT ON SEQ ERROR
002700  012706  177776                  MOV     #-2,R6          ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
002704  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
002710  000261                          SEC                     ;SET C-BIT
002712  005200                  REG6A:  INC     R0              ;INCREMENT BIT COUNT
002714  001405                          BEQ     R6ERR           ;BR TO ERROR HALT IF COUNTER=0
002716  006106                          ROL     R6              ;ROTATE 1 POSITION
002720  103774                          BCS     REG6A           ;CONTINUE UNTIL C-BIT IS CLEAR
002722  022706  177777                  CMP     #-1,R6          ;CHECK DATA
002726  001404                          BEQ     TS33
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 763           <====
002730                          R6ERR:
002730  012742  000036                  MOV     #36,-(R2)       ;MOVE TO MAILBOX #  *******  36  *******
002734  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002736  000000                          HALT                    ;FAILURE WITH R6
```

```
                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                .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 33        TEST IF PSW WILL HOLD ZEROES
                                ;*****************************************************************************
002740  005212          TS33:   INC     (R2)            ;UPDATE TEST NUMBER
002742  022712  000033          CMP     #33,(R2)        ;SEQUENCE ERROR?
002746  001010                  BNE     TS34-10         ;BR TO ERROR HALT ON SEQ ERROR
002750  012706  001000          MOV     #STBOT,R6
002754  012737  000000  177776  MOV     #0,@#PS         ;SET PSW TO ZERO
002762  005737  177776          TST     @#PS            ;SUCCESSFUL
002766  001404                  BEQ     TS34
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 767           <====
002770  012742  000037          MOV     #37,-(R2)       ;MOVE TO MAILBOX #  *******  37  *******
002774  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002776  000000                  HALT                    ;PSW NOT 0
                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 34        TEST IF PSW WILL HOLD ONES AND ZEROES
                                ;*****************************************************************************
003000  005212          TS34:   INC     (R2)            ;UPDATE TEST NUMBER
003002  022712  000034          CMP     #34,(R2)        ;SEQUENCE ERROR?
003006  001007                  BNE     TS35-10         ;BR TO ERROR HALT ON SEQ ERROR
003010  012737  000252  177776  MOV     #252,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
003016  023727  177776  000252  CMP     @#PS,#252       ;SUCCESSFUL?
003024  001404                  BEQ     TS35
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 770           <====
003026  012742  000040          MOV     #40,-(R2)       ;MOVE TO MAILBOX #  *******  40  *******
003032  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003034  000000                  HALT                    ;PSW NOT 252
                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 35        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
                                ;*****************************************************************************
```

```
1143  003036  005212                  TS35:   INC     (R2)            ;UPDATE TEST NUMBER
1144  003040  022712  000035                  CMP     #35,(R2)        ;SEQUENCE ERROR?
1145  003044  001007                          BNE     TS36-10         ;BR TO ERROR HALT ON SEQ ERROR
1146  003046  012737  000105  177776          MOV     #105,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
1147  003054  023727  177776  000105          CMP     @#PS,#105       ;SUCCESSFUL?
1148  003062  001404                          BEQ     TS36
1149                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1150                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1151                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1152                                                                  ;           WHICH FOLLOWS W/ 770           <====
1153  00306   012742  000041                  MOV     #41,-(R2)       ;MOVE TO MAILBOX #  *******  41  *******
1154  003070  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1155  003072  000000                          HALT                    ;PSW NOT 105
1156                                                                  ;  OR SEQUENCE ERROR
1157
1158                                    ;*****************************************************************************************
1159                                    ;TEST 36        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
1160                                    ;*****************************************************************************************
1161  003074  005212                  TS36:   INC     (R2)            ;UPDATE TEST NUMBER
1162  003076  022712  000036                  CMP     #36,(R2)        ;SEQUENCE ERROR?
1163  003102  001007                          BNE     TS37-10         ;BR TO ERROR HALT ON SEQ ERROR
1164  003104  012737  000357  177776          MOV     #357,@#PS       ;MOVE ONES TO PSW
1165  003112  023727  177776  000357          CMP     @#PS,#357       ;SUCCESSFUL
1166  003120  001404                          BEQ     TS37
1167                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1168                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1169                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1170                                                                  ;           WHICH FOLLOWS W/ 770           <====
1171  003122  012742  000042                  MOV     #42,-(R2)       ;MOVE TO MAILBOX #  *******  42  *******
1172  003126  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1173  003130  000000                          HALT                    ;PSW NOT 357
1174                                                                  ;  OR SEQUENCE ERROR
1175                                    .SBTTL  CONDITION CODE TEST
1176
1177                                    ;*****************************************************************************************
1178                                    ;
1179                                    ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.
1180                                    ;THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1181                                    ;BEQ AND BNE ARE TESTED FOR PROPER EXECUTION.  THEN THE Z-BIT IS
1182                                    ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1183                                    ;AGAIN FOR PROPER OPERATION.
1184                                    ;       THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1185                                    ;CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1186                                    ;BRANCH ROM.  THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1187                                    ;LEAVING THE PC UNALTERED IS TESTED.  ONLY THOSE ROM ADDRESSES SPECIFICALLY
1188                                    ;USED IN THE TEST ARE VERIFIED HERE.
1189                                    ;
1190                                    ;*****************************************************************************************
1191                                    ;TEST 37        TEST BRANCHES AROUND Z-BIT
1192                                    ;*****************************************************************************************
1193  003132  005212                  TS37:   INC     (R2)            ;UPDATE TEST NUMBER
1194  003134  022712  000037                  CMP     #37,(R2)        ;SEQUENCE ERROR?
1195  003140  001014                          BNE     TS40-10         ;BR TO ERROR HALT ON SEQ ERROR
1196                                          ; FIRST WITH Z-BIT ON
1197  003142  000257                          CCC                     ;CC 0100: JUST Z-BIT
1198  003144  000264                          SEZ
```

```
1199  003146  001001                        BNE     BRZ1            ;CHECK OPPOSITE CONDITION
1200  003150  001404                        BEQ     BRZ2
1201                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1202                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1203                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1204                                                                  ;           WHICH FOLLOWS W/ 773           <====
1205  003152                          BRZ1:
1206  003152  012742  000043                MOV     #43,-(R2)       ;MOVE TO MAILBOX #  *******  43  *******
1207  003156  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1208  003160  000000                        HALT                    ;IMPROPER BR W/ Z=1
1209                                        ;CHECK WITH Z-BIT OFF
1210  003162  000277                  BRZ2: SCC                     ;CC=1011: ALL BUT Z-BIT
1211  003164  000244                        CLZ
1212  003166  001401                        BEQ     BRZ3
1213  003170  001004                        BNE     TS40
1214                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1215                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1216                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1217                                                                  ;           WHICH FOLLOWS W/ 763           <====
1218  003172                          BRZ3:
1219  003172  012742  000044                MOV     #44,-(R2)       ;MOVE TO MAILBOX #  *******  44  *******
1220  003176  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1221  003200  000000                        HALT                    ;IMPROPER BR W/ Z=0
1222                                                                  ;  OR SEQUENCE ERROR
1223
1224                            ;*****************************************************************************
1225                            ;
1226                            ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
1227                            ;THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1228                            ;BMI AND BPL ARE TESTED FOR PROPER EXECUTION.  THEN THE N-BIT IS
1229                            ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1230                            ;AGAIN FOR PROPER OPERATION.
1231                            ;       THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1232                            ;CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1233                            ;BRANCH ROM.  THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1234                            ;LEAVING THE PC UNALTERED IS TESTED.  ONLY THOSE ROM ADDRESSES SPECIFICALLY
1235                            ;USED IN THE TEST ARE VERIFIED HERE.
1236                            ;
1237                            ;*****************************************************************************
1238                            ;TEST 40        TEST BRANCHES AROUND N-BIT
1239                            ;*****************************************************************************
1240  003202  005212            TS40:   INC     (R2)            ;UPDATE TEST NUMBER
1241  003204  022712  000040            CMP     #40,(R2)        ;SEQUENCE ERROR?
1242  003210  001014                    BNE     TS41-10         ;BR TO ERROR HALT ON SEQ ERROR
1243                                    ;FIRST WITH N-BIT ON
1244  003212  000257                    CCC                     ;CC=1000: JUST N-BIT
1245  003214  000270                    SEN
1246  003216  100001                    BPL     BRN1            ;CHECK OPPOSITE CONDITION
1247  003220  100404                    BMI     BRN2
1248                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1249                                                            ;           CONDITIONAL BRANCH INST. AND   <====
1250                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
1251                                                            ;           WHICH FOLLOWS W/ 773           <====
1252  003222                          BRN1:
1253  003222  012742  000045                MOV     #45,-(R2)       ;MOVE TO MAILBOX #  *******  45  *******
1254  003226  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
1255  003230  000000                          HALT                    ;IMPROPER BR W/ N-1
1256                                          ;CHECK WITH N-BIT OFF
1257  003232  000277                BRN2:     SCC                     ;CC=0111
1258  003234  000250                          CLN
1259  003236  100401                          BMI     BRN3            ;CHECK OPPOSITE CONDITION
1260  003240  100004                          BPL     TS41
1261                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1262                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1263                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1264                                                                  ;           WHICH FOLLOWS W/ 763           <====
1265  003242                          BRN3:
1266  003242  012742  000046                 MOV     #46,-(R2)       ;MOVE TO MAILBOX #  *******  46  *******
1267  003246  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1268  003250  000000                          HALT                    ;IMPROPER BR W/ N=0
1269                                                                  ;  OR SEQUENCE ERROR
1270
1271                                ;*****************************************************************************
1272                                ;
1273                                ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.
1274                                ;THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1275                                ;BVS AND BVC ARE TESTED FOR PROPER EXECUTION.  THEN THE V-BIT IS
1276                                ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1277                                ;AGAIN FOR PROPER OPERATION.
1278                                ;
1279                                ;*****************************************************************************
1280                                ;TEST 41        TEST BRANCHES AROUND V-BIT
1281                                ;*****************************************************************************
1282  003252  005212                TS41:     INC     (R2)            ;UPDATE TEST NUMBER
1283  003254  022712  000041                 CMP     #41,(R2)        ;SEQUENCE ERROR?
1284  003260  001014                          BNE     TS42-10         ;BR TO ERROR HALT ON SEQ ERROR
1285                                          ;FIRST WITH V-BIT ON
1286  003262  000257                          CCC                     ;CC=0010: JUST V-BIT
1287  003264  000262                          SEV
1288  003266  102001                          BVC     BRV1            ;CHECK OPPOSITE CONDITION
1289  003270  102404                          BVS     BRV2
1290                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1291                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1292                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1293                                                                  ;           WHICH FOLLOWS W/ 773           <====
1294  003272                          BRV1:
1295  003272  012742  000047                 MOV     #47,-(R2)       ;MOVE TO MAILBOX #  *******  47  *******
1296  003276  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1297  003300  000000                          HALT                    ;IMPROPER BR W/ V=1
1298                                          ;CHECK WITH V-BIT OFF
1299  003302  000277                BRV2:     SCC                     ;CC=1101: ALL BVT V-BIT
1300  003304  000242                          CLV
1301  003306  102401                          BVS     BRV3            ;CHECK OPPOSITE CONDITION
1302  003310  102004                          BVC     TS42
1303                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1304                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1305                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1306                                                                  ;           WHICH FOLLOWS W/ 763           <====
1307  003312                          BRV3:
1308  003312  012742  000050                 MOV     #50,-(R2)       ;MOVE TO MAILBOX #  *******  50  *******
1309  003316  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1310  003320  000000                          HALT                    ;IMPROPER BR W/ V=0
```

```
1311                                                    ;  OR SEQUENCE ERROR
1312
1313                            ;*************************************************************************************
1314                            ;
1315                            ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.
1316                            ;THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1317                            ;BCS AND BCC ARE TESTED FOR PROPER EXECUTION.  THEN THE C-BIT IS
1318                            ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1319                            ;AGAIN FOR PROPER OPERATION.
1320                            ;
1321                            ;*************************************************************************************
1322                            ;TEST 42        TEST BRANCHES AROUND C-BIT
1323                            ;*************************************************************************************
1324  003322  005212            TS42:   INC     (R2)            ;UPDATE TEST NUMBER
1325  003324  022712  000042            CMP     #42,(R2)        ;SEQUENCE ERROR?
1326  003330  001014            BNE     TS43-10         ;BR TO ERROR HALT ON SEQ ERROR
1327                                    ;FIRST WITH C-BIT ON
1328  003332  000257            CCC                     ;CC=0001: JUST C-BIT
1329  003334  000261            SEC
1330  003336  103001            BCC     BRC1            ;CHECK OPPOSITE CONDITION
1331  003340  103404            BCS     BRC2
1332                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1333                                                    ;           CONDITIONAL BRANCH INST. AND   <====
1334                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
1335                                                    ;           WHICH FOLLOWS W/ 773           <====
1336  003342                    BRC1:
1337  003342  012742  000051            MOV     #51,-(R2)       ;MOVE TO MAILBOX #  *******  51  *******
1338  003346  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1339  003350  000000            HALT                    ;IMPROPER BR W/ C=1
1340                                    ;CHECK WITH C-BIT OFF
1341  003352  000277            BRC2:   SCC             ;CC=1110
1342  003354  000241            CLC
1343  003356  103401            BCS     BRC3            ;CHECK OPPOSITE CONDITION
1344  003360  100404            BMI     TS43
1345                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-- -
1346                                                    ;           CONDITIONAL BRANCH INST. AND   <====
1347                                                    ;           REPLACE THE MOVE INSTRUCTION   <==-=
1348                                                    ;           WHICH FOLLOWS W/ 763           <====
1349  003362                    BRC3:
1350  003362  012742  000052            MOV     #52,-(R2)       ;MOVE TO MAILBOX #  *******  52  *******
1351  003366  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1352  003370  000000            HALT                    ;IMPROPER BR W/ C=0
1353                                                    ;  OR SEQUENCE ERROR
1354
1355                            ;*************************************************************************************
1356                            .SBTTL  MICROCODE TESTS
1357                            ;
1358                            ;       THE TEST EXERCISES  BRANCHES IN THE MICROCODE BY
1359                            ;TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
1360                            ;ALL POSSIBLE MODES.  FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
1361                            ;AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
1362                            ;ADDRESSING MODES.  BYTE MODES ARE ALSO TESTED.  AS EACH NEW
1363                            ;MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
1364                            ;A SMALL LOOP CONVENIENT FOR SCOPING.  THE TEST IS SET UP USING
1365                            ;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
1366                            ;VERIFIED.
```

```
1367                                    ;       IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
1368                                    ;FAULT.
1369                                    ;
1370                                    ;*****************************************************************************
1371
1372
1373
1374                                    ;*****************************************************************************
1375                                    ;
1376                                    ;       THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
1377                                    ;MODE WITH THE SINGLE OPERAND INSTRUCTION.  FOLLOWING THE SEQUENCE CHECK,
1378                                    ;THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
1379                                    ;CHECKS THAT THE Z-BIT WAS PROPERLY SET.  THIS TEST CAN CHECK IR DECODE
1380                                    ;AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0.  FOLLOWING THIS TEST
1381                                    ;SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0.  THESE
1382                                    ;INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
1383                                    ;OF THE SOP INSTRUCTIONS IN THIS TEST.  THE DATA IN THIS TEST IS
1384                                    ;OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
1385                                    ;
1386                                    ;*****************************************************************************
1387                                    ;TEST 43        TEST MODE 0 USING SOP INST.
1388                                    ;*****************************************************************************
1389  003372  005212                    TS43:   INC     (R2)            ;UPDATE TEST NUMBER
1390  003374  022712  000043            CMP     #43,(R2)        ;SEQUENCE ERROR?
1391  003400  001020                    BNE     TS44-10         ;BR TO ERROR HALT ON SEQ ERROR
1392  003402  005000                    CLR     R0              ;TRY THE CLEAR INST.
1393  003404  001404                    BEQ     SOP0A
1394                                                            ;       TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <----
1395                                                            ;                 CONDITIONAL BRANCH INST. AND  <= ==
1396                                                            ;                 REPLACE THE MOVE INSTRUCTION  <-- -
1397                                                            ;                 WHICH FOLLOWS W/ 775          <- --
1398  003406  012742  000053            MOV     #53,-(R2)       ;MOVE TO MAILBOX #  *******  53  *******
1399  003412  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1400  003414  000000                    HALT                    ;CLR DID NOT SET Z-BIT
1401  003416  005200            SOP0A:  INC     R0              ;TRY THE INCREMENT INST.
1402  003420  005100                    COM     R0              ;TRY COMPLEMENT
1403  003422  005200                    INC     R0
1404  003424  100404                    BMI     SOP0B
1405                                                            ;       TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <  -
1406                                                            ;                 CONDITIONAL BRANCH INST. AND  <  --
1407                                                            ;                 REPLACE THE MOVE INSTRUCTION  <---
1408                                                            ;                 WHICH FOLLOWS W/ 765          <  -
1409  003426  012742  000054            MOV     #54,-(R2)       ;MOVE TO MAILBOX #  *******  54  *******
1410  003432  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1411  003434  000000                    HALT                    ;NEGATE DID NOT SET N-BIT
1412  003436  005100            SOP0B:  COM     R0              ;TRY COMPLEMENT INST.
1413  003440  001404                    BEQ     TS44
1414                                                            ;       TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <- ==
1415                                                            ;                 CONDITIONAL BRANCH INST. AND  <  -
1416                                                            ;                 REPLACE THE MOVE INSTRUCTION  <---
1417                                                            ;                 WHICH FOLLOWS W/ 757          <=--
1418  003442  012742  000055            MOV     #55,-(R2)       ;MOVE TO MAILBOX #  *******  55  *******
1419  003446  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1420  003450  000000                    HALT                    ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
1421                                                            ;  OR SEQUENCE ERROR
1422
```

```
1423
1424                            ;*****************************************************************************
1425                            ;
1426                            ;       THIS TEST INTRODUCES THE REMAINING SOP INSTRUCIONS AND TESTS
1427                            ;THEM IN MODE 0.  THE PURPOSE IS TO PROVIDE A BASELINE OF
1428                            ;INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS.  SINCE THE MICROCODE FOR
1429                            ;THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
1430                            ;SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
1431                            ;FUNCTIONING.
1432                            ;
1433                            ;*****************************************************************************
1434                            ;TEST 44        TEST REMAINDER OF SOP INSTS IN MODE 0
1435                            ;*****************************************************************************
1436  003452  005212            TS44:   INC     (R2)            ;UPDATE TEST NUMBER
1437  003454  022712  000044            CMP     #44,(R2)        ;SEQUENCE ERROR?
1438  003460  001021            BNE     TS45-10         ;BR TO ERROR HALT ON SEQ ERROR
1439  003462  005000            CLR     R0              ;INITIALIZE
1440  003464  005300            DEC     R0              ;TRY DECREMENT INST.
1441  003466  100404            BMI     SOPOC
1442                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
1443                                                    ;           CONDITIONAL BRANCH INST. AND   <----
1444                                                    ;           REPLACE THE MOVE INSTRUCTION   <----
1445                                                    ;           WHICH FOLLOWS W/ 774           <----
1446  003470  012742  000056            MOV     #56,-(R2)       ;MOVE TO MAILBOX #  *******  56  *******
1447  003474  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1448  003476  000000            HALT                    ;N-BIT NOT SET ON DEC
1449  003500  000261            SOPOC:  SEC             ;INITIALIZE CARRY
1450  003502  005500            ADC     R0              ;TRY ADD CARRY INST
1451  003504  001007            BNE     SOPOD
1452  003506  000261            SEC                     ;INITIALIZE CARRY
1453  003510  005600            SBC     R0              ;TRY SUBTRACT-CARRY INST
1454  003512  100004            BPL     SOPOD
1455  003514  005100            COM     R0
1456  003516  005200            INC     R0
1457  003520  005300            DEC     R0
1458  003522  001404            BEQ     TS45
1459                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
1460                                                    ;           CONDITIONAL BRANCH INST. AND   <----
1461                                                    ;           REPLACE THE MOVE INSTRUCTION   <----
1462                                                    ;           WHICH FOLLOWS W/ 756           <----
1463  003524                    SOPOD:
1464  003524  012742  000057            MOV     #57,-(R2)       ;MOVE TO MAILBOX #  *******  57  *******
1465  003530  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1466  003532  000000            HALT                    ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F
1467                                                    ;  OR SEQUENCE ERROR
1468
1469                            ;*****************************************************************************
1470                            ;
1471                            ;       THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
1472                            ;THE MODE 0 BYTE MICROCODE IS TESTED.  THE METHOD AND SEQUENCE
1473                            ;OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
1474                            ;
1475                            ;*****************************************************************************
1476                            ;TEST 45        TEST MODE 0 EVEN BYTE USING SOP INST
1477                            ;*****************************************************************************
1478  003534  005212            TS45:   INC     (R2)            ;UPDATE TEST NUMBER
```

```
1479  003536  022712  000045              CMP     #45,(R2)        ;SEQUENCE ERROR?
1480  003542  001012                      BNE     TS46-10         ;BR TO ERROR HALT ON SEQ ERROR
1481  003544  105000                      CLRB    R0              ;TRY CLEARING EVEN BYTE OF REGISTER
1482  003546  001404                      BEQ     SOPB0A
1483                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1484                                                              ;           CONDITIONAL BRANCH INST. AND   <====
1485                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
1486                                                              ;           WHICH FOLLOWS W/ 775           <====
1487  003550  012742  000060              MOV     #60,-(R2)       ;MOVE TO MAILBOX #  *******  60  *******
1488  003554  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1489  003556  000000                      HALT                    ;CLRB DID NOT SET Z-BIT
1490  003560  105100              SOPB0A: COMB    R0              ;TRY SETTING EVEN BYTE OF REGISTER
1491  003562  100002                      BPL     SOPB0B
1492  003564  105200                      INCB    R0              ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
1493  003566  001404                      BEQ     TS46
1494                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1495                                                              ;           CONDITIONAL BRANCH INST. AND   <====
1496                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
1497                                                              ;           WHICH FOLLOWS W/ 765           <====
1498  003570                      SOPB0B:
1499  003570  012742  000061              MOV     #61,-(R2)       ;MOVE TO MAILBOX #  *******  61  *******
1500  003574  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1501  003576  000000                      HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
1502                                                              ;  OR SEQUENCE ERROR
1503
1504                              ;*****************************************************************************************
1505                              ;
1506                              ;       THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
1507                              ;SINGLE OPERAND MODE 1 INSTRUCTIONS.  AGAIN, THE CLR INSTRUCTION
1508                              ;IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
1509                              ;CONDITION CODES ARE SET.  OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
1510                              ;COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
1511                              ;
1512                              ;*****************************************************************************************
1513                              ;TEST 46        TEST MODE 1 USING SOP INST.
1514                              ;*****************************************************************************************
1515  003600  005212              TS46:   INC     (R2)            ;UPDATE TEST NUMBER
1516  003602  022712  000046              CMP     #46,(R2)        ;SEQUENCE ERROR?
1517  003606  001014                      BNE     TS47-10         ;BR TO ERROR HALT ON SEQ ERROR
1518  003610  005000                      CLR     R0              ;INITIALIZE R0
1519  003612  005010                      CLR     (R0)            ;TRY CLEAR INST W/MODE 1
1520  003614  001404                      BEQ     SOP1A
1521                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1522                                                              ;           CONDITIONAL BRANCH INST. AND   <===
1523                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
1524                                                              ;           WHICH FOLLOWS W/ 774           <====
1525  003616  012742  000062              MOV     #62,-(R2)       ;MOVE TO MAILBOX #  *******  62  *******
1526  003622  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1527  003624  000000                      HALT                    ;CLR DID NOT SET Z-BIT
1528  003626  005310              SOP1A:  DEC     (R0)            ;TRY DECREMENT INST W/MODE 1
1529  003630  100003                      BPL     SOP1B
1530  003632  000261                      SEC                     ;INITIALIZE CARRY
1531  003634  005510                      ADC     (R0)            ;TRY ADD-CARRY W/MODE 1
1532  003636  001404                      BEQ     TS47
1533                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1534                                                              ;           CONDITIONAL BRANCH INST. AND   <====
```

```
1535                                                    ;           REPLACE THE MOVE INSTRUCTION   <- ==
1536                                                    ;           WHICH FOLLOWS W/ 763           <==--
1537  003640                            SOP1B:
1538  003640  012742  000063                  MOV     #63,-(R2)       ;MOVE TO MAILBOX #  *******  63  *******
1539  003644  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1540  003646  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST
1541                                                                  ;  OR SEQUENCE ERROR
1542
1543                                    ;*****************************************************************************
1544                                    ;
1545                                    ;       THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
1546                                    ;SINGLE OPERAND INSTRUCTIONS.
1547                                    ;       THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
1548                                    ;AND VERIFIED.
1549                                    ;
1550                                    ;*****************************************************************************
1551                                    ;TEST 47        TEST MODE 1 EVEN BYTE USING SOP INST
1552                                    ;*****************************************************************************
1553  003650  005212                    TS47:   INC     (R2)            ;UPDATE TEST NUMBER
1554  003652  022712  000047                   CMP     #47,(R2)        ;SEQUENCE ERROR?
1555  003656  001020                           BNE     TS50-10         ;BR TO ERROR HALT ON SEQ ERROR
1556  003660  005000                           CLR     R0              ;INITIALIZE R0
1557  003662  005010                           CLR     (R0)            ;INITIALIZE LOC. 0
1558  003664  005110                           COM     (R0)
1559  003666  105010                           CLRB    (R0)            ;TRY TO CLEAR BYTE 0
1560  003670  001404                           BEQ     SOPB1A
1561                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=---
1562                                                                  ;           CONDITIONAL BRANCH INST. AND   <-=--
1563                                                                  ;           REPLACE THE MOVE INSTRUCTION   <==--
1564                                                                  ;           WHICH FOLLOWS W/ 772           <==-
1565  003672  012742  000064                   MOV     #64,-(R2)       ;MOVE TO MAILBOX #  *******  64  *******
1566  003676  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1567  003700  000000                           HALT                    ;CLRB DID NOT SET Z-BIT
1568  003702  005210                    SOPB1A: INC     (R0)            ;INCREMENT TO TEST WORD
1569  003704  100005                           BPL     SOPB1B
1570  003706  105110                           COMB    (R0)            ;COMPLEMENT:  ODD BYTE = 376
1571  003710  10521C                           INCB    (R0)            ;INC:  ODD BYTE = 377
1572  003712  100002                           BPL     SOPB1B
1573  003714  105210                           INCB    (R0)            ;INCREMENT ODD BYTE=0
1574  003716  001404                           BEQ     TS50
1575                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1576                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1577                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1578                                                                  ;           WHICH FOLLOWS W/ 757           <====
1579  003720                            SOPB1B:
1580  003720  012742  000065                   MOV     #65,-(R2)       ;MOVE TO MAILBOX #  *******  65  *******
1581  003724  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1582  003726  000000                           HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST
1583                                                                  ;  OR SEQUENCE ERROR
1584
1585
1586                                    ;*****************************************************************************
1587                                    ;
1588                                    ;       THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL
1589                                    ;FUNCTION CORRECTLY FOR ODD BYTES.
1590                                    ;       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 50         TEST MODE 1 ODD BYTE USING SOP INST
                                ;*****************************************************************************
003730  005212                  TS50:   INC     (R2)            ;UPDATE TEST NUMBER
003732  022712  000050                  CMP     #50,(R2)        ;SEQUENCE ERROR?
003736  001022                          BNE     TS51-10         ;BR TO ERROR HALT ON SEQ ERROR
003740  005000                          CLR     R0              ;INITIALIZE R0
003742  005010                          CLR     (R0)            ;INITIALIZE LOC. 0
003744  005110                          COM     (R0)
003746  005200                          INC     R0              ;R0=ODD BYTE
003750  105010                          CLRB    (R0)            ;TRY TO CLEAR BYTE 1
003752  001404                          BEQ     SOPB1C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
003754  012742  000066                          MOV     #66,-(R2)       ;MOVE TO MAILBOX #  *******  66  *******
003760  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003762  000000                          HALT                    ;CLRB DID NOT SET Z-BIT
003764  005300                  SOPB1C: DEC     R0              ;R0=WORD ADDR.
003766  005210                          INC     (R0)            ;INCREMENT TO TEST WORD
003770  005200                          INC     R0              ;R0=ODD BYTE
003772  105110                          COMB    (R0)            ;TRY TO COMPLEMENT BYTE 1
003774  105210                          INCB    (R0)
003776  100002                          BPL     SOPB1D
004000  105210                          INCB    (R0)            ;TRY TO INCREMENT BYTE 1
004002  001404                          BEQ     TS51
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
                                                                ;           CONDITIONAL BRANCH INST. AND   <-  -
                                                                ;           REPLACE THE MOVE INSTRUCTION   <-=-
                                                                ;           WHICH FOLLOWS W/ 755           <---=
004004                          SOPB1D:
004004  012742  000067                  MOV     #67,-(R2)       ;MOVE TO MAILBOX #  *******  67  *******
004010  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004012  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       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 51         TEST MODE 2 USING SOP INST.
                                ;*****************************************************************************
004014  005212                  TS51:   INC     (R2)            ;UPDATE TEST NUMBER
004016  022712  000051                  CMP     #51,(R2)        ;SEQUENCE ERROR?
```

```
1647  004022  001023                        BNE     TS52-10         ;BR TO ERROR HALT ON SEQ ERROR
1648  004024  005000                        CLR     R0              ;SET R0=400
1649  004026  105100                        COMB    R0
1650  004030  005200                        INC     R0
1651  004032  005010                        CLR     (R0)            ;CLEAR 400
1652  004034  005110                        COM     (R0)            ;INITIALIZE: 400=-1
1653  004036  005020                        CLR     (R0)+           ;TRY CLEARING WITH MODE 2
1654  004040  001404                        BEQ     SOPZA
1655                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1656                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1657                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1658                                                                ;           WHICH FOLLOWS W/ 770           <====
1659  004042  012742  000070                        MOV     #70,-(R2)       ;MOVE TO MAILBOX #  *******  70  *******
1660  004046  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1661  004050  000000                        HALT                    ;CLR INST DID NOT SET Z-BIT
1662  004052  005300                SOPZA:  DEC     R0              ;RESET R0
1663  004054  005300                        DEC     R0
1664  004056  005120                        COM     (R0)+           ;TRY COMPLEMENTING WITH MODE 2
1665  004060  100004                        BPL     SOP2B
1666  004062  005300                        DEC     R0              ;RESET R0
1667  004064  005300                        DEC     R0
1668  004066  005220                        INC     (R0)+           ;TRY INCREMENTING WITH MODE 2
1669  004070  001404                        BEQ     TS52
1670                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1671                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1672                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1673                                                                ;           WHICH FOLLOWS W/ 754           <====
1674  004072                          SOP2B:
1675  004072  012742  000071                        MOV     #71,-(R2)       ;MOVE TO MAILBOX #  *******  71  *******
1676  004076  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1677  004100  000000                        HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST
1678                                                                ;  OR SEQUENCE ERROR
1679
1680                                ;*****************************************************************************
1681                                ;
1682                                ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
1683                                ;ADDRESS EVEN BYTES.  R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION
1684                                ;400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
1685                                ;MODE 2.
1686                                ;       R0 IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS
1687                                ;WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST.  THIS PROCEDURE ALSO
1688                                ;VERIFIES THE PROPER INCREMENTING OF THE REGISTER.
1689                                ;
1690                                ;*****************************************************************************
1691                                ;TEST 52         TEST MODE 2 EVEN BYTE USING SOP INST.
1692                                ;*****************************************************************************
1693  004102  005212                TS52:   INC     (R2)            ;UPDATE TEST NUMBER
1694  004104  022712  000052                CMP     #52,(R2)        ;SEQUENCE ERROR?
1695  004110  001023                        BNE     TS53-10         ;BR TO ERROR HALT ON SEQ ERROR
1696  004112  005000                        CLR     R0              ;SET R0=400
1697  004114  105100                        COMB    R0
1698  004116  005200                        INC     R0
1699  004120  005010                        CLR     (R0)            ;CLEAR 400
1700  004122  005110                        COM     (R0)            ;INITIALIZE: 400=-1
1701  004124  105020                        CLRB    (R0)+           ;TRY TO CLEAT 400 W/MODE 2
1702  004126  001404                        BEQ     SOPB2A
```

```
1703                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1704                                                    ;           CONDITIONAL BRANCH INST. AND   <====
1705                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
1706                                                    ;           WHICH FOLLOWS W/ 770           <====
1707  004130  012742  000072                  MOV     #72,-(R2)       ;MOVE TO MAILBOX #  *******  72  *******
1708  004134  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1709  004136  000000                          HALT                    ;CLR DID NOT SET Z-BIT
1710  004140  005300                SOPB2A:   DEC     R0              ;RESULT R0-400
1711  004142  005210                          INC     (R0)            ;INC 400 TO TEST WORD
1712  004144  105110                          COMB    (R0)
1713  004146  105220                          INCB    (R0)+           ;TRY TO INC EVEN BYTE
1714  004150  100003                          BPL     SOPB2B
1715  004152  005300                          DEC     R0              ;RESET R0=400
1716  004154  105220                          INCB    (R0)+           ;TRY INCREMENT OF EVEN BYTE
1717  004156  001404                          BEQ     TS53
1718                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1719                                                    ;           CONDITIONAL BRANCH INST. AND   <====
1720                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
1721                                                    ;           WHICH FOLLOWS W/ 754           <====
1722  004160                          SOPB2B:
1723  004160  012742  000073                  MOV     #73,-(R2)       ;MOVE TO MAILBOX #  *******  73  *******
1724  004164  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1725  004166  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1726                                                    ;  OR SEQUENCE ERROR
1727
1728                            ;*****************************************************************************
1729                            ;
1730                            ;       THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
1731                            ;TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
1732                            ;
1733                            ;*****************************************************************************
1734                            ;TEST 53        TEST MODE 2 ODD BYTE USING SOP INST.
1735                            ;*****************************************************************************
1736  004170  005212                ;TS53:    INC     (R2)            ;UPDATE TEST NUMBER
1737  004172  022712  000053                  CMP     #53,(R2)        ;SEQUENCE ERROR?
1738  004176  001026                          BNE     TS54-10         ;BR TO ERROR HALT ON SEQ ERROR
1739  004200  005000                          CLR     R0              ;SET R0=400
1740  004202  105100                          COMB    R0
1741  004204  005200                          INC     R0
1742  004206  005010                          CLR     (R0)            ;CLEAR LOC 400
1743  004210  005110                          COM     (R0)            ;INITIALIZE: 400--1
1744  004212  005200                          INC     R0              ;R0=ODD BYTE
1745  004214  105020                          CLRB    (R0)+           ;TRY TO CLEAR ODD BYTE
1746  004216  001404                          BEQ     SOPB2C
1747                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1748                                                    ;           CONDITIONAL BRANCH INST. AND   < - -
1749                                                    ;           REPLACE THE MOVE INSTRUCTION   <- ==
1750                                                    ;           WHICH FOLLOWS W/ 767           <= --
1751  004220  012742  000074                  MOV     #74,-(R2)       ;MOVE TO MAILBOX #  *******  74  *******
1752  004224  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1753  004226  000000                          HALT                    ;CLRB DID NOT SET 7-BIT
1754  004230  005300                SOPB2C:   DEC     R0              ;R0=WORD ADDR.
1755  004232  005300                          DEC     R0
1756  004234  005220                          INC     (R0)+           ;INCREMENT WORD
1757  004236  005300                          DEC     R0              ;POINT TO ODD BYTE
1758  004240  105110                          COMB    (R0)            ;COMPLEMENT ODD BYTE
```

```
1759  004242  105220                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
1760  004244  100003                  BPL     SOPB2D
1761  004246  005300                  DEC     R0              ;RESET R0 TO ODD BYTE
1762  004250  105220                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTF
1763  004252  001404                  BEQ     TS54
1764                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < -
1765                                                          ;           CONDITIONAL BRANCH INST. AND   <   -
1766                                                          ;           REPLACE THE MOVE INSTRUCTION   <   -
1767                                                          ;           WHICH FOLLOWS W/ 751           <-
1768  004254                  SOPB2D:
1769  004254  012742  000075          MOV     #75,-(R2)       ;MOVE TO MAILBOX #  *******  75  *******
1770  004260  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1771  004262  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1772                                                          ;  OR SEQUENCE ERROR
1773
1774                          ;*****************************************************************************
1775                          ;
1776                          ;       THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES.  PREVIOUSLY
1777                          ;TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
1778                          ;
1779                          ;*****************************************************************************
1780                          ;TEST 54        TEST MODE 0 USING NEGATE INSTRUCTION
1781                          ;*****************************************************************************
1782  004264  005212          TS54:   INC     (R2)            ;UPDATE TEST NUMBER
1783  004266  022712  000054          CMP     #54,(R2)        ;SEQUENCE ERROR?
1784  004272  001035                  BNE     TS55-10         ;BR TO ERROR HALT ON SEQ ERROR
1785  004274  005000                  CLR     R0              ;SET R0=0
1786  004276  005200                  INC     R0              ;    R0=1
1787  004300  005400                  NEG     R0              ;TRY NEGATE MODE 0:  R0 -1
1788  004302  100003                  BPL     NEG00           ;CC=1001?
1789  004304  001402                  BEQ     NEG00
1790  004306  102401                  BVS     NEG00
1791  004310  103404                  BCS     NEG01
1792                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1793                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1794                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1795                                                          ;           WHICH FOLLOWS W/ 770           <====
1796  004312                  NEG00:
1797  004312  012742  000076          MOV     #76,-(R2)       ;MOVE TO MAILBOX #  *******  76  *******
1798  004316  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1799  004320  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1800
1801  004322  005200          NEG01:  INC     R0              ;TEST DATA RESULT
1802  004324  001404                  BEQ     NEG02
1803                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1804                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1805                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1806                                                          ;           WHICH FOLLOWS W/ 762           <====
1807  004326  012742  000077          MOV     #77,-(R2)       ;MOVE TO MAILBOX #  *******  77  *******
1808  004332  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1809  004334  000000                  HALT                    ;DATA RESULT OF NEGATE INCORRECT
1810
1811  004336  105100          NEG02:  COMB    R0              ;R0=377
1812  004340  105400                  NEGB    R0              ;R0=1
1813  004342  100403                  BMI     NEG03           ;CC=0001?
1814  004344  001402                  BEQ     NEG03
```

```
1815  004346  102401                          BVS     NEG03
1816  004350  103404                          BCS     NEG04
1817                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===
1818                                                          ;           CONDITIONAL BRANCH INST. AND   <=-==
1819                                                          ;           REPLACE THE MOVE INSTRUCTION   <  --
1820                                                          ;           WHICH FOLLOWS W/ 750           <-=-=
1821  004352                            NEG03:
1822  004352  012742  000100                      MOV     #100,-(R2)      ;MOVE TO MAILBOX #  *******  100  *******
1823  004356  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1824  004360  000000                      HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
1825  004362  005300                    NEG04:  DEC     R0              ;TEST DATA RESULT
1826  004364  001404                      BEQ     TS55
1827                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < ==-
1828                                                          ;           CONDITIONAL BRANCH INST. AND   <
1829                                                          ;           REPLACE THE MOVE INSTRUCTION   <
1830                                                          ;           WHICH FOLLOWS W/ 742           <  -
1831  004366  012742  000101                      MOV     #101,-(R2)      ;MOVE TO MAILBOX #  *******  101  *******
1832  004372  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1833  004374  000000                      HALT                    ;DATA RESULT OF NEGB INCORRECT
1834                                                          ;  OR SEQUENCE ERROR
1835                                    ;*****************************************************************************
1836                                    ;TEST 55        TEST MODE 1 USING NEGATE INST.
1837                                    ;*****************************************************************************
1838  004376  005212                    TS55:   INC     (R2)            ;UPDATE TEST NUMBER
1839  004400  022712  000055                      CMP     #55,(R2)        ;SEQUENCE ERROR?
1840  004404  001040                      BNE     TS56-10         ;BR TO ERROR HALT ON SEQ ERROR
1841  004406  005000                      CLR     R0              ;POINT TO LOC. 0
1842  004410  005010                      CLR     (R0)            ;CLEAR LOC. 0
1843  004412  005210                      INC     (R0)            ;LOC. 0=1
1844  004414  005410                      NEG     (R0)            ;TRY NEG. LOC. 0=-1
1845  004416  100003                      BPL     NEG10           ;CC=1001
1846  004420  001402                      BEQ     NEG10
1847  004422  102401                      BVS     NEG10
1848  004424  103404                      BCS     NEG11
1849                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < - -
1850                                                          ;           CONDITIONAL BRANCH INST. AND   <==-
1851                                                          ;           REPLACE THE MOVE INSTRUCTION   <- ==
1852                                                          ;           WHICH FOLLOWS W/ 767           <---
1853  004426                            NEG10:
1854  004426  012742  000102                      MOV     #102,-(R2)      ;MOVE TO MAILBOX #  *******  102  *******
1855  004432  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1856  004434  000000                      HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1857
1858  004436  005237  000000            NEG11:  INC     @#0             ;TEST DATA RESULT
1859  004442  001404                      BEQ     NEG12
1860                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1861                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1862                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1863                                                          ;           WHICH FOLLOWS W/ 760           <====
1864  004444  012742  000103                      MOV     #103,-(R2)      ;MOVE TO MAILBOX #  *******  103  *******
1865  004450  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1866  004452  000000                      HALT                    ;DATA RESULT OF NEGATE INCORRECT
1867  004454  105110                    NEG12:  COMB    (R0)            ;LOC. 0=377
1868  004456  105410                      NEGB    (R0)            ;TRY NEGB LOC. 0=1
1869  004460  100403                      BMI     NEG13           ;CC=0001?
1870  004462  001402                      BEQ     NEG13
```

```
1871  004464  102401                  BVS     NEG13
1872  004466  103404                  BCS     NEG14
1873                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
1874                                                ;           CONDITIONAL BRANCH INST. AND   <====
1875                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1876                                                ;           WHICH FOLLOWS W/ 746           <-==-
1877  004470                  NEG13:
1878  004470  012742  000104          MOV     #104,-(R2)      ;MOVE TO MAILBOX #  *******  104  *******
1879  004474  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1880  004476  000000                  HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
1881  004500  005337  000000  NEG14:  DEC     @#0             ;TEST DATA RESULT
1882  004504  001404                  BEQ     TS56
1883                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1884                                                ;           CONDITIONAL BRANCH INST. AND   <=-==
1885                                                ;           REPLACE THE MOVE INSTRUCTION   <--==
1886                                                ;           WHICH FOLLOWS W/ 737           <=-==
1887  004506  012742  000105          MOV     #105,-(R2)      ;MOVE TO MAILBOX #  *******  105  *******
1888  004512  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1889  004514  000000                  HALT                    ;DATA RESULT OF NEGB INCORRECT
1890                                                ;  OR SEQUENCE ERROR
1891  ;*****************************************************************************************
1892  ;TEST 56         TEST MODE 2 USING NEGATE INSTRUCTION
1893  ;*****************************************************************************************
1894  004516  005212  TS56:   INC     (R2)            ;UPDATE TEST NUMBER
1895  004520  022712  000056          CMP     #56,(R2)        ;SEQUENCE ERROR?
1896  004524  001032                  BNE     TS57-10         ;BR TO ERROR HALT ON SEQ ERROR
1897  004526  005000                  CLR     R0              ;POINT TO LOC. 0
1898  004530  005010                  CLR     (R0)            ;CLEAR LOC. 0
1899  004532  005210                  INC     (R0)            ;LOC. 0=1
1900  004534  005420                  NEG     (R0)+           ;TRY NEG.: LOC. 0--1
1901  004536  100003                  BPL     NEG20           ;CC=1001?
1902  004540  001402                  BEQ     NEG20
1903  004542  102401                  BVS     NEG20
1904  004544  103404                  BCS     NEG21
1905                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1906                                                ;           CONDITIONAL BRANCH INST. AND   <====
1907                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1908                                                ;           WHICH FOLLOWS W/ 767           <====
1909  004546                  NEG20:
1910  004546  012742  000106          MOV     #106,-(R2)      ;MOVE TO MAILBOX #  *******  106  *******
1911  004552  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1912  004554  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1913  004556  105300  NEG21:  DECB    R0              ;R0=LOC. 0
1914  004560  105300          DECB    R0
1915  004562  105420          NEGB    (R0)+           ;BYTE 0=1   R0=1
1916  004564  105420          NEGB    (R0)+           ;BYTE 1=1   R0=2
1917  004566  105340          DECB    -(R0)           ;R0=1  LOC. 0=01
1918  004570  005300          DEC     R0              ;R0=0
1919  004572  001404          BEQ     NEG22
1920                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1921                                                ;           CONDITIONAL BRANCH INST. AND   <====
1922                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1923                                                ;           WHICH FOLLOWS W/ 754           <====
1924  004574  012742  000107          MOV     #107,-(R2)      ;MOVE TO MAILBOX #  *******  107  *******
1925  004600  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1926  004602  000000                  HALT                    ;REGISTER NOT INCREMENTED CORRECTLY
```

```
1927  004604  005337  000000          NEG22:  DEC     @#0             ;LOC. 0=0
1928  004610  001404                          BEQ     TS57
1929                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1930                                          ;           CONDITIONAL BRANCH INST. AND   <====
1931                                          ;           REPLACE THE MOVE INSTRUCTION   <-==-
1932                                          ;           WHICH FOLLOWS W/ 745           <====
1933  004612  012742  000110                  MOV     #110,-(R2)      ;MOVE TO MAILBOX #  *******  110  *******
1934  004616  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1935  004620  000000                          HALT                    ;NEG BYTE INSTRUCTIONS FAILED
1936                                                                  ;  OR SEQUENCE ERROR
1937
1938                                  ;*****************************************************************************
1939                                  ;
1940                                  ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS.  IT
1941                                  ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 400
1942                                  ;THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
1943                                  ;INSTRUCTIONS UNDER TEST.
1944                                  ;       R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
1945                                  ;INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0.  THEN R0
1946                                  ;IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
1947                                  ;LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST.  THE PROPER INCREMENTING
1948                                  ;OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
1949                                  ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
1950                                  ;(LOC. 400-402) HAS THE PROPER VALUES (0).
1951                                  ;
1952                                  ;*****************************************************************************
1953                                  ;TEST 57        TEST MODE 3 USING SOP INST.
1954                                  ;*****************************************************************************
1955  004622  005212                  TS57:   INC     (R2)            ;UPDATE TEST NUMBER
1956  004624  022712  000057                  CMP     #57,(R2)        ;SEQUENCE ERROR?
1957  004630  001020                          BNE     TS60-10         ;BR TO ERROR HALT ON SEQ ERROR
1958  004632  005000                          CLR     R0              ;SET R0=400
1959  004634  105100                          COMB    R0
1960  004636  005200                          INC     R0
1961  004640  005010                          CLR     (R0)            ;CLEAR LOC 400
1962  004642  005030                          CLR     @(R0)+          ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402
1963  004644  001404                          BEQ     SOP3A
1964                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1965                                          ;           CONDITIONAL BRANCH INST. AND   <====
1966                                          ;           REPLACE THE MOVE INSTRUCTION   <-==-
1967                                          ;           WHICH FOLLOWS W/ 771           <====
1968  004646  012742  000111                  MOV     #111,-(R2)      ;MOVE TO MAILBOX #  *******  111  *******
1969  004652  005242                          INC     -(P2)           ;SET MSGTYP TO FATAL ERROR
1970  004654  000000                          HALT                    ;CLR DID NOT SET Z-BIT
1971  004656  005300                  SOP3A:  DEC     R0              ;RESET R0=400
1972  004660  005300                          DEC     R0
1973  004662  005130                          COM     @(R0)+          ;TRY TO COMPLEMENT LOC 0 OF MODE 3  ;R0=402
1974  004664  100002                          BPL     SOP3B
1975  004666  005230                          INC     @(R0)+          ;TRY TO INCREMENT LOC 0 W/MODE 3  ;R0=404
1976  004670  001404                          BEQ     TS60
1977                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1978                                          ;           CONDITIONAL BRANCH INST. AND   <=-==
1979                                          ;           REPLACE THE MOVE INSTRUCTION   <===-
1980                                          ;           WHICH FOLLOWS W/ 757           <====
1981  004672                          SOP3B:
1982  004672  012742  000112                  MOV     #112,-(R2)      ;MOVE TO MAILBOX #  *******  112  *******
```

```
1983  004676  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1984  004700  000000                          HALT                    ;CUMMULATIVE RESULT OF ABOVE INST FAILED
1985                                                                  ;  OR SEQUENCE ERROR
1986
1987                                  ;*****************************************************************************
1988                                  ;
1989                                  ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
1990                                  ;WHICH ADDRESS EVEN BYTES.  AGAIN, THE TARGET LOCATION 0 IS USED
1991                                  ;AND THE SAME TABLE AT 400 IS EMPLOYED.
1992                                  ;       AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION
1993                                  ;0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.
1994                                  ;       SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE
1995                                  ;TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.
1996                                  ;IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS
1997                                  ;THE PROPER VALUES (0).
1998                                  ;
1999                                  ;*****************************************************************************
2000                                  ;TEST 60        TEST MODE 3 EVEN BYTE USING SOP INST.
2001                                  ;*****************************************************************************
2002  004702  005212                  TS60:   INC     (R2)            ;UPDATE TEST NUMBER
2003  004704  022712  000060                  CMP     #60,(R2)        ;SEQUENCE ERROR?
2004  004710  001026                          BNE     TS61-10         ;BR TO ERROR HALT ON SEQ ERROR
2005  004712  005004                          CLR     R4              ;SET R4=400
2006  004714  105104                          COMB    R4
2007  004716  005204                          INC     R4
2008  004720  005000                          CLR     R0              ;INITIALIZE LOC. 0=-1
2009  004722  005010                          CLR     (R0)
2010  004724  005110                          COM     (R0)            ;LOC. 0=-1
2011  004726  105034                          CLRB    @(R4)+          ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400  R4=402
2012  004730  001404                          BEQ     SOPB3A
2013                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2014                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2015                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2016                                                                  ;           WHICH FOLLOWS W/ 767           <====
2017  004732  012742  000113                          MOV     #113,-(R2)      ;MOVE TO MAILBOX #  *******  113  *******
2018  004736  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2019  004740  000000                          HALT                    ;CLRB DID NOT SET Z-BIT
2020  004742  005304                  SOPB3A: DEC     R4              ;RESET POINTER  R4=400
2021  004744  005304                          DEC     R4
2022  004746  005234                          INC     @(R4)+          ;TRY INCREMENTING WORD  LOC.0=177401  R4=402
2023  004750  100006                          BPL     SOPB3B
2024  004752  105434                          NEGB    @(R4)+          ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
2025  004754  100004                          BPL     SOPB3B
2026  004756  005304                          DEC     R4              ;R4=402
2027  004760  005304                          DEC     R4
2028  004762  105234                          INCB    @(R4)+          ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
2029  004764  001404                          BEQ     TS61
2030                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2031                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2032                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2033                                                                  ;           WHICH FOLLOWS W/ 751           <====
2034  004766                          SOPB3B:
2035  004766  012742  000114                  MOV     #114,-(R2)      ;MOVE TO MAILBOX #  *******  114  *******
2036  004772  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2037  004774  000000                          HALT                    ;CUMMULATIVE RESULT OF ABOVE INST FAILED
2038                                                                  ;  OR SEQUENCE ERROR
```

```
2039
2040                            ;*****************************************************************************************
2041                            ;
2042                            ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
2043                            ;WHICH ADDRESS ODD BYTES.  THE TARGET IS BYTE 1.  A TABLE AT
2044                            ;LOC. 400-406 IS USED.  R0 SERVES AS THE TABLE POINTER.
2045                            ;       R0 IS INITIALIZED TO 400.  LOC. 0 IS SET TO -1 USING THE
2046                            ;FIRST TWO TABLE ENTRIES.  A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING
2047                            ;TABLE ADDRESS AT 404.  R0 IS DECREMENTED TO 402 AND SEVERAL SOP
2048                            ;MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER
2049                            ;REGISTER INCREMENTING.
2050                            ;       THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND
2051                            ;AFTER THE TEST IS RUN.
2052                            ;
2053                            ;*****************************************************************************************
2054                            ;TEST 61        TEST MODE 3 ODD BYTE USING SOP INST.
2055                            ;*****************************************************************************************
2056  004776  005212            TS61:   INC     (R2)            ;UPDATE TEST NUMBER
2057  005000  022712  000061            CMP     #61,(R2)        ;SEQUENCE ERROR?
2058  005004  001024                    BNE     TS62-10         ;BR TO ERROR HALT ON SEQ ERROR
2059  005006  005000                    CLR     R0              ;SET R0=400
2060  005010  105100                    COMB    R0
2061  005012  005200                    INC     R0
2062  005014  005030                    CLR     @(R0)+          ;INITIALIZE
2063  005016  005130                    COM     @(R0)+          ;LOC 0=-1 R0=404
2064  005020  105030                    CLRB    @(R0)+          ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406
2065  005022  001404                    BEQ     SOPB3C
2066                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-=-=
2067                                                            ;           CONDITIONAL BRANCH INST. AND   <==-=
2068                                                            ;           REPLACE THE MOVE INSTRUCTION   <==-=
2069                                                            ;           WHICH FOLLOWS W/ 770           <====
2070  005024  012742  000115            MOV     #115,-(R2)      ;MOVE TO MAILBOX #  *******  115  *******
2071  005030  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2072  005032  000000                    HALT                    ;CLRB DID NOT SET Z-BIT
2073  005034  005300            SOPB3C: DEC     R0              ;RESET R0=402
2074  005036  005300                    DEC     R0
2075  005040  005300                    DEC     R0              ;POINT TO EVEN BYTE ADDR.
2076  005042  005300                    DEC     R0
2077  005044  005230                    INC     @(R0)+          ;INCREMENT WORD LOC. 0=400 R0=404
2078  005046  105430                    NEGB    @(R0)+          ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406
2079  005050  100002                    BPL     SOPB3D
2080  005052  105230                    INCB    @(R0)+          ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410
2081  005054  001404                    BEQ     TS62
2082                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2083                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2084                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2085                                                            ;           WHICH FOLLOWS W/ 753           <====
2086  005056                    SOPB3D:
2087  005056  012742  000116            MOV     #116,-(R2)      ;MOVE TO MAILBOX #  *******  116  *******
2088  005062  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2089  005064  000000                    HALT                    ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED
2090                                                            ;  OR SEQUENCE ERROR
2091                            ;*****************************************************************************************
2092                            ;TEST 62        TEST MODE 3 USING NEGATE INSTRUCTION
2093                            ;*****************************************************************************************
2094  005066  005212            TS62:   INC     (R2)            ;UPDATE TEST NUMBER
```

```
2095  005070  022712  000062            CMP     #62,(R2)        ;SEQUENCE ERROR?
2096  005074  001054                    BNE     TS63-10         ;BR TO ERROR HALT ON SEQ ERROR
2097  005076  005000                    CLR     R0              ;R0=400
2098  005100  105100                    COMB    R0
2099  005102  005200                    INC     R0
2100  005104  005010                    CLR     (R0)            ;LOC. 400=0
2101  005106  005004                    CLR     R4              ;R4=0
2102  005110  005014                    CLR     (R4)            ;LOC. 0-0
2103  005112  005214                    INC     (R4)            ;LOC. 0=
2104  005114  005430                    NEG     @(R0)+          ;TRY NEGATE   LOC. 0=-1  U-402
2105  005116  100003                    BPL     NEG30           ;CC=1001?
2106  005120  001402                    BEQ     NEG30
2107  005122  102401                    BVS     NEG30
2108  005124  103404                    BCS     NEG31
2109                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2110                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2111                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2112                                                            ;           WHICH FOLLOWS W/ 763           <====
2113  005126                    NEG30:
2114  005126  012742  000117            MOV     #117,-(R2)      ;MOVE TO MAILBOX #  *******  117  *******
2115  005132  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2116  005134  000000                    HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2117  005136  005214            NEG31:  INC     (R4)            ;LOC. 0=0
2118  005140  001404                    BEQ     NEG32
2119                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2120                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2121                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2122                                                            ;           WHICH FOLLOWS W/ 755           <====
2123  005142  012742  000120            MOV     #120,-(R2)      ;MOVE TO MAILBOX #  *******  120  *******
2124  005146  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2125  005150  000000                    HALT                    ;DATA RESULT OF NEG INCORRECT
2126  005152  105137  000001    NEG32:  COMB    @#1             ;LOC 0=177400
2127  005156  005237  000000            INC     @#0             ;LOC. 0=177401
2128  005162  105430                    NEGB    @(R0)+          ;TRY NEGB LOC. 0=177777  R0=404
2129  005164  100404                    BMI     NEG33
2130                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2131                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2132                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2133                                                            ;           WHICH FOLLOWS W/ 743           <====
2134  005166  012742  000121            MOV     #121,-(R2)      ;MOVE TO MAILBOX #  *******  121  *******
2135  005172  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2136  005174  000000                    HALT                    ;NEGB FAILED WITH EVEN BYTE
2137  005176  105430            NEG33:  NEGB    @(R0)+          ;TRY NEGB  LOC.0-777  R0=406
2138  005200  100004                    BPL     NEG34
2139                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2140                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2141                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2142                                                            ;           WHICH FOLLOWS W/ 735           <====
2143  005202  012742  000122            MOV     #122,-(R2)      ;MOVE TO MAILBOX #  *******  122  *******
2144  005206  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2145  005210  000000                    HALT                    ;NEGB FAILED WITH ODD BYTE
2146  005212  105137  000001    NEG34:  COMB    @#1             ;LOC. 0=177377
2147  005216  105237  000001            INCB    @#1             ;LOC. 0=177777
2148  005222  005214                    INC     (R4)            ;LOC. 0=0
2149  005224  001404                    BEQ     TS63
2150                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 723           <====
005226  012742  000123       MOV     #123,-(R2)    ;MOVE TO MAILBOX #  *******  123  *******
005232  005242                INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
005234  000000                HALT                  ;DATA RESULT OF NEGB'S INCORRECT
                                                    ;  OR SEQUENCE ERROR

;*****************************************************************************************
;
;       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 63        TEST MODE 4 USING SOP INSTS
;*****************************************************************************************
005236  005212  TS63:   INC     (R2)          ;UPDATE TEST NUMBER
005240  022712  000063   CMP     #63,(R2)      ;SEQUENCE ERROR?
005244  001021          BNE     TS64-10       ;BR TO ERROR HALT ON SEQ ERROR
005246  005000          CLR     R0            ;SET R0=400
005250  105100          COMB    R0
005252  005200          INC     R0
005254  005040          CLR     -(R0)         ;TRY TO CLEAR USING MODE 4
005256  001404          BEQ     SOP4A
                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                              ;           WHICH FOLLOWS W/ 772           <====
005260  012742  000124       MOV     #124,-(R2)    ;MOVE TO MAILBOX #  *******  124  *******
005264  005242                INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
005266  000000                HALT                  ;CLR DID NOT SET Z-BIT
005270  005200  SOP4A:  INC     R0            ;RESET R0
005272  005200                INC     R0
005274  005140                COM     -(R0)         ;TRY TO COMPLEMENT USING MODE 4
005276  100004                BPL     SOP4B
005300  005200                INC     R0            ;MOVE POINTER
005302  005200                INC     R0
005304  005240                INC     -(R0)
005306  001404                BEQ     TS64
                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                              ;           WHICH FOLLOWS W/ 756           <====
005310  SOP4B:
005310  012742  000125       MOV     #125,-(R2)    ;MOVE TO MAILBOX #  *******  125  *******
005314  005242                INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
005316  000000                HALT                  ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
                                                    ;  OR SEQUENCE ERROR

;*****************************************************************************************
;
;       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 64        TEST MODE 5 USING SOP INSTS
                        ;*****************************************************************************************
005320  005212          TS64:   INC     (R2)            ;UPDATE TEST NUMBER
005322  022712  000064          CMP     #64,(R2)        ;SEQUENCE ERROR?
005326  001025                  BNE     TS65-10         ;BR TO ERROR HALT ON SEQ ERROR
005330  012700  000370          MOV     #370,R0         ;CLEAR LOCATION 370-376
005334  005020                  CLR     (R0)+           ;370
005336  005020                  CLR     (R0)+           ;372
005340  005020                  CLR     (R0)+           ;374
005342  005010                  CLR     (R0)            ;376
005344  005000                  CLR     R0              ;SET R0=376 (LOW BYTE)
005346  005020                  CLR     (R0)+
005350  105400                  NEGB    R0
005352  005050                  CLR     @-(R0)          ;TRY TO CLEAR LOC 0 W/MODE 5
005354  001404                  BEQ     SOP5A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
005356  012742  000126          MOV     #126,-(R2)      ;MOVE TO MAILBOX #  *******  126  *******
005362  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005364  000000                  HALT                    ;CLR DID NOT SET Z-BIT
005366  005200          SOP5A:  INC     R0              ;RESET R0
005370  005200                  INC     R0
005372  005150                  COM     @-(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
005374  100002                  BPL     SOP5B
005376  005250                  INC     @-(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 5
005400  001404                  BEQ     TS65
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
005402          SOP5B:
005402  012742  000127          MOV     #127,-(R2)      ;MOVE TO MAILBOX #  *******  127  *******
005406  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005410  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************************
                        ;
                        ;       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 65         TEST MODE 6 USING SOP INSTS
                                ;*********************************************************************************************
005412  005212                  TS65:   INC     (R2)            ;UPDATE TEST NUMBER
005414  022712  000065                  CMP     #65,(R2)        ;SEQUENCE ERROR?
005420  001020                          BNE     TS66-10         ;BR TO ERROR HALT ON SEQ ERROR
005422  005000                          CLR     R0              ;SET R0=400
005424  105100                          COMB    R0
005426  005200                          INC     R0
005430  005060  177400                  CLR     -400(R0)        ;TRY TO CLEAR LOCATION 0 W/MODE 6
005434  001404                          BEQ     SOP6A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
005436  012742  000130                  MOV     #130,-(R2)      ;MOVE TO MAILBOX #  *******  130  *******
005442  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005444  000000                          HALT                    ;CLR DID NOT SET Z-BIT
005446  005160  177400          SOP6A:  COM     -400(R0)        ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
005452  100003                          BPL     SOP6B
005454  005260  177400                  INC     -400(R0)        ;TRY TO INCREMENT LOCATION 0 W/MODE 6
005460  001404                          BEQ     TS66
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
005462                  SOP6B:
005462  012742  000131                  MOV     #131,-(R2)      ;MOVE TO MAILBOX #  *******  131  *******
005466  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005470  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
                                                                ;  OR SEQUENCE ERROR

                                ;*********************************************************************************************
                                ;
                                ;       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 66         TEST MODE 7 USING SOP INST.
                                ;*********************************************************************************************
005472  005212                  TS66:   INC     (R2)            ;UPDATE TEST NUMBER
005474  022712  000066                  CMP     #66,(R2)        ;SEQUENCE ERROR?
005500  001021                          BNE     TS67-10         ;BR TO ERROR HALT ON SEQ ERROR
005502  005000                          CLR     R0              ;SET R0=400
005504  105100                          COMB    R0
005506  005200                          INC     R0
005510  005210                          INC     (R0)            ;R0=1
005512  005070  000002                  CLR     @2(R0)          ;TRY TO CLEAR LOC. 0 W/MODE 7
005516  001404                          BEQ     SOP7A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
2319                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2320                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2321                                                    ;           WHICH FOLLOWS W/ 770           <====
2322  005520  012742  000132          MOV     #132,-(R2)      ;MOVE TO MAILBOX #  *******  132  *******
2323  005524  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2324  005526  000000                  HALT                    ;CLR DID NOT SET Z-BIT
2325  005530  005170  000002  SOP7A:  COM     @2(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
2326  005534  100003                  BPL     SOP7B
2327  005536  005270  000002          INC     @2(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 7
2328  005542  001404                  BEQ     TS67
2329                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2330                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2331                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2332                                                    ;           WHICH FOLLOWS W/ 756           <====
2333  005544          SOP7B:
2334  005544  012742  000133          MOV     #133,-(R2)      ;MOVE TO MAILBOX #  *******  133  *******
2335  005550  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2336  005552  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
2337                                                    ;  OR SEQUENCE ERROR
2338
2339                    ;*****************************************************************************************
2340                    ;TEST 67        TEST MODE 4 WITH NEGATE INSTRUCTION
2341                    ;*****************************************************************************************
2342  005554  005212    TS67:   INC     (R2)            ;UPDATE TEST NUMBER
2343  005556  022712  000067          CMP     #67,(R2)        ;SEQUENCE ERROR?
2344  005562  001024                  BNE     TS70-10         ;BR TO ERROR HALT ON SEQ ERROR
2345  005564  005000                  CLR     R0
2346  005566  005010                  CLR     (R0)
2347  005570  005120                  COM     (R0)+           ;LOC. 0=177777,  R0-2
2348  005572  005440                  NEG     -(R0)           ;TRY NEGATE, LOC. 0=1
2349  005574  100403                  BMI     NEG40           ;CC=0001?
2350  005576  001402                  BEQ     NEG40
2351  005600  102401                  BVS     NEG40
2352  005602  103404                  BCS     NEG41
2353                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2354                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2355                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2356                                                    ;           WHICH FOLLOWS W/ 767           <====
2357  005604          NEG40:
2358  005604  012742  000134          MOV     #134,-(R2)      ;MOVE TO MAILBOX #  *******  134  *******
2359  005610  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2360  005612  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2361  005614  005400  NEG41:  NEG     R0              ;TST R0 WITH A NEG.
2362  005616  001404                  BEQ     NEG42
2363                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2364                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2365                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2366                                                    ;           WHICH FOLLOWS W/ 761           <====
2367  005620  012742  000135          MOV     #135,-(R2)      ;MOVE TO MAILBOX #  *******  135  *******
2368  005624  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2369  005626  000000                  HALT                    ;R0 NOT DECREMENTED PROPERLY
2370  005630  005310  NEG42:  DEC     (R0)            ;TEST DTA RESULT OF NEG
2371  005632  001404                  BEQ     TS70
2372                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2373                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2374                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                        ;            WHICH FOLLOWS W/ 753          <=---=
005634  012742  000136                MOV     #136,-(R2)      ;MOVE TO MAILBOX #  *******  136  *******
005640  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005642  000000                        HALT                    ;DATA RESULT OF NEG INCORRECT
                                                        ;  OR SEQUENCE ERROR
                ;*******************************************************************************************
                ;TEST 70         TEST MODE 5 WITH NEGATE INSTRUCTION
                ;*******************************************************************************************
005644  005212          TS70:   INC     (R2)            ;UPDATE TEST NUMBER
005646  022712  000070          CMP     #70,(R2)        ;SEQUENCE ERROR?
005652  001031                  BNE     TS71-10         ;BR TO ERROR HALT ON SEQ ERROR
005654  005000                  CLR     R0              ;R0=0
005656  005010                  CLR     (R0)            ;LOC. 0=0
005660  105100                  COMB    R0              ;R0=377
005662  005200                  INC     R0              ;R0=400
005664  005010                  CLR     (R0)            ;SET 400 = 0
005666  005004                  CLR     R4              ;R4=0
005670  005314                  DEC     (R4)            ;LOC. 0=177777
005672  005450                  NEG     @-(R0)          ;TRY NEGATE:  LOC. 0=1
005674  100403                  BMI     NEG50           ;CC=0001?
005676  001402                  BEQ     NEG50
005700  102401                  BVS     NEG50
005702  103404                  BCS     NEG51
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
005704          NEG50:
005704  012742  000137                MOV     #137,-(R2)      ;MOVE TO MAILBOX #  *******  137  *******
005710  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005712  000000                        HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005714  005314          NEG51:  DEC     (R4)
005716  001404                  BEQ     NEG52
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 755           <====
005720  012742  000140                MOV     #140,-(R2)      ;MOVE TO MAILBOX #  *******  140  *******
005724  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005726  000000                        HALT                    ;DATA RESULT OF NEG INCORRECT
005730  105100          NEG52:  COMB    R0
005732  005300                  DEC     R0
005734  001404                  BEQ     TS71
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <=--=
                                                        ;           WHICH FOLLOWS W/ 746           <= -=
005736  012742  000141                MOV     #141,-(R2)      ;MOVE TO MAILBOX #  *******  141  *******
005742  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005744  000000                        HALT                    ;REGISTER NOT DECREMENTED PROPERLY
                                                        ;  OR SEQUENCE ERROR
                ;*******************************************************************************************
                ;TEST 71         TEST MODE 6 WITH NEGATE
                ;*******************************************************************************************
005746  005212          TS71:   INC     (R2)            ;UPDATE TEST NUMBER
005750  022712  000071          CMP     #71,(R2)        ;SEQUENCE ERROR?
```

```
2431  005754  001022                    BNE     TS72-10         ;BR TO ERROR HALT ON SEQ ERROR
2432  005756  005000                    CLR     R0              ;R0=0
2433  005760  005004                    CLR     R4              ;R4=0
2434  005762  105100                    COMB    R0              ;R0=377
2435  005764  005014                    CLR     (R4)            ;LOC. 0=0
2436  005766  105024                    CLRB    (R4)+           ;LOC. 0=177777,  R4=1
2437  005770  105114                    COMB    (R4)            ;LOC. 0=177400
2438  005772  005460  177401            NEG     -377(R0)        ;LOC. 0-400
2439  005776  100403                    BMI     NEG60           ;CC=0001
2440  006000  001402                    BEQ     NEG60
2441  006002  102401                    BVS     NEG60
2442  006004  103404                    BCS     NEG61
2443                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2444                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2445                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2446                                                            ;           WHICH FOLLOWS W/ 763           <====
2447  006006                    NEG60:
2448  006006  012742  000142            MOV     #142,-(R2)      ;MOVE TO MAILBOX #  *******  142  *******
2449  006012  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2450  006014  000000                    HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2451  006016  105314            NEG61:  DECB    (R4)
2452  006020  001404                    BEQ     TS72
2453                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2454                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2455                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2456                                                            ;           WHICH FOLLOWS W/ 755           <====
2457  006022  012742  000143            MOV     #143,-(R2)      ;MOVE TO MAILBOX #  *******  143  *******
2458  006026  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2459  006030  000000                    HALT                    ;DATA RESULT OF NEG INCORRECT
2460                                                            ;  OR SEQUENCE ERROR
2461                              ;*****************************************************************************
2462                              ;TEST 72        TEST MODE 7 W/ NEGATE
2463                              ;*****************************************************************************
2464  006032  005212            TS72:   INC     (R2)            ;UPDATE TEST NUMBER
2465  006034  022712  000072            CMP     #72,(R2)        ;SEQUENCE ERROR?
2466  006040  001024                    BNE     TS73-10         ;BR TO ERROR HALT ON SEQ ERROR
2467  006042  005000                    CLR     R0              ;R0=0
2468  006044  005010                    CLR     (R0)            ;LOC. 0-0
2469  006046  005110                    COM     (R0)            ;LOC. 0=177777
2470  006050  105100                    COMB    R0              ;R0=377
2471  006052  105470  000005            NEGB    @5(R0)          ;R0+5=404,  404=1,  LOC. 0=777
2472  006056  100403                    BMI     NEG70           ;CC=0001?
2473  006060  001402                    BEQ     NEG70
2474  006062  102401                    BVS     NEG70
2475  006064  103404                    BCS     NEG71
2476                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2477                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2478                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2479                                                            ;           WHICH FOLLOWS W/ 765           <====
2480  006066                    NEG70:
2481  006066  012742  000144            MOV     #144,-(R2)      ;MOVE TO MAILBOX #  *******  144  *******
2482  006072  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2483  006074  000000                    HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2484  006076  105100            NEG71:  COMB    R0              ;R0=0
2485  006100  105120                    COMB    (R0)+           ;LOC. 0 400,  R0 =
2486  006102  105310                    DECB    (R0)            ;LOC. 0=0
```

```
2487  006104  005467  171670          NEG     0               ;USE NEG MODE 67 TO TST FOR ZERO
2488  006110  001404                  BEQ     TS73
2489                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < -
2490                                                          ;           CONDITIONAL BRANCH INST. AND   <
2491                                                          ;           REPLACE THE MOVE INSTRUCTION   < =
2492                                                          ;           WHICH FOLLOWS W/ 753           <
2493  006112  012742  000145          MOV     #145,-(R2)      ;MOVE TO MAILBOX #  *******  145  *******
2494  006116  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2495  006120  000000                  HALT                    ;DATA RESULT OF NEG WAS INCORRECT
2496                                                          ;  OR SEQUENCE ERROR
2497
2498                            ;*****************************************************************************
2499                            ;
2500                            ;       THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
2501                            ;INSTRUCTIONS.  CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
2502                            ;INSTRUCTION (SOPX).  THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
2503                            ;77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
2504                            ;OF THESE INSTRUCTIONS.
2505                            ;
2506                            ;*****************************************************************************
2507                            ;TEST 73        TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
2508                            ;*****************************************************************************
2509  006122  005212            TS73:   INC     (R2)            ;UPDATE TEST NUMBER
2510  006124  022712  000073            CMP     #73,(R2)        ;SEQUENCE ERROR?
2511  006130  001017                    BNE     SOPB            ;BR TO ERROR HALT ON SEQ ERROR
2512  006132  005027                    CLR     (R7)+           ;CLEAR NEXT LOCATION: (SOPX)
2513  006134  177777            SOPX:   -1                      ;USE MODE 27
2514  006136  001404                    BEQ     SOPA
2515                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2516                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2517                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2518                                                          ;           WHICH FOLLOWS W/ 774           <====
2519  006140  012742  000146          MOV     #146,-(R2)      ;MOVE TO MAILBOX #  *******  146  *******
2520  006144  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2521  006146  000000                  HALT                    ;CLR DID NOT SET Z-BIT
2522  006150  005237  006134    SOPA:   INC     @#SOPX          ;INC SOPX W/MODE 37
2523  006154  005467  177754          NEG     SOPX            ;NEGATE SOPX W/MODE 67
2524  006160  100003                  BPL     SOPB
2525  006162  005277  000012          INC     @SOPXAD         ;INC SOPX W/MODE 77
2526  006166  001405                  BEQ     TS74
2527                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2528                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2529                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2530                                                          ;           WHICH FOLLOWS W/ 760           <====
2531  006170                    SOPB:
2532  006170  012742  000147          MOV     #147,-(R2)      ;MOVE TO MAILBOX #  *******  147  *******
2533  006174  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2534  006176  000000                  HALT                    ;INC DID NOT SET Z-BIT
2535                                                          ;  OR SEQUENCE ERROR
2536  006200  006134            SOPXAD: SOPX                  ;INDIRECT ADDRESS OF SOPX
2537
2538                            ;*****************************************************************************
2539                            ;
2540                            ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
2541                            ;USING MODE 0.  R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
2542                            ;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 74         TEST MODE 0 SOP NON-MODIFYING
                                ;*****************************************************************************
006202  005212                TS74:   INC     (R2)            ;UPDATE TEST NUMBER
006204  022712  000074                CMP     #74,(R2)        ;SEQUENCE ERROR?
006210  001010                        BNE     TS75-10         ;BR TO ERROR HALT ON SEQ ERROR
006212  005000                        CLR     R0              ;INITIALIZE R0=0
006214  000277                        SCC                     ;SET CC=1011
006216  000244                        CLZ
006220  005700                        TST     R0              ;TRY TST W/ MODE 0
006222  102403                        BVS     SNM0A           ;CHECK THAT CC=0100
006224  100402                        BMI     SNM0A
006226  103401                        BCS     SNM0A
006230  001404                        BEQ     TS75
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 767           <====
006232                          SNM0A:
006232  012742  000150                MOV     #150,-(R2)      ;MOVE TO MAILBOX # ******* 150 *******
006236  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006240  000000                        HALT                    ;CONDITION CODES NOT SET PROPERLY
                                                            ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       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 75         TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING
                                ;*****************************************************************************
006242  005212                TS75.   INC     (R2)            ;UPDATE TEST NUMBER
006244  022712  000075                CMP     #75,(R2)        ;SEQUENCE ERROR?
006250  001010                        BNE     TS76-10         ;BR TO ERROR HALT ON SEQ ERROR
006252  005000                        CLR     R0              ;INITIALIZE
006254  105100                        COMB    R0              ;R0=377
006256  000277                        SCC                     ;SET CC=0111
006260  000250                        CLN
006262  105700                        TSTB    R0              ;TRY TST EVEN BYTE
006264  102402                        BVS     SNMB0A          ;CHECK CC=1000
006266  101401                        BLOS    SNMB0A
006270  100404                        BMI     TS76
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= ==
                                                            ;           CONDITIONAL BRANCH INST. AND   <-===
                                                            ;           REPLACE THE MOVE INSTRUCTION   <=-==
                                                            ;           WHICH FOLLOWS W/ 767           <=--=
006272                          SNMB0A:
006272  012742  000151                MOV     #151,-(R2)      ;MOVE TO MAILBOX # ******* 151 *******
006276  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
2599  006300  000000                        HALT                    ;CONDITION CODES NOT SET PROPERLY
2600                                                                  ;  OR SEQUENCE ERROR
2601
2602                                  ;*************************************************************************
2603                                  ;
2604                                  ;       THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
2605                                  ;R0 IS USED TO POINT TO AND CLEAR LOC. 0.  THE COMPLEMENT OF THE
2606                                  ;EXPECTED CONDITION CODES ARE LOADED IN THE PSW.  A TST INSTRUCTION
2607                                  ;IS THEN EXECUTED ON LOC. 0 USING R0 AND CONDITIONAL BRANCHES TEST
2608                                  ;THE RESULTS.
2609                                  ;
2610                                  ;*************************************************************************
2611                                  ;TEST 76        TEST MODE 1 SOP NON-MODIFYING
2612                                  ;*************************************************************************
2613  006302  005212                  TS76:   INC     (R2)            ;UPDATE TEST NUMBER
2614  006304  022712  000076                  CMP     #76,(R2)        ;SEQUENCE ERROR?
2615  006310  001011                          BNE     TS77-10         ;BR TO ERROR HALT ON SEQ ERROR
2616  006312  005000                          CLR     R0              ;POINT TO LOC 0
2617  006314  005010                          CLR     (R0)            ;CLEAR LOC 0
2618  006316  000277                          SCC                     ;INITIALIZE
2619  006320  000244                          CLZ                     ;CC=1011
2620  006322  005710                          TST     (R0)            ;TRY TST W/ MODE 1
2621  006324  102403                          BVS     SNM1A           ;CHECK CC=0100
2622  006326  103402                          BCS     SNM1A
2623  006330  100401                          BMI     SNM1A
2624  006332  001404                          BEQ     TS77
2625                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2626                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2627                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2628                                                                  ;           WHICH FOLLOWS W/ 766           <====
2629  006334                          SNM1A:
2630  006334  012742  000152                  MOV     #152,-(R2)      ;MOVE TO MAILBOX #  *******  152  *******
2631  006340  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2632  006342  000000                          HALT                    ;CC'S NOT SET PROPERLY
2633                                                                  ;  OR SEQUENCE ERROR
2634
2635                                  ;*************************************************************************
2636                                  ;
2637                                  ;       THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
2638                                  ;THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
2639                                  ;AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
2640                                  ;PROPER CONDITION CODE BITS.
2641                                  ;
2642                                  ;*************************************************************************
2643                                  ;TEST 77        TEST MODE 1 BYTE INST. NON-MODIFYING
2644                                  ;*************************************************************************
2645  006344  005212                  TS77:   INC     (R2)            ;UPDATE TEST NUMBER
2646  006346  022712  000077                  CMP     #77,(R2)        ;SEQUENCE ERROR?
2647  006352  001026                          BNE     TS100-10        ;BR TO ERROR HALT ON SEQ ERROR
2648  006354  005000                          CLR     R0              ;POINT TO LOC 0
2649  006356  005010                          CLR     (R0)            ;CLEAR LOC 0
2650  006360  105110                          COMB    (R0)            ;COMPLEMENT BYTE 0
2651  006362  000277                          SCC                     ;SET CC=0111
2652  006364  000250                          CLN
2653  006366  105710                          TSTB    (R0)            ;TRY TST ON EVEN BYTE
2654  006370  102402                          BVS     SNMB1A
```

```
2655  006372  101401                          BLOS    SNMB1A
2656  006374  100404                          BMI     SNMB1B
2657                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
2658                                                          ;           CONDITIONAL BRANCH INST. AND   <==-=
2659                                                          ;           REPLACE THE MOVE INSTRUCTION   <-===
2660                                                          ;           WHICH FOLLOWS W/ 766           <====
2661  006376                            SNMB1A:
2662  006376  012742  000153                        MOV     #153,-(R2)      ;MOVE TO MAILBOX #  *******  153  *******
2663  006402  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2664  006404  000000                          HALT                    ;CC'S NOT CORRECT
2665  006406  005000                    SNMB1B: CLR     R0
2666  006410  005200                          INC     R0
2667  006412  000277                          SCC                     ;SET CC=1011
2668  006414  000244                          CLZ
2669  006416  105710                          TSTB    (R0)            ;TRY TO TST AN ODD BYTE
2670  006420  102403                          BVS     SNMB1C          ;CHECK CC=0100
2671  006422  103402                          BCS     SNMB1C
2672  006424  100401                          BMI     SNMB1C
2673  006426  001404                          BEQ     TS100
2674                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2675                                                          ;           CONDITIONAL BRANCH INST. AND   <-===
2676                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2677                                                          ;           WHICH FOLLOWS W/ 751           <-= =
2678  006430                            SNMB1C:
2679  006430  012742  000154                        MOV     #154,-(R2)      ;MOVE TO MAILBOX #  *******  154  *******
2680  006434  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2681  006436  000000                          HALT                    ;CC'S NOT CORRECT
2682                                                                  ;  OR SEQUENCE ERROR
2683
2684                                    ;*****************************************************************************
2685                                    ;
2686                                    ;       THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
2687                                    ;USING MODE 2.  IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
2688                                    ;MODE 1 TESTS.  ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
2689                                    ;IT IS INCREMENTED PROPERLY.
2690                                    ;
2691                                    ;*****************************************************************************
2692                                    ;TEST 100       TEST MODE 2 WITH SOP NON-MODIFYING
2693                                    ;*****************************************************************************
2694  006440  005212                    TS100:  INC     (R2)            ;UPDATE TEST NUMBER
2695  006442  022712  000100                    CMP     #100,(R2)       ;SEQUENCE ERROR?
2696  006446  001020                          BNE     TS101-10        ;BR TO ERROR HALT ON SEQ ERROR
2697  006450  005000                          CLR     R0              ;INITIALIZE R0=0
2698  006452  005010                          CLR     (R0)            ;CLEAR LOC 0
2699  006454  000277                          SCC                     ;SET CC=1011
2700  006456  000244                          CLZ
2701  006460  005720                          TST     (R0)+           ;TRY TST W/ MODE 2
2702  006462  102403                          BVS     SNM2A           ;CHECK CC=0100
2703  006464  103402                          BCS     SNM2A
2704  006466  100401                          BMI     SNM2A
2705  006470  001404                          BEQ     SNM2B
2706                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= ==
2707                                                          ;           CONDITIONAL BRANCH INST. AND   <=-=-
2708                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2709                                                          ;           WHICH FOLLOWS W/ 766           <-- -
2710  006472                            SNM2A:
```

```
2711  006472  012742  000155                  MOV     #155,-(R2)      ;MOVE TO MAILBOX #  *******  155  *******
2712  006476  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2713  006500  000000                          HALT                    ;CC'S NOT CORRECT
2714  006502  005300                  SNM2B:  DEC     R0              ;RESET R0
2715  006504  005300                          DEC     R0
2716  006506  001404                          BEQ     TS101
2717                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2718                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2719                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2720                                                                  ;           WHICH FOLLOWS W/ 757           <====
2721  006510  012742  000156                  MOV     #156,-(R2)      ;MOVE TO MAILBOX #  *******  156  *******
2722  006514  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2723  006516  000000                          HALT                    ;MODE 2 DID NOT INC REQ CORRECTLY
2724                                                                  ;  OR SEQUENCE ERROR
2725
2726                                  ;*****************************************************************************************
2727                                  ;
2728                                  ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
2729                                  ;INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0.  WITH LOCATION 0
2730                                  ;SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
2731                                  ;TO VERIFY THE CORRECT CC ARE SET.  THE REGISTER IS CHECKED FOR
2732                                  ;PROPER INCREMENTING.
2733                                  ;
2734                                  ;*****************************************************************************************
2735                                  ;TEST 101       TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
2736                                  ;*****************************************************************************************
2737  006520  005212                  TS101:  INC     (R2)            ;UPDATE TEST NUMBER
2738  006522  022712  000101                  CMP     #101,(R2)       ;SEQUENCE ERROR?
2739  006526  001042                          BNE     TS102-10        ;BR TO ERROR HALT ON SEQ ERROR
2740  006530  005000                          CLR     R0              ;CLEAR R0
2741  006532  005010                          CLR     (R0)            ;CLEAR LOC 0
2742  006534  105110                          COMB    (R0)            ;SET LOC 0=377
2743  006536  000277                          SCC                     ;SET CC=0111
2744  006540  000250                          CLN
2745  006542  105720                          TSTB    (R0)+           ;TRY TST OF EVEN BYTE
2746  006544  102402                          BVS     SNMB2A
2747  006546  101401                          BLOS    SNMB2A
2748  006550  100404                          BMI     SNMB2B
2749                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2750                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2751                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2752                                                                  ;           WHICH FOLLOWS W/ 766           <====
2753  006552                  SNMB2A:
2754  006552  012742  000157                  MOV     #157,-(R2)      ;MOVE TO MAILBOX #  *******  157  *******
2755  006556  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2756  006560  000000                          HALT                    ;CC'S NOT SET CORRECTLY
2757  006562  005300                  SNMB2B: DEC     R0              ;DECREMENT R0
2758  006564  001404                          BEQ     SNMB2C
2759                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2760                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2761                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2762                                                                  ;           WHICH FOLLOWS W/ 760           <====
2763  006566  012742  000160                  MOV     #160,-(R2)      ;MOVE TO MAILBOX #  *******  160  *******
2764  006572  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2765  006574  000000                          HALT                    ;MODE 2 DID NOT INC REG CORRECTLY
2766  006576  005200                  SNMB2C: INC     R0              ;POINT TO ODD BYTE
```

```
2767  006600  000277                          SCC                     ;SET CC=1011
2768  006602  000244                          CLZ
2769  006604  105720                          TSTB    (R0)+           ;TRY TST OF ODD BYTE
2770  006606  102403                          BVS     SNMB2D          ;CHECK CC'S=0100
2771  006610  103402                          BCS     SNMB2D
2772  006612  100401                          BMI     SNMB2D
2773  006614  001404                          BEQ     SNMB2E
2774                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2775                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2776                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2777                                                                  ;           WHICH FOLLOWS W/ 744           <====
2778  006616                          SNMB2D:
2779  006616  012742  000161                    MOV     #161,-(R2)      ;MOVE TO MAILBOX #  *******  161  *******
2780  006622  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2781  006624  000000                          HALT                    ;CC'S NOT CORRECT
2782  006626  005300                  SNMB2E: DEC     R0
2783  006630  005300                          DEC     R0
2784  006632  001404                          BEQ     TS102
2785                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2786                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2787                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2788                                                                  ;           WHICH FOLLOWS W/ 735           <====
2789  006634  012742  000162                    MOV     #162,-(R2)      ;MOVE TO MAILBOX #  *******  162  *******
2790  006640  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2791  006642  000000                          HALT                    ;R0 DID NOT INCREMENT PROPERLY
2792                                                                  ;  OR SEQUENCE ERROR
2793
2794                                  ;*****************************************************************************************
2795                                  ;
2796                                  ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
2797                                  ;A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
2798                                  ;THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
2799                                  ;TST MODE 3 INSTRUCTION.
2800                                  ;
2801                                  ;*****************************************************************************************
2802                                  ;TEST 102       TEST MODE 3 W/ SOP NON-MODIFYING INSTS
2803                                  ;*****************************************************************************************
2804  006644  005212                  TS102:  INC     (R2)            ;UPDATE TEST NUMBER
2805  006646  022712  000102                  CMP     #102,(R2)       ;SEQUENCE ERROR?
2806  006652  001022                          BNE     TS103-10        ;BR TO ERROR HALT ON SEQ ERROR
2807  006654  005000                          CLR     R0              ;R0=0
2808  006656  005010                          CLR     (R0)            ;CLEAR LOC 0
2809  006660  105100                          COMB    R0              ;R0=376
2810  006662  005300                          DEC     R0
2811  006664  000277                          SCC                     ;SET CC=1011
2812  006666  000244                          CLZ
2813  006670  005730                          TST     @(R0)+          ;TRY TST W/ MODE 3
2814  006672  102403                          BVS     SNM3A           ;CHECK CC=0100
2815  006674  103402                          BCS     SNM3A
2816  006676  100401                          BMI     SNM3A
2817  006700  001404                          BEQ     SNM3B
2818                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
2819                                                                  ;           CONDITIONAL BRANCH INST. AND   <=-==
2820                                                                  ;           REPLACE THE MOVE INSTRUCTION   <=-==
2821                                                                  ;           WHICH FOLLOWS W/ 764           <=---
2822  006702                          SNM3A:
```

```
2823  006702  012742  000163                  MOV     #163,-(R2)      ;MOVE TO MAILBOX #  *******  163  *******
2824  006706  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2825  006710  000000                          HALT                    ;CC'S NOT CORRECT
2826  006712  005300                  SNM3B:  DEC     R0              ;R0=377
2827  006714  105100                          COMB    R0              ;R0=0
2828  006716  001404                          BEQ     TS103
2829                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==--
2830                                                                  ;           CONDITIONAL BRANCH INST. AND   < =--
2831                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-- -
2832                                                                  ;           WHICH FOLLOWS W/ 755           <---=
2833  006720  012742  000164                  MOV     #164,-(R2)      ;MOVE TO MAILBOX #  *******  164  *******
2834  006724  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2835  006726  000000                          HALT                    ;MODE 3 DID NOT INC REG CORRECTLY
2836                                                                  ;  OR SEQUENCE ERROR
2837
2838                                  ;*****************************************************************************************
2839                                  ;
2840                                  ;       THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3
2841                                  ;LOC. 0 IS SET TO 377.  TABLE AT LOC. 402-404 IS USED TO TEST
2842                                  ;BYTE 0 AND BYTE 1.  THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND
2843                                  ;THE CC'S ARE VERIFIED.
2844                                  ;       THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND
2845                                  ;AFTER THE TEST IS RUN.
2846                                  ;
2847                                  ;*****************************************************************************************
2848                                  ;TEST 103       TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
2849                                  ;*****************************************************************************************
2850  006730  005212                  TS103:  INC     (R2)            ;UPDATE TEST NUMBER
2851  006732  022712  000103                  CMP     #103,(R2)       ;SEQUENCE ERROR?
2852  006736  001036                          BNE     TS104-10        ;BR TO ERROR HALT ON SEQ ERROR
2853  006740  005000                          CLR     R0              ;R0=0
2854  006742  005010                          CLR     (R0)            ;CLEAR LOC 0
2855  006744  105110                          COMB    (R0)            ;LOC. 0 =377
2856  006746  105100                          COMB    R0
2857  006750  005200                          INC     R0
2858  006752  005720                          TST     (R0)+           ;R0=402
2859  006754  000277                          SCC                     ;CC=0111
2860  006756  000250                          CLN
2861  006760  105730                          TSTB    @(R0)+          ;TRY TST OF EVEN BYTE
2862  006762  102402                          BVS     SNMB3A          ;CHECK CC=1000
2863  006764  101401                          BLOS    SNMB3A
2864  006766  100404                          BMI     SNMB3B
2865                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
2866                                                                  ;           CONDITIONAL BRANCH INST. AND   <---_
2867                                                                  ;           REPLACE THE MOVE INSTRUCTION   <=-=-
2868                                                                  ;           WHICH FOLLOWS W/ 763           <= =-
2869  006770                          SNMB3A:
2870  006770  012742  000165                  MOV     #165,-(R2)      ;MOVE TO MAILBOX #  *******  165  *******
2871  006774  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2872  006776  000000                          HALT                    ;CC'S NOT CORRECT
2873  007000  000277                  SNMB3B: SCC                     ;SET CC=1011
2874  007002  000244                          CLZ
2875  007004  105730                          TSTB    @(R0)+          ;TRY TST OF ODD BYTE
2876  007006  102403                          BVS     SNMB3C          ;CHECK CC=0100
2877  007010  103402                          BCS     SNMB3C
2878  007012  100401                          BMI     SNMB3C
```

```
2879  007014  001404                        BEQ     SNM83D
2880                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2881                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2882                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2883                                                          ;           WHICH FOLLOWS W/ 750           <====
2884  007016                            SNM83C:
2885  007016  012742  000166                   MOV     #166,-(R2)      ;MOVE TO MAILBOX #  *******  166  *******
2886  007022  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2887  007024  000000                           HALT                    ;CC'S NOT CORRECT
2888  007026  005720                    SNM83D: TST     (R0)+           ;R0=410
2889  007030  005710                           TST     (R0)
2890  007032  100404                           BMI     TS104
2891                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2892                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2893                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2894                                                          ;           WHICH FOLLOWS W/ 741           <====
2895  007034  012742  000167                   MOV     #167,-(R2)      ;MOVE TO MAILBOX #  *******  167  *******
2896  007040  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2897  007042  000000                           HALT                    ;TSTB DID NOT INCREMENT R0 CORRETCLY
2898                                                                    ;  OR SEQUENCE ERROR
2899                                    ;*****************************************************************************
2900                                    ;
2901                                    ;       THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
2902                                    ;LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
2903                                    ;EXPECTED RESULTS.  R0 AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
2904                                    ;THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
2905                                    ;IS CHECKED FOR PROPER DECREMENTING.
2906                                    ;
2907                                    ;*****************************************************************************
2908                                    ;TEST 104       TEST MODE 4 W/ SOP NON-MODIFYING INSTS
2909                                    ;*****************************************************************************
2910  007044  005212                    TS104:  INC     (R2)            ;UPDATE TEST NUMBER
2911  007046  022712  000104                   CMP     #104,(R2)       ;SEQUENCE ERROR?
2912  007052  001017                           BNE     TS105-10        ;BR TO ERROR HALT ON SEQ ERROR
2913  007054  005000                           CLR     R0              ;R0=0
2914  007056  005010                           CLR     (R0)            ;LOC 0=0
2915  007060  005120                           COM     (R0)+           ;LOC 0=-1
2916  007062  000277                           SCC                     ;SET CC=1011
2917  007064  000244                           CLZ
2918  007066  005740                           TST     -(R0)           ;TRY TST W/ MODE 4
2919  007070  102402                           BVS     SNM4A           ;CHECK CC=0100
2920  007072  101401                           BLOS    SNM4A
2921  007074  100404                           BMI     SNM4B
2922                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2923                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2924                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2925                                                          ;           WHICH FOLLOWS W/ 766           <====
2926  007076                            SNM4A:
2927  007076  012742  000170                   MOV     #170,-(R2)      ;MOVE TO MAILBOX #  *******  170  *******
2928  007102  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2929  007104  000000                           HALT                    ;CC'S NOT CORRECT
2930  007106  005700                    SNM4B:  TST     R0
2931  007110  001404                           BEQ     TS105
2932                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2933                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2934                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
2935
2936  007112  012742  000171                MOV     #171,-(R2)      ;MOVE TO MAILBOX #  *******  171  *******
2937  007116  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2938  007120  000000                        HALT                    ;TST MODE 4 DID NOT DEC R0 CORRECTLY
2939                                                                  ;  OR SEQUENCE ERROR
2940
2941                                ;*****************************************************************************
2942                                ;
2943                                ;       THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
2944                                ;IT USES A POINTER AT LOC. 376 TO TEST LOC. 0.  R0 IS SET
2945                                ;TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
2946                                ;R0 IS CHECKED TO INSURE PROPER DECREMENTING.
2947                                ;
2948                                ;*****************************************************************************
2949                                ;TEST 105       TEST MODE 5 W/ SOP NON-MODIFYING INSTS
2950                                ;*****************************************************************************
2951  007122  005212                TS105:  INC     (R2)            ;UPDATE TEST NUMBER
2952  007124  022712  000105                CMP     #105,(R2)       ;SEQUENCE ERROR?
2953  007130  001022                        BNE     TS106-10        ;BR TO ERROR HALT ON SEQ ERROR
2954  007132  005000                        CLR     R0              ;R0=0
2955  007134  005010                        CLR     (R0)            ;LOC 0-0
2956  007136  005110                        COM     (R0)            ;LOC 0=-1
2957  007140  105100                        COMB    R0              ;R0=377
2958  007142  005200                        INC     R0              ;R0=400
2959  007144  000277                        SCC                     ;SET CC=0111
2960  007146  000250                        CLN
2961  007150  005750                        TST     @-(R0)          ;TRY TST W/ MODE 5
2962  007152  102402                        BVS     SNM5A           ;CHECK CC=1000
2963  007154  101401                        BLOS    SNM5A
2964  007156  100404                        BMI     SNM5B
2965                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2966                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2967                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2968                                                                  ;           WHICH FOLLOWS W/ 764           <====
2969  007160                        SNM5A:
2970  007160  012742  000172                MOV     #172,-(R2)      ;MOVE TO MAILBOX #  *******  172  *******
2971  007164  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2972  007166  000000                        HALT                    ;CC'S NOT SET PROPERLY
2973  007170  005200                SNM5B:  INC     R0              ;R0=377
2974  007172  105100                        COMB    R0              ;R0=0
2975  007174  001404                        BEQ     TS106
2976                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2977                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2978                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2979                                                                  ;           WHICH FOLLOWS W/ 755           <====
2980  007176  012742  000173                MOV     #173,-(R2)      ;MOVE TO MAILBOX #  *******  173  *******
2981  007202  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2982  007204  000000                        HALT                    ;MODE 5 DID NOT DEC R0 CORRECTLY
2983                                                                  ;  OR SEQUENCE ERROR
2984
2985                                ;*****************************************************************************
2986                                ;
2987                                ;       THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.
2988                                ;R0 IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED
2989                                ;USING R0 AND AN OFFSET OF -377.  THE CC'S ARE CHECKED AS WELL
2990                                ;AS R0 TO INSURE IT WAS NOT ALTERED.
```

```
                                ;
                                ;*********************************************************************************
                                ;TEST 106       TEST MODE 6 W/ SOP NON-MODIFYING INSTS
                                ;*********************************************************************************
007206  005212                  TS106:  INC     (R2)            ;UPDATE TEST NUMBER
007210  022712  000106                  CMP     #106,(R2)       ;SEQUENCE ERROR?
007214  001021                          BNE     TS107-10        ;BR TO ERROR HALT ON SEQ ERROR
007216  005000                          CLR     R0              ;R0=0
007220  005010                          CLR     (R0)            ;LOC 0=0
007222  005110                          COM     (R0)            ;LOC 0=-1
007224  105100                          COMB    R0              ;R0=377
007226  000277                          SCC                     ;SET CC=0111
007230  000250                          CLN
007232  005760  177401                  TST     -377(R0)        ;TRY TST W/ MODE 6
007236  102402                          BVS     SNM6A           ;CHECK CC=1000
007240  101401                          BLOS    SNM6A
007242  100404                          BMI     SNM6B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
007244                          SNM6A:
007244  012742  000174                  MOV     #174,-(R2)      ;MOVE TO MAILBOX #  *******  174  *******
007250  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007252  000000                          HALT                    ;CC'S INCORRECT
007254  105100                  SNM6B:  COMB    R0              ;R0=0
007256  001404                          BEQ     TS107
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 756           <====
007260  012742  000175                  MOV     #175,-(R2)      ;MOVE TO MAILBOX #  *******  175  *******
007264  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007266  000000                          HALT                    ;TST MODE 6 INCORRECTLY CHANGED R0
                                                                ;  OR SEQUENCE ERROR
```

```
                        ;*******************************************************************************
                        ;
                        ;       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 107       TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
                        ;*******************************************************************************
007270  005212          TS107:  INC     (R2)            ;UPDATE TEST NUMBER
007272  022712  000107          CMP     #107,(R2)       ;SEQUENCE ERROR?
007276  001021                  BNE     TS110-10        ;BR TO ERROR HALT ON SEQ ERROR
007300  005000                  CLR     R0              ;R0=0
007302  005010                  CLR     (R0)            ;LOC 0=0
007304  005110                  COM     (R0)            ;LOC 0=-1
007306  105100                  COMB    R0              ;R0=377
007310  000277                  SCC                     ;CC=0111
007312  000250                  CLN
007314  005770  000001          TST     @1(R0)          ;TRY TST W/ MODE 7
007320  102402                  BVS     SNM7A           ;CHECK CC=1000
007322  101401                  BLOS    SNM7A
007324  100404                  BMI     SNM7B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
007326                  SNM7A:
007326  012742  000176          MOV     #176,-(R2)      ;MOVE TO MAILBOX #  *******  176  *******
007332  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007334  000000                  HALT                    ;CC'S NOT CORRECT
007336  105100          SNM7B:  COMB    R0              ;R0=0
007340  001404                  BEQ     TS110
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
007342  012742  000177          MOV     #177,-(R2)      ;MOVE TO MAILBOX #  *******  177  *******
007346  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007350  000000                  HALT                    ;TST MODE 7 INCORRECTLY CHANGED R0
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 110        TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.
                                ;*****************************************************************************
007352  005212                  TS110:  INC     (R2)            ;UPDATE TEST NUMBER
007354  022712  000110                  CMP     #110,(R2)       ;SEQUENCE ERROR?
007360  001006                          BNE     TS111-10        ;BR TO ERROR HALT ON SEQ ERROR
007362  005000                          CLR     R0              ;R0=0
007364  005100                          COM     R0              ;R0=-1
007366  005004                          CLR     R4              ;R4=0
007370  060004                          ADD     R0,R4           ;TRY ADD: R4=-1
007372  005204                          INC     R4              ;R4=0
007374  001404                          BEQ     TS111
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
007376  012742  000200                      MOV     #200,-(R2)      ;MOVE TO MAILBOX #  *******  200  *******
007402  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007404  000000                          HALT                    ;ADD INST. FAILED W/ MODE 0
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.
                                ;
                                ;*****************************************************************************
                                ;TEST 111        MOV MODE 0 TO MODE 0
                                ;*****************************************************************************
007406  005212                  TS111:  INC     (R2)            ;UPDATE TEST NUMBER
007410  022712  000111                  CMP     #111,(R2)       ;SEQUENCE ERROR?
007414  001006                          BNE     TS112-10        ;BR TO ERROR HALT ON SEQ ERROR
007416  005000                          CLR     R0              ;R0=0
007420  005004                          CLR     R4              ;R4=0
007422  005100                          COM     R0              ;R0=-1
007424  010004                          MOV     R0,R4           ;TRY MOVE -1 TO R4
007426  005204                          INC     R4              ;INC R4
007430  001404                          BEQ     TS112
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
                                                                ;           CONDITIONAL BRANCH INST. AND   <
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
                                                                ;           WHICH FOLLOWS W/ 771           <
007432  012742  000201                      MOV     #201,-(R2)      ;MOVE TO MAILBOX #  *******  201  *******
007436  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007440  000000                          HALT                    ;MOVE FAILED MODE 0 TO MODE 0
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
```

```
3124
3125                                    ;
3126                                    ;*********************************************************************************
3127                                    ;TEST 112       TEST SUB MODE 0,0
3128    007442  005212                  ;*********************************************************************************
3129    007444  022712  000112  TS112:  INC     (R2)            ;UPDATE TEST NUMBER
3130    007450  001016                  CMP     #112,(R2)       ;SEQUENCE ERROR?
3131    007452  005000                  BNE     TS113-10        ;BR TO ERROR HALT ON SEQ ERROR
3132    007454  005004                  CLR     R0              ;R0=0
3133    007456  005204                  CLR     R4              ;R4=0
3134    007460  160400                  INC     R4              ;R4=1
3135    007462  100003                  SUB     R4,R0           ;TRY SUB 0,0   R0--1
3136    007464  001402                  BPL     SUB0            ;CC=1001
3137    007466  102401                  BEQ     SUB0
3138    007470  103404                  BVS     SUB0
3139                                    BCS     SUB0A
3140                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3141                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3142                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3143    007472                          SUB0:           ;           WHICH FOLLOWS W/ 767           <====
3144    007472  012742  000202                  MOV     #202,-(R2)      ;MOVE TO MAILBOX #  *******  202  *******
3145    007476  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3146    007500  000000                  HALT                    ;CONDITION CODE FAILED ON SUB
3147    007502  005200          SUB0A:  INC     R0
3148    007504  001404                  BEQ     TS113
3149                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3150                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3151                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3152                                                    ;           WHICH FOLLOWS W/ 761           <====
3153    007506  012742  000203                  MOV     #203,-(R2)      ;MOVE TO MAILBOX #  *******  203  *******
3154    007512  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3155    007514  000000                  HALT                    ;DATA RESULT OF SUB FAILED
3156                                                            ;  OR SEQUENCE ERROR
```

```
3157
3158
3159
3160                                    ;*****************************************************************************
3161                                    ;
3162                                    ;       THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
3163                                    ;WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
3164                                    ;SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
3165                                    ;BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
3166                                    ;VERIFIED.
3167                                    ;
3168                                    ;*****************************************************************************
3169                                    ;*****************************************************************************
3170                                    ;TEST 113       TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
3171                                    ;*****************************************************************************
3172 007516  005212                    TS113:  INC     (R2)            ;UPDATE TEST NUMBER
3173 007520  022712  000113                    CMP     #113,(R2)       ;SEQUENCE ERROR?
3174 007524  001051                            BNE     TS114-10        ;BR TO ERROR HALT ON SEQ ERROR
3175 007526  005000                            CLR     R0              ;R0=0
3176 007530  010004                            MOV     R0,R4           ;TRY MOVE MODE 0,0
3177 007532  001404                            BEQ     DOP0A
3178                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3179                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3180                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3181                                                                    ;           WHICH FOLLOWS W/ 774           <====
3182 007534  012742  000204                    MOV     #204,-(R2)      ;MOVE TO MAILBOX #  *******  204  *******
3183 007540  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3184 007542  000000                            HALT                    ;Z-BIT NOT SET
3185 007544  005200                    DOP0A:  INC     R0              ;R0=1
3186 007546  005100                            COM     R0              ;R0=177776
3187 007550  005104                            COM     R4              ;R4=177777
3188 007552  040004                            BIC     R0,R4           ;TRY BIC: R4=1
3189 007554  005304                            DEC     R4              ;R4=0
3190 007556  001404                            BEQ     DOP0B
3191                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3192                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3193                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3194                                                                    ;           WHICH FOLLOWS W/ 762           <====
3195 007560  012742  000205                    MOV     #205,-(R2)      ;MOVE TO MAILBOX #  *******  205  *******
3196 007564  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3197 007566  000000                            HALT                    ;BIC CLEAR RESULT INCORRECT
3198 007570  050004                    DOP0B:  BIS     R0,R4           ;TRY BIS: R4=177777
3199 007572  005204                            INC     R4
3200 007574  005204                            INC     R4              ;R4=0
3201 007576  001404                            BEQ     DOP0C
3202                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3203                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3204                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3205                                                                    ;           WHICH FOLLOWS W/ 752           <====
3206 007600  012742  000206                    MOV     #206,-(R2)      ;MOVE TO MAILBOX #  *******  206  *******
3207 007604  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3208 007606  000000                            HALT                    ;RESULT OF BIS INCORRECT
3209 007610  005000                    DOP0C:  CLR     R0              ;R0=0
3210 007612  105100                            COMB    R0              ;R0=377
3211 007614  005004                            CLR     R4              ;R4=0
3212 007616  005104                            COM     R4              ;R4=177777
3213 007620  040004                            BIC     R0,R4           ;R4=177400
3214 007622  060004                            ADD     R0,R4           ;TRY ADD: R4=177777
3215 007624  005204                            INC     R4              ;R4=0
```

```
3213  007626  001404                        BEQ     DOPOD
3214                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3215                                                      ;           CONDITIONAL BRANCH INST. AND   <
3216                                                      ;           REPLACE THE MOVE INSTRUCTION   <
3217                                                      ;           WHICH FOLLOWS W/ 736           .
3218  007630  012742  000207                  MOV     #207,-(R2)    ;MOVE TO MAILBOX #  *******  207  *******
3219  007634  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
3220  007636  000000                        HALT                  ;RESULT OF ADD INCORRECT
3221  007640  160004                DOPOD:  SUB     R0,R4         ;177401-R4
3222  007642  105404                        NEGB    R4            ;R4=177777
3223  007644  005204                        INC     R4            ;R0-0
3224  007646  001404                        BEQ     TS114
3225                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3226                                                      ;           CONDITIONAL BRANCH INST. AND   <
3227                                                      ;           REPLACE THE MOVE INSTRUCTION   <
3228                                                      ;           WHICH FOLLOWS W/ 726           <
3229  007650  012742  000210                  MOV     #210,-(R2)    ;MOVE TO MAILBOX #  *******  210  *******
3230  007654  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
3231  007656  000000                        HALT                  ;RESULT OF SUB INCORRECT
3232                                                      ;  OR SEQUENCE ERROR
3233
```

```
3234                                    ;*****************************************************************************
3235                                    ;
3236                                    ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
3237                                    ;DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
3238                                    ;
3239                                    ;*****************************************************************************
3240                                    ;TEST 114       TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
3241                                    ;*****************************************************************************
3242  007660  005212                   TS114:  INC     (R2)            ;UPDATE TEST NUMBER
3243  007662  022712  000114                   CMP     #114,(R2)       ;SEQUENCE ERROR?
3244  007666  001024                           BNE     TS115-10        ;BR TO ERROR HALT ON SEQ ERROR
3245  007670  005000                           CLR     R0              ;R0=0
3246  007672  005010                           CLR     (R0)            ;LOC. 0-0
3247  007674  105110                           COMB    (R0)            ;LOC. 0-377
3248  007676  005220                           INC     (R0)+           ;LOC. 0=400   R0-2
3249  007700  005400                           NEG     R0              ;R0=-2
3250  007702  060037  000000                   ADD     R0,@#0          ;TRY ADD 0,3;  LOC. 0-376
3251  007706  100405                           BMI     DOP03A          ;CC=0001?
3252  007710  001402                           BEQ     DOP03A
3253  007712  102401                           BVS     DOP03A
3254  007714  103404                           BCS     DOP03B
3255                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3256                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3257                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3258                                                                   ;           WHICH FOLLOWS W/ 764           <====
3259  007716                           DOP03A:
3260  007716  012742  000211                   MOV     #211,-(R2)      ;MOVE TO MAILBOX #  *******  211  *******
3261  007722  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3262  007724  000000                           HALT                    ;CC'S NOT SET CORRECTLY
3263  007726  105137  000000           DOP03B: COMB    @#0             ;LOC. 0=1
3264  007732  005337  000000                   DEC     @#0             ;LOC. 0-0
3265  007736  001404                           BEQ     TS115
3266                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3267                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3268                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3269                                                                   ;           WHICH FOLLOWS W/ 753           <====
3270  007740  012742  000212                   MOV     #212,-(R2)      ;MOVE TO MAILBOX #  *******  212  *******
3271  007744  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3272  007746  000000                           HALT                    ;DATA RESULT INCORRECT
3273                                                                   ;  OR SEQUENCE ERROR
```

```
3274                                    ;*****************************************************************************
3275                                    ;
3276                                    ;       THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
3277                                    ;R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY.  COMPARE INSTRUCTIONS ARE
3278                                    ;THEN EXECUTED AND CHECKED.  FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
3279                                    ;
3280                                    ;*****************************************************************************
3281                                    ;TEST 115       TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
3282                                    ;*****************************************************************************
3283  007750  005212                    TS115:  INC     (R2)            ;UPDATE TEST NUMBER
3284  007752  022712  000115                    CMP     #115,(R2)       ;SEQUENCE ERROR?
3285  007756  001042                            BNE     TS116-10        ;BR TO ERROR HALT ON SEQ ERROR
3286  007760  005000                            CLR     R0              ;R0=0
3287  007762  005004                            CLR     R4              ;R4=0
3288  007764  005204                            INC     R4              ;R4-1
3289  007766  020400                            CMP     R4,R0           ;TRY COMPARE R4 TO R0
3290  007770  003004                            BGT     DNM1
3291                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3292                                                                    ;           CONDITIONAL BRANCH INST. AND   <
3293                                                                    ;           REPLACE THE MOVE INSTRUCTION   <
3294                                                                    ;           WHICH FOLLOWS W/ 772           <-- -
3295  007772  012742  000213                    MOV     #213,-(R2)      ;MOVE TO MAILBOX #  *******  213  *******
3296  007776  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3297  010000  000000                            HALT                    ;CC'S NOT CORRECT FOR CMP
3298  010002  020004                    DNM1:   CMP     R0,R4           ;TRY COMPARE R0 TO R4
3299  010004  002404                            BLT     DNM2
3300                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
3301                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3302                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3303                                                                    ;           WHICH FOLLOWS W/ 764           <====
3304  010006  012742  000214                    MOV     #214,-(R2)      ;MOVE TO MAILBOX #  *******  214  *******
3305  010012  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3306  010014  000000                            HALT                    ;CC'S NOT CORRECT FOR CMP
3307  010016  005200                    DNM2:   INC     R0              ;R0=1
3308  010020  020400                            CMP     R4,R0           ;TRY COMPARE R4=1 TO R0=1
3309  010022  001404                            BEQ     DNM3
3310                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3311                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3312                                                                    ;           REPLACE THE MOVF INSTRUCTION   <====
3313                                                                    ;           WHICH FOLLOWS W/ 755           <====
3314  010024  012742  000215                    MOV     #215,-(R2)      ;MOVE TO MAILBOX #  *******  215  *******
3315  010030  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3316  010032  000000                            HALT                    ;CC'S NOT CORRECT (Z=1) FOR CMP
3317  010034  005000                    DNM3:   CLR     R0              ;R0=0
3318  010036  005100                            COM     R0              ;R0-177777
3319  010040  005004                            CLR     R4              ;R4=0
3320  010042  030004                            BIT     R0,R4           ;TRY BIT R0 TO R4
3321  010044  001404                            BEQ     DNM4
3322                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
3323                                                                    ;           CONDITIONAL BRANCH INST. AND   <= ==
3324                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3325                                                                    ;           WHICH FOLLOWS W/ 744           <- -
3326  010046  012742  000216                    MOV     #216,-(R2)      ;MOVE TO MAILBOX #  *******  216  *******
3327  010052  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3328  010054  000000                            HALT                    ;CC'S NOT CORRECT FOR BIT
3329  010056  005304                    DNM4:   DEC     R4              ;R4=177777
```

```
3330  010060  030004                        BIT     R0,R4           ;TRY BIT AGAIN
3331  010062  100404                        BMI     TS116
3332                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=---
3333                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3334                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3335                                                                  ;           WHICH FOLLOWS W/ 735           < ===
3336  010064  012742  000217                MOV     #217,-(R2)      ;MOVE TO MAILBOX #  *******  217  *******
3337  010070  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3338  010072  000000                        HALT                    ;CC'S NOT CORRECT FOR BIT
3339                                                                  ;  OR SEQUENCE ERROR
3340                                ;*****************************************************************************************
3341                                ;
3342                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
3343                                ;IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
3344                                ;
3345                                ;*****************************************************************************************
3346                                ;TEST 116       TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.
3347                                ;************************************* *************************************************
3348  010074  005212                TS116:  INC     (R2)            ;UPDATE TEST NUMBER
3349  010076  022712  000116                CMP     #116,(R2)       ;SEQUENCE ERROR?
3350  010102  001022                        BNE     TS117-10        ;BR TO ERROR HALT ON SEQ ERROR
3351  010104  005000                        CLR     R0              ;R0=0
3352  010106  005010                        CLR     (R0)            ;LOC. 0=0
3353  010110  005110                        COM     (R0)            ;LOC. 0=177777
3354  010112  005200                        INC     R0              ;R0=1
3355  010114  020037  000000                CMP     R0,@#0          ;TRY CMP MODE 0,3
3356  010120  100403                        BMI     DNM03A          ;CC=0001
3357  010122  001402                        BEQ     DNM03A
3358  010124  102401                        BVS     DNM03A
3359  010126  103404                        BCS     DNM03B
3360                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-  -
3361                                                                  ;           CONDITIONAL BRANCH INST. AND   <
3362                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-  -
3363                                                                  ;           WHICH FOLLOWS W/ 765           <= -_
3364  010130                        DNM03A:
3365  010130  012742  000220                MOV     #220,-(R2)      ;MOVE TO MAILBOX #  *******  220  *******
3366  010134  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3367  010136  000000                        HALT                    ;CC'S NOT SET CORRECTLY
3368  010140  005300                DNM03B: DEC     R0
3369  010142  001002                        BNE     DNM03C
3370  010144  005210                        INC     (R0)
3371  010146  001404                        BEQ     TS117
3372                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-==-
3373                                                                  ;           CONDITIONAL BRANCH INST. AND   <==
3374                                                                  ;           REPLACE THE MOVE INSTRUCTION   <=-=
3375                                                                  ;           WHICH FOLLOWS W/ 755           < --=
3376  010150                        DNM03C:
3377  010150  012742  000221                MOV     #221,-(R2)      ;MOVE TO MAILBOX #  *******  221  *******
3378  010154  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3379  010156  000000                        HALT                    ;DATA INCORRECTLY MODIFIED BY CMP
3380                                                                  ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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 117        TEST MODE 1 W/ DOP INST.
                                ;*****************************************************************************************
010160  005212                  TS117:  INC     (R2)            ;UPDATE TEST NUMBER
010162  022712  000117                  CMP     #117,(R2)       ;SEQUENCE ERROR?
010166  001007                          BNE     TS120-10        ;BR TO ERROR HALT ON SEQ ERROR
010170  005000                          CLR     R0              ;R0=0
010172  005100                          COM     R0              ;R0=177777
010174  005004                          CLR     R4              ;R4=0
010176  005014                          CLR     (R4)            ;LOC 0=0
010200  005214                          INC     (R4)            ;LOC 0=1
010202  061400                          ADD     (R4),R0         ;TRY ADD SOURCE MODE 1
010204  001404                          BEQ     TS120
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-  -
                                                                ;           CONDITIONAL BRANCH INST. AND   <  -=
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
                                                                ;           WHICH FOLLOWS W/ 770           <
010206  012742  000222                  MOV     #222,-(R2)      ;MOVE TO MAILBOX #  *******  222  *******
010212  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010214  000000                          HALT                    ;RESULT OF ADD INCORRECT
                                                                ;  OR SEQUENCE ERROR
```

```
3409
3410                            ;*****************************************************************************************
3411                            ;
3412                            ;       THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS
3413                            ;EVEN BYTES.  LOC. 0 IS SET TO -1 AND R4 IS CLEARED.  THEN R4 IS
3414                            ;SET TO -1 USING A BISB THRU R0 WITH MODE 1.
3415                            ;
3416                            ;*****************************************************************************************
3417                            ;TEST 120       TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
3418                            ;*****************************************************************************************
3419  010216  005212            TS120:  INC     (R2)            ;UPDATE TEST NUMBER
3420  010220  022712  000120            CMP     #120,(R2)       ;SEQUENCE ERROR?
3421  010224  001007                    BNE     TS121-10        ;BR TO ERROR HALT ON SEQ ERROR
3422  010226  005000                    CLR     R0              ;R0=0
3423  010230  005010                    CLR     (R0)            ;LOC. 0=0
3424  010232  005110                    COM     (R0)            ;LOC. 0=177777
3425  010234  005004                    CLR     R4              ;R4=0
3426  010236  151004                    BISB    (R0),R4         ;TRY MODE 1- EVEN BYTE W/ DOP
3427  010240  105104                    COMB    R4              ;R4=0
3428  010242  001404                    BEQ     TS121
3429                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3430                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3431                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3432                                                            ;           WHICH FOLLOWS W/ 770           <====
3433  010244  012742  000223                  MOV     #223,-(R2)      ;MOVE TO MAILBOX #  *******  223  *******
3434  010250  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3435  010252  000000                    HALT                    ;RESULT OF BISB IS INCORRECT
3436                                                            ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************************
                        ;
                        ;       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 121       TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
                        ;*****************************************************************************************
010254  005212          TS121:  INC     (R2)            ;UPDATE TEST NUMBER
010256  022712  000121          CMP     #121,(R2)       ;SEQUENCE ERROR?
010262  001007                  BNE     TS122-10        ;BR TO ERROR HALT ON SEQ ERROR
010264  005000                  CLR     R0              ;R0=0
010266  005010                  CLR     (R0)            ;LOC 0=0
010270  005110                  COM     (R0)            ;LOC 0=177777
010272  005004                  CLR     R4              ;R4=0
010274  105104                  COMB    R4              ;R4=377
010276  121004                  CMPB    (R0),R4         ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
010300  001404                  BEQ     TS122
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
010302  012742  000224          MOV     #224,-(R2)      ;MOVE TO MAILBOX #  *******  224  *******
010306  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010310  000000                  HALT                    ;RESULT OF CMPB INCORRECT
                                                        ;  OR SEQUENCE ERROR
```

```
3466
3467                            ;*****************************************************************************
3468                            ;
3469                            ;       THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
3470                            ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
3471                            ;R4 IS SET TO -1.  MOVB ARE USED TO MOVE BYTE 0 TO R4.  THIS
3472                            ;VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
3473                            ;FUNCTION WITH MODE 0.
3474                            ;       THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
3475                            ;THE LOGIC FOR COMPLEMENTARY DATA.
3476                            ;       THIS TEST EXERCISES UNIQUE MICROCODE.
3477                            ;
3478                            ;*****************************************************************************
3479                            ;TEST 122       TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
3480                            ;*****************************************************************************
3481  010312  005212            TS122:  INC     (R2)            ;UPDATE TEST NUMBER
3482  010314  022712  000122            CMP     #122,(R2)       ;SEQUENCE ERROR?
3483  010320  001020                    BNE     TS123-10        ;BR TO ERROR HALT ON SEQ ERROR
3484  010322  005000                    CLR     R0              ;R0=0
3485  010324  005010                    CLR     (R0)            ;LOC 0=0
3486  010326  105110                    COMB    (R0)            ;LOC 0=177400
3487  010330  005110                    COM     (R0)
3488  010332  005004                    CLR     R4              ;R4=0
3489  010334  005104                    COM     R4              ;R4=177777
3490  010336  111004                    MOVB    (R0),R4         ;R4=0
3491  010340  005704                    TST     R4              ;CHECK SIGN OF WORD
3492  010342  001404                    BEQ     DOP1
3493                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3494                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3495                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3496                                                            ;           WHICH FOLLOWS W/ 766           <====
3497  010344  012742  000225                  MOV     #225,-(R2)      ;MOVE TO MAILBOX #  *******  225  *******
3498  010350  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3499  010352  000000                    HALT                    ;MOVB SHOULD SIGN X-TEND
3500  010354  005110            DOP1:   COM     (R0)            ;LOC 0=177777
3501  010356  111004                    MOVB    (R0),R4         ;DO MOVB W/ EVEN BYTE
3502  010360  100404                    BMI     TS123
3503                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3504                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3505                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3506                                                            ;           WHICH FOLLOWS W/ 757           <====
3507  010362  012742  000226                  MOV     #226,-(R2)      ;MOVE TO MAILBOX #  *******  226  *******
3508  010366  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3509  010370  000000                    HALT                    ;MOVB SHOULD SIGN X-TEND
3510                                                            ;  OR SEQUENCE ERROR
```

```
3511
3512                            ;*****************************************************************************
3513                            ;
3514                            ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE
3515                            ;ODD BYTES.  LOC. 0 IS SET TO 177400.  R0 IS SET TO 0 AND R4 IS
3516                            ;SET TO 1.  THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.
3517                            ;THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.
3518                            ;
3519                            ;*****************************************************************************
3520                            ;TEST 123       TEST MODE 1-ODD BYTE W/ DOP INSTS.
3521                            ;*****************************************************************************
3522  010372  005212            TS123:  INC     (R2)            ;UPDATE TEST NUMBER
3523  010374  022712  000123           CMP     #123,(R2)       ;SEQUENCE ERROR?
3524  010400  001010                   BNE     TS124-10        ;BR TO ERROR HALT ON SEQ ERROR
3525  010402  005000                   CLR     R0              ;R0=0
3526  010404  005010                   CLR     (R0)            ;LOC. 0=0
3527  010406  005004                   CLR     R4              ;R4=0
3528  010410  005204                   INC     R4              ;R4=1
3529  010412  105114                   COMB    (R4)            ;LOC. 0=177400
3530  010414  151410                   BISB    (R4),(R0)       ;TRY TO BIS LOW ORDER BITS W/ MODE 1
3531  010416  005210                   INC     (R0)            ;CHECK RESULT
3532  010420  001404                   BEQ     TS124
3533                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3534                                                           ;           CONDITIONAL BRANCH INST. AND   <====
3535                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
3536                                                           ;           WHICH FOLLOWS W/ 767           <====
3537  010422  012742  000227           MOV     #227,-(R2)      ;MOVE TO MAILBOX #  *******  227  *******
3538  010426  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3539  010430  000000                   HALT                    ;RESULT OF BISB INCORRECT
3540                                                           ;  OR SEQUENCE ERROR
```

```
3541
3542                            ;*****************************************************************************
3543                            ;
3544                            ;       THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS.  LOC. 0 IS SET TO -1.
3545                            ;R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0
3546                            ;TO R7.  THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER
3547                            ;IS CHECKED.
3548                            ;
3549                            ;*****************************************************************************
3550                            ;TEST 124       TEST MODE 2 W/ DOP INSTS.
3551                            ;*****************************************************************************
3552  010432  005212            TS124:  INC     (R2)            ;UPDATE TEST NUMBER
3553  010434  022712  000124            CMP     #124,(R2)       ;SEQUENCE ERROR?
3554  010440  001015                    BNE     TS125-10        ;BR TO ERROR HALT ON SEQ ERROR
3555  010442  005000                    CLR     R0              ;R0=0
3556  010444  005010                    CLR     (R0)            ;LOC. 0=0
3557  010446  005110                    COM     (R0)            ;LOC. 0=177777
3558  010450  012004                    MOV     (R0)+,R4        ;TRY MOVE MODE 2,0
3559  010452  005204                    INC     R4              ;CHECK R4
3560  010454  001404                    BEQ     DOP2
3561                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3562                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3563                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3564                                                            ;           WHICH FOLLOWS W/ 771           <====
3565  010456  012742  000230          MOV     #230,-(R2)      ;MOVE TO MAILBOX #  *******  230  *******
3566  010462  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3567  010464  000000                    HALT                    ;RESULT OF MOV INST INCORRECT
3568  010466  005300            DOP2:   DEC     R0              ;TEST R0 AFTER MODE 2
3569  010470  005300                    DEC     R0
3570  010472  001404                    BEQ     TS125
3571                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3572                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3573                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3574                                                            ;           WHICH FOLLOWS W/ 762           <====
3575  010474  012742  000231          MOV     #231,-(R2)      ;MOVE TO MAILBOX #  *******  231  *******
3576  010500  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3577  010502  000000                    HALT                    ;REGISTER NOT INCREMENTED IN MODE 2
3578                                                            ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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 125       TEST MODE 2 - EVEN BYTE W/ DOP INST.
                                ;*****************************************************************************************
010504  005212        TS125:  INC     (R2)            ;UPDATE TEST NUMBER
010506  022712  000125        CMP     #125,(R2)       ;SEQUENCE ERROR?
010512  001016                BNE     TS126-10        ;BR TO ERROR HALT ON SEQ ERROR
010514  005000                CLR     R0              ;R0=0
010516  010010                MOV     R0,(R0)         ;LOC. 0=0
010520  005110                COM     (R0)            ;LOC. 0=177777
010522  142010                BICB    (R0)+,(R0)      ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
010524  105737  000001        TSTB    @#1             ;CHECK RESULT
010530  001404                BEQ     DOPB2A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
010532  012742  000232                MOV     #232,-(R2)      ;MOVE TO MAILBOX #  *******  232  *******
010536  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010540  000000                        HALT                    ;BICB DESTINATION INCORRECT
010542  105137  000000  DOPB2A: COMB    @#0             ;CHECK BICB SOURCE
010546  001404                BEQ     TS126
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
010550  012742  000233        MOV     #233,-(R2)      ;MOVE TO MAILBOX #  *******  233  *******
010554  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010556  000000                HALT                    ;BICB SOURCE INCORRECTLY CHANGED
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*******************************************************************************
                                ;
                                ;       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 PPOPER BYTE WAS MOVED AND SIGN X-TENDED.
                                ;
                                ;*******************************************************************************
                                ;TEST 126       TEST MODE 2 - ODD BYTE W/ DOP INST.
                                ;*******************************************************************************
010560  005212                TS126:  INC     (R2)            ;UPDATE TEST NUMBER
010562  022712  000126              CMP     #126,(R2)       ;SEQUENCE ERROR?
010566  001017                      BNE     TS127-10        ;BR TO ERROR HALT ON SEQ ERROR
010570  005000                      CLR     R0              ;R0=0
010572  005004                      CLR     R4              ;R4=0
010574  005010                      CLR     (R0)            ;LOC. 0=0
010576  005110                      COM     (R0)            ;LOC. 0=177777
010600  105120                      COMB    (R0)+           ;LOC 0=177400; R0=1
010602  112004                      MOVB    @R0)+,R4        ;TRY DOP MODE 2 W/ ODD BYTE
010604  005204                      INC   • R4              ;CHECK RESULT OF MOVB
010606  001404                      BEQ     DOPB2B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <-- -
                                                            ;           WHICH FOLLOWS W/ 767           <=- -
010610  012742  000234                MOV     #234,-(R2)      ;MOVE TO MAILBOX #  *******  234  *******
010614  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010616  000000                        HALT                    ;RESULT OF MOVB INCORRECT
010620  005740                DOPB2B: TST     -(R0)           ;BUMP R0 DOWN BY 2
010622  005700                        TST     R0              ;CHECK R0
010624  001404                        BEQ     TS127
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
                                                            ;           CONDITIONAL BRANCH INST. AND   <  --
                                                            ;           REPLACE THE MOVE INSTRUCTION   < =--
                                                            ;           WHICH FOLLOWS W/ 760           <-
010626  012742  000235                MOV     #235,-(R2)      ;MOVE TO MAILBOX #  *******  235  *******
010632  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010634  000000                        HALT                    ;MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY
                                                            ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 127       TEST MODE 3 W/ DOP INSTS.
                                ;*****************************************************************************
010636  005212                  TS127:  INC     (R2)            ;UPDATE TEST NUMBER
010640  022712  000127                  CMP     #127,(R2)       ;SEQUENCE ERROR?
010644  001011                          BNE     TS130-10        ;BR TO ERROR HALT ON SEQ ERROR
010646  012737  052525  000000          MOV     #052525,@#0     ;MOVE 52525 TO LOC. 0
010654  012700  125252                  MOV     #125252,R0      ;SET ALT. ONE AND ZERO IN R0
010660  053700  000000                  BIS     @#0,R0          ;TRY TO SET ALL OTHER BITS W/ MODE 3
010664  005200                          INC     R0              ;TEST RESULT
010666  001404                          BEQ     TS130
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
010670  012742  000236                  MOV     #236,-(R2)      ;MOVE TO MAILBOX #  *******  236  *******
010674  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010676  000000                          HALT                    ;BIS W/ MODE 3 INCORRECT RESULT
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       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 130       TEST MODE 3 - EVEN BYTE W/ DOP INSTS.
                                ;*****************************************************************************
010700  005212                  TS130:  INC     (R2)            ;UPDATE TEST NUMBER
010702  022712  000130                  CMP     #130,(R2)       ;SEQUENCE ERROR?
010706  001011                          BNE     TS131-10        ;BR TO ERROR HALT ON SEQ ERROR
010710  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0' PATTERN TO LOC. 0
010716  005000                          CLR     R0              ;R0=0
010720  153700  000000                  BISB    @#0,R0          ;TRY R0=252 W/ MODE 3 - EVEN BYTE
010724  022700  000252                  CMP     #252,R0         ;BISB W/ EVEN BYTE SUCCESSFUL?
010730  001404                          BEQ     TS131
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
010732  012742  000237                  MOV     #237,-(R2)      ;MOVE TO MAILBOX #  *******  237  *******
010736  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010740  000000                          HALT                    ;BISB W/ MODE 3 - EVEN BYTE FAILED
                                                                ;  OR SEQUENCE ERROR
```

```
3710                                    ;*****************************************************************************
3711                                    ;
3712                                    ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
3713                                    ;WHICH ADDRESS ODD BYTES.  THE SAME PROCEDURE USED IN PREVIOUS
3714                                    ;TEST IS USED HERE.  THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
3715                                    ;THE EXPECTED RESULT IS:  R0   125.
3716                                    ;
3717                                    ;*****************************************************************************
3718                                    ;TEST 131       TEST MODE 3 - ODD BYTE W/ DOP INSTS.
3719                                    ;*****************************************************************************
3720  010742  005212                    TS131:  INC     (R2)            ;UPDATE TEST NUMBER
3721  010744  022712  000131                   CMP     #131,(R2)       ;SEQUENCE ERROR?
3722  010750  001011                           BNE     TS132-10        ;BR TO ERROR HALT ON SEQ ERROR
3723  010752  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0'S PATTERN TO LOC 0
3724  010760  005000                           CLR     R0              ;R0=0
3725  010762  153700  000001                   BISB    @#1,R0          ;TRY R0=152 W/ MODE 3 - ODD BYTE
3726  010766  022700  000125                   CMP     #125,R0         ;R0=125?
3727  010772  001404                           BEQ     TS132
3728                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3729                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3730                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3731                                                                   ;           WHICH FOLLOWS W/ 766           <====
3732  010774  012742  000240                   MOV     #240,-(R2)      ;MOVE TO MAILBOX #  *******  240  *******
3733  011000  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3734  011002  000000                           HALT                    ;BISB W/ MODE 3 - ODD BYTE FAILED
3735                                                                   ;  OR SEQUENCE ERROR
3736
3737                                    ;*****************************************************************************
3738                                    ;TEST 132       TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
3739                                    ;*****************************************************************************
3740  011004  005212                    TS132:  INC     (R2)            ;UPDATE TEST NUMBER
3741  011006  022712  000132                   CMP     #132,(R2)       ;SEQUENCE ERROR?
3742  011012  001017                           BNE     TS133-10        ;BR TO ERROR HALT ON SEQ ERROR
3743  011014  005000                           CLR     R0              ;R0=0
3744  011016  105100                           COMB    R0              ;R0-377
3745  011020  000263                           +SEC!SEV                ;SET C AND V BITS
3746  011022  132700  000200                   BITB    #200,R0         ;TRY DOPNM DEST. MODE 0-BYTE
3747  011026  001403                           BEQ     DNMB0A          ;BR TO ERROR IF Z BIT SET
3748  011030  102402                           BVS     DNMB0A          ;BR TO ERROR IF V BIT SET
3749  011032  103001                           BCC     DNMB0A          ;BR TO ERROR IF C BIT CLEAR.
3750  011034  100404                           BMI     DNMB0B
3751                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3752                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3753                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3754                                                                   ;           WHICH FOLLOWS W/ 766           <====
3755  011036                            DNMB0A:
3756  011036  012742  000241                   MOV     #241,-(R2)      ;MOVE TO MAILBOX #  *******  241  *******
3757  011042  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3758  011044  000000                           HALT                    ;CC'S INCORRECT
3759  011046  105100                    DNMB0B: COMB    R0              ;CHECK DESTINATION DATA
3760  011050  001404                           BEQ     TS133
3761                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3762                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3763                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3764                                                                   ;           WHICH FOLLOWS W/ 760           <====
3765  011052  012742  000242                   MOV     #242,-(R2)      ;MOVE TO MAILBOX #  *******  242  *******
```

```
3766  011056  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3767  011060  000000                        HALT                    ;DEST. DATA MODIFIED
3768                                                                ;  OR SEQUENCE ERROR
3769
3770                                ;*****************************************************************************
3771                                ;TEST 133       TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3772                                ;*****************************************************************************
3773  011062  005212                TS133:  INC     (R2)            ;UPDATE TEST NUMBER
3774  011064  022712  000133                CMP     #133,(R2)       ;SEQUENCE ERROR?
3775  011070  001017                        BNE     TS134-10        ;BR TO ERROR HALT ON SEQ ERROR
3776  011072  005000                        CLR     R0              ;R0=0
3777  011074  005010                        CLR     (R0)            ;LOC. 0=0
3778  011076  000241                        CLC                     ;CLEAR C BIT
3779  011100  032710  177777                BIT     #177777,(R0)    ;TRY DOPNM DEST. MODE 1
3780  011104  100403                        BMI     DNM1A           ;BR TO ERROR IF N BIT SET
3781  011106  102402                        BVS     DNM1A           ;BR TO ERROR IF V BIT SET
3782  011110  103401                        BCS     DNM1A           ;BR TO ERROR IF C BIT SET
3783  011112  001404                        BEQ     DNM1B
3784                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3785                                                                ;           CONDITIONAL BRANCH INST. AND   <
3786                                                                ;           REPLACE THE MOVE INSTRUCTION   <
3787                                                                ;           WHICH FOLLOWS W/ 766           <
3788  011114                        DNM1A:
3789  011114  012742  000243                MOV     #243,-(R2)      ;MOVE TO MAILBOX #  *******  243  *******
3790  011120  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3791  011122  000000                        HALT                    ;COND. CODES INCORRECT
3792  011124  005710                DNM1B:  TST     (R0)            ;CHECK TEST DATA
3793  011126  001404                        BEQ     TS134
3794                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3795                                                                ;           CONDITIONAL BRANCH INST. AND   <
3796                                                                ;           REPLACE THE MOVE INSTRUCTION   <
3797                                                                ;           WHICH FOLLOWS W/ 760           <-
3798  011130  012742  000244                MOV     #244,-(R2)      ;MOVE TO MAILBOX #  *******  244  *******
3799  011134  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3800  011136  000000                        HALT                    ;DESTINATION DATA MODIFIED
3801                                                                ;  OR SEQUENCE ERROR
3802
3803                                ;*****************************************************************************
3804                                ;TEST 134       TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
3805                                ;*****************************************************************************
3806  011140  005212                TS134:  INC     (R2)            ;UPDATE TEST NUMBER
3807  011142  022712  000134                CMP     #134,(R2)       ;SEQUENCE ERROR?
3808  011146  001027                        BNE     TS135-10        ;BR TO ERROR HALT ON SEQ ERROR
3809  011150  005000                        CLR     R0              ;R0=0
3810  011152  005010                        CLR     (R0)            ;LOC. 0=0
3811  011154  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125252
3812  011160  032720  077777                BIT     #77777,(R0)+    ;TRY DOPNM INST W/ MODE 2
3813  011164  102402                        BVS     DNM2A           ;BR TO ERROR IF V BIT SET
3814  011166  001401                        BEQ     DNM2A           ;BR TO ERROR IF Z-BIT SET
3815  011170  100004                        BPL     DNM2B
3816                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
3817                                                                ;           CONDITIONAL BRANCH INST. AND   <
3818                                                                ;           REPLACE THE MOVE INSTRUCTION   <
3819                                                                ;           WHICH FOLLOWS W/ 766           <
3820  011172                        DNM2A:
3821  011172  012742  000245                MOV     #245,-(R2)      ;MOVE TO MAILBOX #  *******  245  *******
```

```
3822  011176  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3823  011200  000000                        HALT                    ;COND. CODES INCORRECT
3824  011202  005300                DNM2B:  DEC     R0              ;DECREMENT R0 TO CHECK IT.
3825  011204  005300                        DEC     R0
3826  011206  001404                        BEQ     DNM2D
3827                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-- -
3828                                                                ;           CONDITIONAL BRANCH INST. AND   <---
3829                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3830                                                                ;           WHICH FOLLOWS W/ 757           <====
3831  011210                        DNM2C:
3832  011210  012742  000246                MOV     #246,-(R2)      ;MOVE TO MAILBOX #  *******  246  *******
3833  011214  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3834  011216  000000                        HALT                    ;MODE 2 REGISTER NOT INCREMENTED BY 2
3835  011220  022710  125252        DNM2D:  CMP     #125252,(R0)    ;CHECK DEST. DATA
3836  011224  001404                        BEQ     TS135
3837                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3838                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3839                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3840                                                                ;           WHICH FOLLOWS W/ 750           <====
3841  011226  012742  000247                MOV     #247,-(R2)      ;MOVE TO MAILBOX #  *******  247  *******
3842  011232  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3843  011234  000000                        HALT                    ;DEST. DATA MODIFIED
3844                                                                ;  OR SEQUENCE ERROR
3845
3846                                ;*************************************************************************************
3847                                ;TEST 135       TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
3848                                ;*************************************************************************************
3849  011236  005212                TS135:  INC     (R2)            ;UPDATE TEST NUMBER
3850  011240  022712  000135                CMP     #135,(R2)       ;SEQUENCE ERROR?
3851  011244  001051                        BNE     TS136-10        ;BR TO ERROR HALT ON SEQ ERROR
3852  011246  005000                        CLR     R0              ;R0=0
3853  011250  005010                        CLR     (R0)            ;LOC. 0=0
3854  011252  052710  052652                BIS     #52652,(R0)     ;LOC. 0=52652
3855  011256  000263                        +SEC!SEV                ;SET C AND V BITS
3856  011260  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
3857  011264  001403                        BEQ     DNMB2A          ;BR TO ERROR IF Z-BIT SET
3858  011266  103002                        BCC     DNMB2A          ;BR TO ERROR IF C-BIT CLEAR
3859  011270  102401                        BVS     DNMB2A          ;BR TO ERROR IF V-BIT SET
3860  011272  100404                        BMI     DNMB2B
3861                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3862                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3863                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3864                                                                ;           WHICH FOLLOWS W/ 764           <====
3865  011274                        DNMB2A:
3866  011274  012742  000250                MOV     #250,-(R2)      ;MOVE TO MAILBOX #  *******  250  *******
3867  011300  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3868  011302  000000                        HALT                    ;COND. CODES INCORRECT
3869  011304  005300                DNMB2B: DEC     R0              ;CHECK DEST. REGISTER.
3870  011306  001404                        BEQ     DNMB2C
3871                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3872                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3873                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3874                                                                ;           WHICH FOLLOWS W/ 756           <====
3875  011310  012742  000251                MOV     #251,-(R2)      ;MOVE TO MAILBOX #  *******  251  *******
3876  011314  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3877  011316  000000                        HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
```

```
3878  011320  005200                DNMB2C: INC     R0              ;R0=1
3879  011322  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/MODE 2-ODD BYTE
3880  011326  001402                        BEQ     DNMB2D  ;BR TO ERROR IF Z-BIT SET
3881  011330  102401                        BVS     DNMB2D          ;BR TO ERROR IF V-BIT SET
3882  011332  100004                        BPL     DNMB2E
3883                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= --
3884                                                                ;           CONDITIONAL BRANCH INST. AND   <-===
3885                                                                ;           REPLACE THE MOVE INSTRUCTION   <-===
3886                                                                ;           WHICH FOLLOWS W/ 744           <==-=
3887  011334                        DNMB2D:
3888  011334  012742  000252                MOV     #252,-(R2)      ;MOVE TO MAILBOX #  *******  252  *******
3889  011340  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3890  011342  000000                        HALT                    ;COND. CODES INCORRECT
3891  011344  005300                DNMB2E: DEC     R0              ;DEC R0 TO CHECK IT.
3892  011346  005300                        DEC     R0
3893  011350  001404                        BEQ     DNMB2F
3894                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3895                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3896                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3897                                                                ;           WHICH FOLLOWS W/ 735           <====
3898  011352  012742  000253                MOV     #253,-(R2)      ;MOVE TO MAILBOX #  *******  253  *******
3899  011356  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3900  011360  000000                        HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
3901  011362  022710  052652        DNMB2F: CMP     #52652,(R0)     ;CHECK DEST. DATA IS UNMODIFIED
3902  011366  001404                        BEQ     TS136
3903                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3904                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3905                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3906                                                                ;           WHICH FOLLOWS W/ 726           <====
3907  011370  012742  000254                MOV     #254,-(R2)      ;MOVE TO MAILBOX #  *******  254  *******
3908  011374  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3909  011376  000000                        HALT                    ;DEST. DATA WAS MODIFIED.
3910                                                                ;  OR SEQUENCE ERROR
3911
3912
3913                                ;*****************************************************************************************
3914                                ;TEST 136       TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
3915                                ;*****************************************************************************************
3916  011400  005212                TS136:  INC     (R2)            ;UPDATE TEST NUMBER
3917  011402  022712  000136                CMP     #136,(R2)       ;SEQUENCE ERROR?
3918  011406  001050                        BNE     TS137-10        ;BR TO ERROR HALT ON SEQ ERROR
3919  011410  005000                        CLR     R0              ;R0=0
3920  011412  005010                        CLR     (R0)            ;LOC. 0=0
3921  011414  052710  125125                BIS     #125125,(R0)    ;LOC. 0=125125
3922  011420  105100                        COMB    R0              ;R0=377
3923  011422  005200                        INC     R0              ;R0=400
3924  011424  005010                        CLR     (R0)            ;LOC. 400=0
3925  011426  000263                        +SEC!SEV                ;C-BIT=V-BIT=1
3926  011430  132730  000201                BITB    #201,@(R0)+     ;TRY DOPNM W/MODE 3-EVEN BYTE
3927  011434  001403                        BEQ     DNMB3A          ;BR TO ERROR IF Z BIT SET
3928  011436  102402                        BVS     DNMB3A          ;BR TO ERROR IF V BIT SET
3929  011440  103001                        BCC     DNMB3A          ;BR TO ERROR IF C BIT CLEAR
3930  011442  100004                        BPL     DNMB3B
3931                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3932                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3933                                                                ;           REPLACE THE MOVE INSTRUCTION   <=-=-
```

```
                                                        ;           WHICH FOLLOWS W/ 761          <====
011444                                  DNMB3A:
011444  012742  000255                          MOV     #255,-(R2)      ;MOVE TO MAILBOX #  *******  255  *******
011450  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011452  000000                                  HALT                    ;COND. CODES INCORRECT
011454  022700  000402                  DNMB3B: CMP     #402,R0         ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
011460  001404                                  BEQ     DNMB3C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
011462  012742  000256                          MOV     #256,-(R2)      ;MOVE TO MAILBOX #  *******  256  *******
011466  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011470  000000                                  HALT                    ;DEST. REGISTER NOT INCREMENTED BY 2
011472  005200                  DNMB3C: INC     R0              ;R0=404
011474  005200                          INC     R0
011476  132730  000201                  BITB    #201,@(R0)+     ;TRY DOPNM DEST MODE 3-BYTE(ODD)
011502  001402                          BEQ     DNMB3D          ;BR TO ERROR IF Z BIT SET
011504  102401                          BVS     DNMB3D          ;BR TO ERROR IF V BIT SET
011506  100404                          BMI     DNMB3E
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 737           <====
011510                                  DNMB3D:
011510  012742  000257                          MOV     #257,-(R2)      ;MOVE TO MAILBOX #  *******  257  *******
011514  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011516  000000                                  HALT                    ;COND. CODES INCORRECT
011520  005004                  DNMB3E: CLR     R4              ;R4=0
011522  022714  125125                  CMP     #125125,(R4)    ;CHECK DEST. DATA
011526  001404                          BEQ     TS137
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 727           <====
011530  012742  000260                          MOV     #260,-(R2)      ;MOVE TO MAILBOX #  *******  260  *******
011534  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011536  000000                                  HALT                    ;DEST. DATA MODIFIED
                                                                        ;  OR SEQUENCE ERROR

                        ;*******************************************************************************
                        ;TEST 137       TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
                        ;*******************************************************************************
011540  005212          TS137:  INC     (R2)            ;UPDATE TEST NUMBER
011542  022712  000137          CMP     #137,(R2)       ;SEQUENCE ERROR?
011546  001033                  BNE     TS140-10        ;BR TO ERROR HALT ON SEQ ERROR
011550  005000                  CLR     R0              ;R0=0
011552  005010                  CLR     (R0)            ;LOC. 0=0
011554  052710  125252          BIS     #125252,(R0)    ;LOC. 0=125125
011560  052700  000002          BIS     #2,R0           ;R0=2
011564  000277                  SCC                     ;SET ALL COND. CODE BITS
011566  032740  020000          BIT     #20000,-(R0)    ;TRY DOPNM W/ MODE 4
011572  100403                  BMI     DNM4A           ;BR TO ERROR IF N-BIT SET
011574  102402                  BVS     DNM4A           ;BR TO ERROR IF V-BIT SET
011576  103001                  BCC     DNM4A           ;BR TO ERROR IF C-BIT CHAR
011600  001004                  BNE     DNM4B
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
011602                          DNM4A:
011602  012742  000261                  MOV     #261,-(R2)      ;MOVE TO MAILBOX #  *******  261  *******
011606  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011610  000000                          HALT                    ;COND. CODES INCORRECT
011612  005700                  DNM4B:  TST     R0              ;CHECK DEST. REGISTER
011614  001404                          BEQ     DNM4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
011616  012742  000262                  MOV     #262,-(R2)      ;MOVE TO MAILBOX #  *******  262  *******
011622  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011624  000000                          HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
011626  022737  125252  000000  DNM4C:  CMP     #125252,@#0     ;CHECK DEST. DATA
011634  001404                          BEQ     TS140
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 744           <====
011636  012742  000263                  MOV     #263,-(R2)      ;MOVE TO MAILBOX #  *******  263  *******
011642  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011644  000000                          HALT                    ;DEST. DATA MODIFIED
                                                        ;  OR SEQUENCE ERROR

;*****************************************************************************************
;TEST 140       TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
;*****************************************************************************************
011646  005212                  TS140:  INC     (R2)            ;UPDATE TEST NUMBER
011650  022712  000140                  CMP     #140,(R2)       ;SEQUENCE ERROR?
011654  001051                          BNE     TS141-10        ;BR TO ERROR HALT ON SEQ ERROR
011656  005000                          CLR     R0              ;R0=0
011660  005010                          CLR     (R0)            ;LOC. 0=0
011662  052710  052652                  BIS     #52652,(R0)     ;LOC. 0=52652
011666  052700  000002                  BIS     #2,R0           ;R0=2
011672  000257                          CCC                     ;COND. CODES=0
011674  132740  000201                  BITB    #201,-(R0)      ;TRY DOPNM INST W/MODE 4 ODD BYTE
011700  102403                          BVS     DNMB4A          ;BR TO ERROR IF V BIT SET
011702  001402                          BEQ     DNMB4A  ;BR TO ERROR IF Z BIT SET
011704  103401                          BCS     DNMB4A          ;BR TO ERROR IF C BIT SET
011706  001004                          BNE     DNMB4B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
011710                          DNMB4A:
011710  012742  000264                  MOV     #264,-(R2)      ;MOVE TO MAILBOX #  *******  264  *******
011714  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011716  000000                          HALT                    ;COND. CODES INCORRECT
011720  022700  000001          DNMB4B: CMP     #1,R0           ;CHECK DEST. REGISTER
011724  001404                          BEQ     DNMB4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
```

```
4046                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4047                                                    ;           WHICH FOLLOWS W/ 753           <====
4048  011726  012742  000265          MOV     #265,-(R2)      ;MOVE TO MAILBOX #  *******  265  *******
4049  011732  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4050  011734  000000                  HALT                    ;DEST REG. NOT DECREMENTED BY 1
4051  011736  132740  000201  DNMB4C: BITB    #201,-(R0)      ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
4052  011742  001401                  BEQ     DNMB4D          ;BR TO ERROR IF Z-BIT SET
4053  011744  100404                  BMI     DNMB4E
4054                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4055                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4056                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4057                                                    ;           WHICH FOLLOWS W/ 743           <====
4058  011746                  DNMB4D:
4059  011746  012742  000266          MOV     #266,-(R2)      ;MOVE TO MAILBOX #  *******  266  *******
4060  011752  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4061  011754  000000                  HALT                    ;COND. CODES INCORRECT
4062  011756  005700  DNMB4E: TST     R0              ;CHECK DEST. REGISTER
4063  011760  001404                  BEQ     DNMB4F
4064                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4065                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4066                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4067                                                    ;           WHICH FOLLOWS W/ 735           <====
4068  011762  012742  000267          MOV     #267,-(R2)      ;MOVE TO MAILBOX #  *******  267  *******
4069  011766  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4070  011770  000000                  HALT                    ;DEST. REG. NOT DECREMENTED BY 1
4071  011772  022710  052652  DNMB4F: CMP     #52652,(R0)     ;CHECK DESTINATION DATA
4072  011776  001404                  BEQ     TS141
4073                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4074                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4075                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4076                                                    ;           WHICH FOLLOWS W/ 726           <====
4077  012000  012742  000270          MOV     #270,-(R2)      ;MOVE TO MAILBOX #  *******  270  *******
4078  012004  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4079  012006  000000                  HALT                    ;DEST. DATA MODIFIED
4080                                                    ;  OR SEQUENCE ERROR
4081
4082  ;*****************************************************************************************
4083  ;TEST 141       TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
4084  ;*****************************************************************************************
4085  012010  005212  TS141:  INC     (R2)            ;UPDATE TEST NUMBER
4086  012012  022712  000141          CMP     #141,(R2)       ;SEQUENCE ERROR?
4087  012016  001034                  BNE     TS142-10        ;BR TO ERROR HALT ON SEQ ERROR
4088  012020  005000                  CLR     R0              ;R0=0
4089  012022  005010                  CLR     (R0)            ;LOC 0=0
4090  012024  052710  100000          BIS     #100000,(R0)    ;LOC. 0=100000
4091  012030  052700  000402          BIS     #402,R0         ;R0=2
4092  012034  000277                  SCC                     ;SET ALL COND. CODE BITS
4093  012036  032750  100000          BIT     #100000,@-(R0)  ;TRY DOPNM W/MODE 5
4094  012042  102403                  BVS     DNM5A           ;BR TO ERROR IF V-BIT SET
4095  012044  103002                  BCC     DNM5A           ;BR TO ERROR IF C-BIT CLEAR
4096  012046  001401                  BEQ     DNM5A           ;BR TO ERROR IF Z-BIT SET
4097  012050  100404                  BMI     DNM5B
4098                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4099                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4100                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4101                                                    ;           WHICH FOLLOWS W/ 762           <====
```

```
4102  012052                          DNM5A:
4103  012052  012742  000271                  MOV     #271,-(R2)      ;MOVE TO MAILBOX #  *******  271  *******
4104  012056  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4105  012060  000000                          HALT                    ;COND. CODES INCORRECT
4106  012062  022700  000400        DNM5B:  CMP #400,R0             ;CHECK DEST. REGISTER
4107  012066  001404                          BEQ     DNM5C
4108                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4109                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4110                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4111                                                                  ;           WHICH FOLLOWS W/ 753           <====
4112  012070  012742  000272                  MOV     #272,-(R2)      ;MOVE TO MAILBOX #  *******  272  *******
4113  012074  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4114  012076  000000                          HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
4115  012100  022737  100000  000000  DNM5C:  CMP     #100000,@#0     ;CHECK DESTINATION DATA
4116  012106  001404                          BEQ     TS142
4117                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4118                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4119                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4120                                                                  ;           WHICH FOLLOWS W/ 743           <====
4121  012110  012742  000273                  MOV     #273,-(R2)      ;MOVE TO MAILBOX #  *******  273  *******
4122  012114  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4123  012116  000000                          HALT                    ;DEST. DATA INCORRECTLY MODIFIED
4124                                                                  ;  OR SEQUENCE ERROR
4125
4126                                  ;*****************************************************************************************
4127                                  ;TEST 142       TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
4128                                  ;*****************************************************************************************
4129  012120  005212                  TS142:  INC     (R2)            ;UPDATE TEST NUMBER
4130  012122  022712  000142                  CMP     #142,(R2)       ;SEQUENCE ERROR?
4131  012126  001033                          BNE     TS143-10        ;BR TO ERROR HALT ON SEQ ERROR
4132  012130  005000                          CLR     R0              ;R0=0
4133  012132  005010                          CLR     (R0)            ;LOC> 0=0
4134  012134  052710  000001                  BIS     #1,(R0)         ;LOC. 0=1
4135  012140  005100                          COM     R0              ;R0=-1 C-BIT=1
4136  012142  032760  000001  000001          BIT     #1,1(R0)        ;TRY DOPNM W/MODE 6
4137  012150  001403                          BEQ     DNM6A           ;BR TO ERROR IF Z-BIT SET
4138  012152  102402                          BVS     DNM6A           ;BR TO ERROR IF V-BIT SET
4139  012154  103001                          BCC     DNM6A           ;BR TO ERROR IF C-BIT CLEAR
4140  012156  100004                          BPL     DNM6B
4141                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4142                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4143                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4144                                                                  ;           WHICH FOLLOWS W/ 763           <====
4145  012160                          DNM6A:
4146  012160  012742  000274                  MOV     #274,-(R2)      ;MOVE TO MAILBOX #  *******  274  *******
4147  012164  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4148  012166  000000                          HALT                    ;COND CODES INCORRECT
4149  012170  022700  177777        DNM6B:  CMP     #-1,R0          ;CHECK DEST. REGISTER
4150  012174  001404                          BEQ     DNM6C
4151                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4152                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4153                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4154                                                                  ;           WHICH FOLLOWS W/ 754           <====
4155  012176  012742  000275                  MOV     #275,-(R2)      ;MOVE TO MAILBOX #  *******  275  *******
4156  012202  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4157  012204  000000                          HALT                    ;DEST. REGISTER MODIFIED
```

```
4158  012206  022737  000001  000000  DNM6C:  CMP     #1,@#0          ;CHECK DEST. DATA
4159  012214  001404                  BEQ     TS143
4160                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
4161                                            ;           CONDITIONAL BRANCH INST. AND   <== =
4162                                            ;           REPLACE THE MOVE INSTRUCTION   <-==-
4163                                            ;           WHICH FOLLOWS W/ 744           <====
4164  012216  012742  000276                  MOV     #276,-(R2)      ;MOVE TO MAILBOX #  *******  276  *******
4165  012222  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4166  012224  000000                  HALT                    ;DEST. DATA MODIFIED
4167                                            ;  OR SEQUENCE ERROR
4168
4169                            ;*****************************************************************************
4170                            ;TEST 143       TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
4171                            ;*****************************************************************************
4172  012226  005212            TS143:  INC     (R2)            ;UPDATE TEST NUMBER
4173  012230  022712  000143            CMP     #143,(R2)       ;SEQUENCE ERROR?
4174  012234  001034                    BNE     TS144-10        ;BR TO ERROR HALT ON SEQ ERROR
4175  012236  005000                    CLR     R0              ;R0=0
4176  012240  005010                    CLR     (R0)            ;LOC. 0=0 C-BIT-0
4177  012242  052710  125125            BIS     #125125,(R0)    ;LOC. 0=125125
4178  012246  052700  000001            BIS     #1,R0           ;R0=1
4179  012252  132770  000125  000403    BITB    #125,@403(R0)   ;TRY DOPNM W/MODE 7
4180  012260  102403                    BVS     DNM7A           ;BR TO ERROR IF V-BIT SET
4181  012262  100402                    BMI     DNM7A           ;BR TO ERROR IF N-BIT SET
4182  012264  103401                    BCS     DNM7A           ;BR TO ERROR IF C-BIT SET
4183  012266  001404                    BEQ     DNM7B
4184                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4185                                            ;           CONDITIONAL BRANCH INST. AND   <====
4186                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4187                                            ;           WHICH FOLLOWS W/ 762           <====
4188  012270            DNM7A:
4189  012270  012742  000277                  MOV     #277,-(R2)      ;MOVE TO MAILBOX #  *******  277  *******
4190  012274  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4191  012276  000000                  HALT                    ;COND. CODES INCORRECT
4192  012300  022700  000001    DNM7B:  CMP     #1,R0           ;CHECK DEST. REGISTER
4193  012304  001404                  BEQ     DNM7C
4194                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4195                                            ;           CONDITIONAL BRANCH INST. AND   <====
4196                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4197                                            ;           WHICH FOLLOWS W/ 753           <====
4198  012306  012742  000300                  MOV     #300,-(R2)      ;MOVE TO MAILBOX #  *******  300  *******
4199  012312  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4200  012314  000000                  HALT                    ;DESTINATION REGISTER MODIFIED
4201  012316  022737  125125  000000  DNM7C:  CMP     #125125,@#0     ;CHECK DEST. DATA
4202  012324  001404                  BEQ     TS144
4203                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4204                                            ;           CONDITIONAL BRANCH INST. AND   <====
4205                                            ;           REPLACE THE MOVE INSTRUCTION   <==-=
4206                                            ;           WHICH FOLLOWS W/ 743           <===-
4207  012326  012742  000301                  MOV     #301,-(R2)      ;MOVE TO MAILBOX #  *******  301  *******
4208  012332  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4209  012334  000000                  HALT                    ;DEST. DATA INCORRECT
4210                                            ;  OR SEQUENCE ERROR
4211
4212                            ;*****************************************************************************
4213                            ;
```

```
                        ;       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 144       TEST MOV DESTINATION MODE 1
                        ;*****************************************************************************
012336  005212          TS144:  INC     (R2)            ;UPDATE TEST NUMBER
012340  022712  000144          CMP     #144,(R2)       ;SEQUENCE ERROR?
012344  001016                  BNE     TS145-10        ;BR TO ERROR HALT ON SEQ ERROR
012346  005000                  CLR     R0              ;R0=0
012350  005010                  CLR     (R0)            ;LOC. 0=0
012352  005100                  COM     R0              ;R0=-1
012354  005004                  CLR     R4              ;R4 POINTS TO LOC. 0
012356  010014                  MOV     R0,(R4)         ;TRY MOVE MODE 0,1
012360  102402                  BVS     MDM1A           ;BR TO ERROR IF V SET
012362  001401                  BEQ     MDM1A           ;BR TO ERROR IF Z SET
012364  100404                  BMI     MDM1B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
012366          MDM1A:
012366  012742  000302          MOV     #302,-(R2)      ;MOVE TO MAILBOX #  *******  302  *******
012372  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012374  000000                  HALT                    ;CONDITION CODE NOT CORRECT
012376  005704          MDM1B:  TST     R4
012400  001404                  BEQ     TS145
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
012402  012742  000303          MOV     #303,-(R2)      ;MOVE TO MAILBOX #  *******  303  *******
012406  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012410  000000                  HALT                    ;DESTINATION REGISTER INCORRECTLY ALTERED
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;
                        ;       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 145       TEST MOV DESTINATION MODE 2
                        ;*****************************************************************************
012412  005212          TS145:  INC     (R2)            ;UPDATE TEST NUMBER
012414  022712  000145          CMP     #145,(R2)       ;SEQUENCE ERROR?
012420  001026                  BNE     TS146-10        ;BR TO ERROR HALT ON SEQ ERROR
012422  005000                  CLR     R0              ;R0=0
012424  005001                  CLR     R1              ;R1=0
012426  005010                  CLR     (R0)            ;LOC. 0=0
012430  005110                  COM     (R0)            ;LOC. 0=-1
012432  010120                  MOV     R1,(R0)+        ;TRY MOVE MODE 0,2
012434  100402                  BMI     MDM2A           ;BR TO ERROR IF N SET
012436  102401                  BVS     MDM2A           ;BR TO ERROR IF V SET
```

```
4270  012440  001404                        BEQ     MDM2B
4271                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4272                                                          ;           CONDITIONAL BRANCH INST. AND   <====
4273                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
4274                                                          ;           WHICH FOLLOWS W/ 767           <====
4275  012442                            MDM2A:
4276  012442  012742  000304                  MOV     #304,-(R2)      ;MOVE TO MAILBOX #  *******  304  *******
4277  012446  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4278  012450  000000                        HALT                    ;CC'S INCORRECT
4279  012452  005300                    MDM2B:  DEC     R0
4280  012454  005300                        DEC     R0
4281  012456  001404                        BEQ     MDM2D
4282                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4283                                                          ;           CONDITIONAL BRANCH INST. AND   <====
4284                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
4285                                                          ;           WHICH FOLLOWS W/ 760           <====
4286  012460                            MDM2C:
4287  012460  012742  000305                  MOV     #305,-(R2)      ;MOVE TO MAILBOX #  *******  305  *******
4288  012464  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4289  012466  000000                        HALT                    ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
4290  012470  005737  000000          MDM2D:  TST     @#0
4291  012474  001404                        BEQ     TS146
4292                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4293                                                          ;           CONDITIONAL BRANCH INST. AND   <====
4294                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
4295                                                          ;           WHICH FOLLOWS W/ 751           <====
4296  012476  012742  000306                  MOV     #306,-(R2)      ;MOVE TO MAILBOX #  *******  306  *******
4297  012502  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4298  012504  000000                        HALT                    ;DESTINATION DATA INCORRECT
4299                                                          ;  OR SEQUENCE ERROR
4300
4301                                ;*****************************************************************************
4302                                ;
4303                                ;       THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
4304                                ;INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
4305                                ;
4306                                ;*****************************************************************************
4307                                ;TEST 146       TEST MOV-BYTE DESTINATION MODE 2
4308                                ;*****************************************************************************
4309  012506  005212                TS146:  INC     (R2)            ;UPDATE TEST NUMBER
4310  012510  022712  000146               CMP     #146,(R2)       ;SEQUENCE ERROR?
4311  012514  001046                       BNE     TS147-10        ;BR TO ERROR HALT ON SEQ ERROR
4312  012516  005000                       CLR     R0              ;R0=0
4313  012520  005010                       CLR     (R0)            ;LOC. 0=0
4314  012522  112720  000125               MOVB    #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
4315  012526  102402                       BVS     MBDM2A          ;BR TO ERROR IF V SET
4316  012530  001401                       BEQ     MBDM2A          ;BR TO ERROR IF Z SET
4317  012532  100004                       BPL     MBDM2B
4318                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4319                                                          ;           CONDITIONAL BRANCH INST. AND   <====
4320                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
4321                                                          ;           WHICH FOLLOWS W/ 770           <====
4322  012534                            MBDM2A:
4323  012534  012742  000307                  MOV     #307,-(R2)      ;MOVE TO MAILBOX #  *******  307  *******
4324  012540  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4325  012542  000000                        HALT                    ;CC'S INCORRECT
```

```
4326  012544  022700  000001          MBDM2B: CMP     #1,R0
4327  012550  001404                  BEQ     MBDM2C
4328                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4329                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4330                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4331                                                    ;           WHICH FOLLOWS W/ 761           <====
4332  012552  012742  000310                  MOV     #310,-(R2)  ;MOVE TO MAILBOX #  *******  310  *******
4333  012556  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
4334  012560  000000                  HALT                ;REGISTER NOT INCREMENTED BY ONE
4335  012562  112720  000252          MBDM2C: MOVB    #252,(R0)+  ;TRY DESTINATION MODE 2 W/ODD BYTE
4336  012566  102402                  BVS     MBDM2D
4337  012570  001401                  BEQ     MBDM2D
4338  012572  100404                  BMI     MBDM2E
4339                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4340                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4341                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4342                                                    ;           WHICH FOLLOWS W/ 750           <====
4343  012574                          MBDM2D:
4344  012574  012742  000311                  MOV     #311,-(R2)  ;MOVE TO MAILBOX #  *******  311  *******
4345  012600  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
4346  012602  000000                  HALT                ;CC'S NOT SET CORRECT
4347  012604  022700  000002          MBDM2E: CMP     #2,R0
4348  012610  001404                  BEQ     MBDM2F
4349                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4350                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4351                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4352                                                    ;           WHICH FOLLOWS W/ 741           <====
4353  012612  012742  000312                  MOV     #312,-(R2)  ;MOVE TO MAILBOX #  *******  312  *******
4354  012616  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
4355  012620  000000                  HALT                ;REGISTER NOT INCREMENTED BY ONE
4356  012622  022737  125125  000000  MBDM2F: CMP     #125125,@#0 ;CHECK DATA
4357  012630  001404                  BEQ     TS147
4358                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4359                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4360                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4361                                                    ;           WHICH FOLLOWS W/ 731           <====
4362  012632  012742  000313                  MOV     #313,-(R2)  ;MOVE TO MAILBOX #  *******  313  *******
4363  012636  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
4364  012640  000000                  HALT                ;DESTINATION DATA INCORRECT
4365                                                    ;  OR SEQUENCE ERROR
4366
4367                        ;*************************************************************************************
4368                        ;
4369                        ;       THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
4370                        ;AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB
4371                        ;INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
4372                        ;*************************************************************************************
4373                        ;TEST 147       TEST MOV(B) DESTINATION MODE 3
4374                        ;*************************************************************************************
4375  012642  005212        TS147:  INC     (R2)            ;UPDATE TEST NUMBER
4376  012644  022712  000147          CMP     #147,(R2)       ;SEQUENCE ERROR?
4377  012650  001057                  BNE     TS150-10        ;BR TO ERROR HALT ON SEQ ERROR
4378  012652  012700  000400          MOV     #400,R0         ;R0=400
4379  012656  005010                  CLR     (R0)            ;LOC. 400 POINTS TO LOC. 0
4380  012660  005037  000000          CLR     @#0             ;LOC. 0=0
4381  012664  012730  125252          MOV     #125252,@(R0)+  ;TRY MOV DESTINATION MODE 2
```

```
4382  012670  102402                        BVS     MDM3A           ;BR TO ERROR IF V SET
4383  012672  001401                        BEQ     MDM3A           ;BR TO ERROR IF Z SET
4384  012674  100404                        BMI     MDM3B
4385                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4386                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4387                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4388                                                                  ;           WHICH FOLLOWS W/ 765           <====
4389  012676                          MDM3A:
4390  012676  012742  000314                MOV     #314,-(R2)      ;MOVE TO MAILBOX #  *******  314  *******
4391  012702  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4392  012704  000000                        HALT                    ;CC'S INCORRECT
4393  012706  022700  000402        MDM3B:  CMP     #402,R0         ;CHECK DEST. MODE REGISTER
4394  012712  001404                        BEQ     MDM3C
4395                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4396                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4397                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4398                                                                  ;           WHICH FOLLOWS W/ 756           <====
4399  012714  012742  000315                MOV     #315,-(R2)      ;MOVE TO MAILBOX #  *******  315  *******
4400  012720  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4401  012722  000000                        HALT                    ;REGISTER NOT INCREMENTED BY 2
4402  012724  022737  125252  000000  MDM3C:  CMP     #125252,@#0     ;CHECK DESTINATION DATA
4403  012732  001404                        BEQ     MDM3D
4404                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4405                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4406                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4407                                                                  ;           WHICH FOLLOWS W/ 746           <====
4408  012734  012742  000316                MOV     #316,-(R2)      ;MOVE TO MAILBOX #  *******  316  *******
4409  012740  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4410  012742  000000                        HALT                    ;DESTINATION DATA INCORRECT
4411  012744  112737  000125  000000  MDM3D:  MOVB    #125,@#0        ;TRY MOVB DESTINATION MODE 2 EVEN BYTE
4412  012752  022737  125125  000000        CMP     #125125,@#0     ;CHECK DATA
4413  012760  001404                        BEQ     MDM3E
4414                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4415                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4416                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4417                                                                  ;           WHICH FOLLOWS W/ 733           <====
4418  012762  012742  000317                MOV     #317,-(R2)      ;MOVE TO MAILBOX #  *******  317  *******
4419  012766  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4420  012770  000000                        HALT                    ;DESTINATION DATA INCORRECT
4421  012772  112737  000525  000001  MDM3E:  MOVB    #525,@#1        ;TRY MOVB DESTINATION MODE 2 ODD BYTE
4422  013000  022737  052525  000000        CMP     #52525,@#0      ;CHECK DATA
4423  013006  001404                        BEQ     TS150
4424                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4425                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4426                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4427                                                                  ;           WHICH FOLLOWS W/ 720           <====
4428  013010  012742  000320                MOV     #320,-(R2)      ;MOVE TO MAILBOX #  *******  320  *******
4429  013014  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4430  013016  000000                        HALT                    ;
4431
4432                          ;*************************************************************************************
4433                          ;
4434                          ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4435                          ;SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4436                          ;R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4437                          ;CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
```

```
                                ;
                                ;*************************************************************************************
                                ;TEST 150       TEST MOV DESTINATION MODE 4
                                ;*************************************************************************************
013020  005212                  TS150:  INC     (R2)            ;UPDATE TEST NUMBER
013022  022712  000150                  CMP     #150,(R2)       ;SEQUENCE ERROR?
013026  001026                          BNE     TS151-10        ;BR TO ERROR HALT ON SEQ ERROR
013030  005000                          CLR     R0              ;R0=0
013032  005010                          CLR     (R0)            ;LOC 0=0
013034  012704  000002                  MOV     #2,R4           ;R4=2
013040  012744  012345                  MOV     #12345,-(R4)    ;TRY MOV DEST. MODE 4
013044  102402                          BVS     MDM4A           ;BR TO ERROR IF V-BIT SET
013046  001401                          BEQ     MDM4A           ;BR TO ERROR IF Z-BIT SET
013050  100004                          BPL     MDM4B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
013052                          MDM4A:
013052  012742  000321                  MOV     #321,-(R2)      ;MOVE TO MAILBOX #  *******  321  *******
013056  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013060  000000                          HALT                    ;CC'S NOT CORRECT
013062  005704                  MDM4B:  TST     R4              ;CHECK DECREMENTING OF MODE 4 REG.
013064  001404                          BEQ     MDM4C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
013066  012742  000322                  MOV     #322,-(R2)      ;MOVE TO MAILBOX #  *******  322  *******
013072  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013074  000000                          HALT                    ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
013076  022710  012345          MDM4C:  CMP     #12345,(R0)     ;CHECK DESTINATION DATA
013102  001404                          BEQ     TS151
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 751           <====
013104  012742  000323                  MOV     #323,-(R2)      ;MOVE TO MAILBOX #  *******  323  *******
013110  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013112  000000                          HALT                    ;DESTINATION DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR

                                ;*************************************************************************************
                                ;
                                ;       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 151       TEST MOVB DESTINATION MODE 4
                                ;*************************************************************************************
013114  005212                  TS151:  INC     (R2)            ;UPDATE TEST NUMBER
013116  022712  000151                  CMP     #151,(R2)       ;SEQUENCE ERROR?
013122  001046                          BNE     TS152-10        ;BR TO ERROR HALT ON SEQ ERROR
```

```
4494  013124  005004                    CLR     R4              ;R4=0
4495  013126  005014                    CLR     (R4)            ;LOC. 0=0
4496  013130  012700  000002            MOV     #2,R0           ;R0 = 2
4497  013134  112740  125125            MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4-ODD BYTE
4498  013140  020027  000001            CMP     R0,#1           ;CHECK THAT DEST. REG. WAS DECREMENTED
4499  013144  001404                    BEQ     MBDM4A
4500                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4501                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4502                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4503                                                            ;           WHICH FOLLOWS W/ 766           <====
4504  013146  012742  000324            MOV     #324,-(R2)      ;MOVE TO MAILBOX #  *******  324  *******
4505  013152  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4506  013154  000000                    HALT                    ;DESTINATION REG. NOT DECREMENTED BY 1
4507  013156  021427  052400  MBDM4A:   CMP     (R4),#52400     ;CHECK DEST. DATA
4508  013162  001404                    BEQ     MBDM4B
4509                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4510                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4511                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4512                                                            ;           WHICH FOLLOWS W/ 757           <====
4513  013164  012742  000325            MOV     #325,-(R2)      ;MOVE TO MAILBOX #  *******  325  *******
4514  013170  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4515  013172  000000                    HALT                    ;DEST. DATA NOT CORRECT
4516  013174  112740  125125  MBDM4B:   MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4--EVEN BYTE
4517  013200  102402                    BVS     MBDM4C          ;BR. TO ERROR IF V-BIT SET
4518  013202  001401                    BEQ     MBDM4C          ;BR TO ERROR IF Z-BIT SET
4519  013204  100004                    BPL     MBDM4D
4520                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4521                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4522                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4523                                                            ;           WHICH FOLLOWS W/ 746           <====
4524  013206            MBDM4C:
4525  013206  012742  000326            MOV     #326,-(R2)      ;MOVE TO MAILBOX #  *******  326  *******
4526  013212  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4527  013214  000000                    HALT                    ;COND. CODES INCORRECT
4528  013216  005700  MBDM4D:   TST     R0              ;CHECK MODE 4 DEST. REGISTER
4529  013220  001404                    BEQ     MBDM4E
4530                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4531                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4532                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4533                                                            ;           WHICH FOLLOWS W/ 740           <====
4534  013222  012742  000327            MOV     #327,-(R2)      ;MOVE TO MAILBOX #  *******  327  *******
4535  013226  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4536  013230  000000                    HALT                    ;DESTINATION REG NOT DECREMENTED BY 1
4537  013232  021427  052525  MBDM4E:   CMP     (R4),#52525     ;CHECK DEST. DAT^A
4538  013236  001404                    BEQ     TS152
4539                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4540                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4541                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4542                                                            ;           WHICH FOLLOWS W/ 731           <====
4543  013240  012742  000330            MOV     #330,-(R2)      ;MOVE TO MAILBOX #  *******  330  *******
4544  013244  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4545  013246  000000                    HALT                    ;DESTINATION DATA INCORRECT
4546                                                            ;  OR SEQUENCE ERROR
4547
4548              ;*****************************************************************************************
4549              ;
```

```
                                ;       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 152       TEST MOV DESTINATION MODE 5
                                ;*****************************************************************************************
013250  005212                TS152:  INC     (R2)            ;UPDATE TEST NUMBER
013252  022712  000152                CMP     #152,(R2)       ;SEQUENCE ERROR?
013256  001051                        BNE     TS153-10        ;BR TO ERROR HALT ON SEQ ERROR
013260  005004                        CLR     R4              ;R4=0
013262  005014                        CLR     (R4)            ;LOC. 0 = 0
013264  012700  000400                MOV     #400,R0         ;R0=400
013270  012750  004321                MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
013274  102402                        BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
013276  001401                        BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
013300  100004                        BPL     MDM5B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- ==
                                                            ;           CONDITIONAL BRANCH INST. AND   <==--
                                                            ;           REPLACE THE MOVE INSTRUCTION   <===-
                                                            ;           WHICH FOLLOWS W/ 766           <=-==
013302                          MDM5A:
013302  012742  000331                MOV     #331,-(R2)      ;MOVE TO MAILBOX #  *******  331  *******
013306  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013310  000000                        HALT                    ;COND. CODES INCORRECT
013312  022700  000376        MDM5B:  CMP     #376,R0         ;CHECK MODE 5 REG. WAS DECREMENTED
013316  001404                        BEQ     MDM5C
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <=-==
                                                            ;           REPLACE THE MOVE INSTRUCTION   <=---
                                                            ;           WHICH FOLLOWS W/ 757           <====
013320  012742  000332                MOV     #332,-(R2)      ;MOVE TO MAILBOX #  *******  332  *******
013324  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013326  000000                        HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
013330  022714  004321        MDM5C:  CMP     #4321,(R4)      ;CHECK DEST. DATA
013334  001404                        BEQ     MDM5D
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 750           <====
013336  012742  000333                MOV     #333,-(R2)      ;MOVE TO MAILBOX #  *******  333  *******
013342  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013344  000000                        HALT                    ;DEST. DATA INCORRECT
013346  012700  000406        MDM5D:  MOV     #406,R0         ;R0-406
013352  112750  000377                MOVB    #377,@-(R0)     ;TRY MOV DEST. MODE 5 --EVEN BYTE
013356  022700  000404                CMP     #404,R0         ;CHECK MODE 5 REG.
013362  001404                        BEQ     MDM5E
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 735           <====
013364  012742  000334                MOV     #334,-(R2)      ;MOVE TO MAILBOX #  *******  334  *******
013370  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
4606  013372  000000                        HALT                            ;MODE 5 REGISTER NOT DECREMENTED BY 2
4607  013374  022714  177721        MDM5E:  CMP     #177721,(R4)            ;CHECK DEST. DATA
4608  013400  001404                        BEQ     TS153
4609                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <  -=
4610                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
4611                                                                        ;           REPLACE THE MOVE INSTRUCTION   <==--
4612                                                                        ;           WHICH FOLLOWS W/ 726           <=-==
4613  013402  012742  000335                MOV     #335,-(R2)              ;MOVE TO MAILBOX #  *******  335  *******
4614  013406  005242                        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4615  013410  000000                        HALT                            ;DEST. DATA INCORRECT
4616                                                                        ;  OR SEQUENCE ERROR
4617
4618                                  ;*************************************************************************************
4619                                  ;
4620                                  ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE
4621                                  ;DESTINATION MODE 6 INDTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0
4622                                  ;FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0
4623                                  ;BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY
4624                                  ;PROPER ADDRESSING AND DATA.
4625                                  ;
4626                                  ;*************************************************************************************
4627                                  ;TEST 153       TEST MOV DESTINATION MODE 6
4628                                  ;*************************************************************************************
4629  013412  005212                  TS153:  INC     (R2)                    ;UPDATE TEST NUMBER
4630  013414  022712  000153                 CMP     #153,(R2)               ;SEQUENCE ERROR?
4631  013420  001054                         BNE     TS154-10                ;BR TO ERROR HALT ON SEQ ERROR
4632  013422  005000                         CLR     R0                      ;R0=0
4633  013424  005010                         CLR     (R0)                    ;LOC. 0=0
4634  013426  005200                         INC     R0                      ;R0=1
4635  013430  012760  052525  177777         MOV     #052525,-1(R0)          ;TRY MOV DEST. MODE 6
4636  013436  102402                         BVS     MDM6A                   ;BR TO ERROR IF V-BIT SET
4637  013440  001401                         BEQ     MDM6A                   ;BR TO ERROR IF Z-BIT SET
4638  013442  100004                         BPL     MDM6B
4639                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4640                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
4641                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
4642                                                                        ;           WHICH FOLLOWS W/ 766           <====
4643  013444                    MDM6A:
4644  013444  012742  000336                MOV     #336,-(R2)              ;MOVE TO MAILBOX #  *******  336  *******
4645  013450  005242                        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4646  013452  000000                        HALT                            ;COND. CODES INCORRECT
4647  013454  022700  000001        MDM6B:  CMP     #1,R0                   ;CHECK DEST. REGISTER UNALTERED
4648  013460  001404                        BEQ     MDM6C
4649                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
4650                                                                        ;           CONDITIONAL BRANCH INST. AND   <===-
4651                                                                        ;           REPLACE THE MOVE INSTRUCTION   <==-=
4652                                                                        ;           WHICH FOLLOWS W/ 757           <-===
4653  013462  012742  000337                MOV     #337,-(R2)              ;MOVE TO MAILBOX #  *******  337  *******
4654  013466  005242                        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4655  013470  000000                        HALT                            ;DEST. REGISTER INCORRECTLY ALTERED
4656  013472  022737  052525  000000 MDM6C:  CMP     #52525,@#0              ;CHECK DEST. DATA
4657  013500  001404                        BEQ     MDM6D
4658                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
4659                                                                        ;           CONDITIIUNAL BRANCH INST. AND  <==-=
4660                                                                        ;           REPLACE THE MOVE INSTRUCTION   <--=-
4661                                                                        ;           WHICH FOLLOWS W/ 747           <==
```

```
4662  013502  012742  000340                MOV     #340,-(R2)      ;MOVE TO MAILBOX #  *******  340  *******
4663  013506  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4664  013510  000000                       HALT                    ;DEST. DATA INCORRECT
4665  013512  012700  000002        MDM6D: MOV     #2,R0           ;R0=2
4666  013516  112760  000377  177777      MOVB    #377,-1(R0)     ;TRY MOVB DEST. MODE 6
4667  013524  022700  000002              CMP     #2,R0           ;CHECK DEST. REGISTER UNALTERED
4668  013530  001404                       BEQ     MDM6E
4669                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4670                                                              ;           CONDITIONAL BRANCH INST. AND   <====
4671                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
4672                                                              ;           WHICH FOLLOWS W/ 733           <====
4673  013532  012742  000341                MOV     #341,-(R2)      ;MOVE TO MAILBOX #  *******  341  *******
4674  013536  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4675  013540  000000                       HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4676  013542  022737  177525  000000 MDM6E: CMP     #177525,@#0     ;CHECK DEST. DATA
4677  013550  001404                       BEQ     TS154
4678                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4679                                                              ;           CONDITIONAL BRANCH INST. AND   <====
4680                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
4681                                                              ;           WHICH FOLLOWS W/ 723           <====
4682  013552  012742  000342                MOV     #342,-(R2)      ;MOVE TO MAILBOX #  *******  342  *******
4683  013556  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4684  013560  000000                       HALT                    ;DEST. DATA INCORRECT
4685                                                              ;  OR SEQUENCE ERROR
4686
4687                          ;*************************************************************************************
4688                          ;
4689                          ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
4690                          ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
4691                          ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
4692                          ;USED TO VERIFY PROPER ADDRESSING AND DATA.
4693                          ;
4694                          ;*************************************************************************************
4695                          ;TEST 154       TEST MOV DESTINATION MODE 7
4696                          ;*************************************************************************************
4697  013562  005212          TS154: INC     (R2)            ;UPDATE TEST NUMBER
4698  013564  022712  000154        CMP     #154,(R2)       ;SEQUENCE ERROR?
4699  013570  001053                BNE     TS155-10        ;BR TO ERROR HALT ON SEQ ERROR
4700  013572  005004                CLR     R4              ;R4=0
4701  013574  005014                CLR     (R4)            ;LOC.0=0
4702  013576  012700  000403        MOV     #403,R0         ;R0=403
4703  013602  012770  070707  177777 MOV     #70707,@-1(R0)  ;TRY MOV W/DEST MODE 7
4704  013610  102402                BVS     MDM7A           ;BR. TO ERROR IF V-BIT SET
4705  013612  001401                BEQ     MDM7A           ;BR TO ERROR IF Z-BIT SET
4706  013614  100004                BPL     MDM7B
4707                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4708                                                        ;           CONDITIONAL BRANCH INST. AND   <====
4709                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
4710                                                        ;           WHICH FOLLOWS W/ 765           <====
4711  013616          MDM7A:
4712  013616  012742  000343                MOV     #343,-(R2)      ;MOVE TO MAILBOX #  *******  343  *******
4713  013622  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4714  013624  000000                       HALT                    ;COND. CODES INCORRECT
4715  013626  022700  000403        MDM7B: CMP     #403,R0         ;CHECK DEST. REGISTER
4716  013632  001404                       BEQ     MDM7C
4717                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
4718                                                    ;           CONDITIONAL BRANCH INST. AND   <=-==
4719                                                    ;           REPLACE THE MOVE INSTRUCTION   <= =-
4720                                                    ;           WHICH FOLLOWS W/ 756           <=--=
4721  013634  012742  000344              MOV     #344,-(R2)      ;MOVE TO MAILBOX #  *******  344  *******
4722  013640  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4723  013642  000000                      HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4724  013644  022737  070707  000000  MDM7C:  CMP     #70707,@#0      ;CHECK DEST. DATA
4725  013652  001404                      BEQ     MDM7D
4726                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-- -
4727                                                    ;           CONDITIONAL BRANCH INST. AND   <-- -
4728                                                    ;           REPLACE THE MOVE INSTRUCTION   <- -=
4729                                                    ;           WHICH FOLLOWS W/ 746           <--=-
4730  013654  012742  000345              MOV     #345,-(R2)      ;MOVE TO MAILBOX #  *******  345  *******
4731  013660  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4732  013662  000000                      HALT                    ;DEST. DATA INCORRECT
4733  013664  112770  107070  000001  MDM7D:  MOVB    #107070,@1(R0)  ;TRY MOVB W/DEST MODE 7--ODD BYTE
4734  013672  022700  000403              CMP     #403,R0         ;CHECK MODE 7 DEST. REG.
4735  013676  001404                      BEQ     MDM7E
4736                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
4737                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4738                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4739                                                    ;           WHICH FOLLOWS W/ 734           <====
4740  013700  012742  000346              MOV     #346,-(R2)      ;MOVE TO MAILBOX #  *******  346  *******
4741  013704  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4742  013706  000000                      HALT                    ;DEST. DATA INCORRECT
4743  013710  022737  034307  000000  MDM7E:  CMP     #34307,@#0      ;CHECK DEST. DATA
4744  013716  001404                      BEQ     TS155
4745                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4746                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4747                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4748                                                    ;           WHICH FOLLOWS W/ 724           <====
4749  013720  012742  000347              MOV     #347,-(R2)      ;MOVE TO MAILBOX #  *******  347  *******
4750  013724  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4751  013726  000000                      HALT                    ;DESTINATION DATA INCORRECT
4752                                                    ;  OR SEQUENCE ERROR
4753
4754            ;*****************************************************************************************
4755            ;
4756            ;       THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
4757            ;THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
4758            ;TABLE OF OPERANDS.  THE TABLE OF OPERANDS AND THE WORK LOCATION IS
4759            ;STORED FOLLOWING THE TEST CODE.  A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
4760            ;THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
4761            ;VALUE.  THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
4762            ;GO UNDETECTED.  WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
4763            ;ODD ADDRESSES ARE USED IN THE TEST.  THE LISTING SHOWS THE
4764            ;EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
4765            ;
4766            ;*****************************************************************************************
4767            ;TEST 155       TEST MODE 4 W/ DOP INSTS.
4768            ;*****************************************************************************************
4769  013730  005212            TS155:  INC     (R2)            ;UPDATE TEST NUMBER
4770  013732  022712  000155            CMP     #155,(R2)       ;SEQUENCE ERROR?
4771  013736  001015                    BNE     DOP4            ;BR TO ERROR HALT ON SEQ ERROR
4772  013740  012700  014012            MOV     #TBL1,R0        ;INITIALIZE R0
4773  013744  014037  014012            MOV     -(R0),@#TBL1    ;TBL1=125252
```

```
4774  013750  064037  014012                  ADD     -(R0),@#TBL1    ;TBL1=000377
4775  013754  144037  014012                  BICB    -(R0),@#TBL1    ;TBL1=000252
4776  013760  154037  014013                  BISB    -(R0),@#TBL1+1  ;TBL1-125252
4777  013764  024037  014012                  CMP     -(R0),@#TBL1    ;CHECK RESULT
4778  013770  001411                          BEQ     TS156
4779                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
4780                                                                  ;           CONDITIONAL BRANCH INST. AND   <
4781                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-
4782                                                                  ;           WHICH FOLLOWS W/ 762           <=
4783  013772                          DOP4:
4784  013772  012742  000350                  MOV     #350,-(R2)      ;MOVE TO MAILBOX #  *******  350  *******
4785  013776  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4786  014000  000000                          HALT                    ;RESULT OF MODE 4 INSTS. INCORRECT
4787                                                                  ;  OR SEQUENCE ERROR
4788
4789  014002  125252                          125252
4790  014004  052652                          52652
4791  014006  053125                          53125
4792  014010  125252                          125252
4793  014012  000000                  TBL1:   0
4794
4795                                  ;*****************************************************************************
4796                                  ;
4797                                  ;       THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
4798                                  ;THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
4799                                  ;THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
4800                                  ;THE DATA TABLE USED IN THE PREVIOUS TEST.  THE TEST IS IDENTICAL TO
4801                                  ;THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
4802                                  ;TABLE AND MODE 5 ADDRESSING.  (SEE PREVIOUS TEST).
4803                                  ;
4804                                  ;*****************************************************************************
4805                                  ;TEST 156       TEST MODE 5 W/ DOP INSTS.
4806                                  ;*****************************************************************************
4807  014014  005212                  TS156:  INC     (R2)            ;UPDATE TEST NUMBER
4808  014016  022712  000156                  CMP     #156,(R2)       ;SEQUENCE ERROR?
4809  014022  001015                          BNE     DOP5            ;BR TO ERROR HALT ON SEQ ERROR
4810  014024  012700  014100                  MOV     #TBL2+2,R0      ;INITIALIZE R0
4811  014030  015037  014012                  MOV     @-(R0),@#TBL1   ;TBL1=125252
4812  014034  065037  014012                  ADD     @-(R0),@#TBL1   ;TBL1=000377
4813  014040  145037  014012                  BICB    @-(R0),@#TBL1   ;TBL1-000252
4814  014044  155037  014013                  BISB    @-(R0),@#TBL1+1 ;TBL1=125252
4815  014050  025037  014012                  CMP     @-(R0),@#TBL1   ;CHECK RESULT
4816  014054  001411                          BEQ     TS157
4817                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
4818                                                                  ;           CONDITIONAL BRANCH INST. AND   <
4819                                                                  ;           REPLACE THE MOVE INSTRUCTION   <
4820                                                                  ;           WHICH FOLLOWS W/ 762           <-
4821  014056                          DOP5:
4822  014056  012742  000351                  MOV     #351,-(R2)      ;MOVE TO MAILBOX #  *******  351  *******
4823  014062  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4824  014064  000000                          HALT                    ;RESULT OF MODE 5 INSTS. INCORRECT
4825                                                                  ;  OR SEQUENCE ERROR
4826  014066  014002                          TBL1-10
4827  014070  014004                          TBL1-6
4828  014072  014005                          TBL1-5
4829  014074  014006                          TBL1-4
```

```
 4830  014076  014010                TBL2:   TBL1-2
```

```
4831                                    ;*****************************************************************************
4832                                    ;
4833                                    ;       THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
4834                                    ;IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
4835                                    ;THIS TIME THE DATA IS ACCESSED USING MODE 6.  R0 IS SET
4836                                    ;TO POINT TO THE MIDDLE OF THE TABLE.  THE TABLE IS ACCESSED FROM
4837                                    ;BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
4838                                    ;THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
4839                                    ;TESTS.
4840                                    ;
4841                                    ;*****************************************************************************
4842                                    ;TEST 157       TEST MODE 6 W/ DOP INSTS.
4843                                    ;*****************************************************************************
4844  014100  005212                    TS157:  INC     (R2)            ;UPDATE TEST NUMBER
4845  014102  022712  000157                    CMP     #157,(R2)       ;SEQUENCE ERROR?
4846  014106  001022                            BNE     TS160-10        ;BR TO ERROR HALT ON SEQ ERROR
4847  014110  012700  014006                    MOV     #TBL1-4,R0      ;INITIALIZE R0
4848  014114  016037  000002  014012            MOV     2(R0),@#TBL1    ;TBL1=125252
4849  014122  066037  000000  014012            ADD     0(R0),@#TBL1    ;TBL1=000377
4850  014130  146037  177777  014012            BICB    -1(R0),@#TBL1   ;TBL1=000252
4851  014136  156037  177776  014013            BISB    -2(R0),@#TBL1+1 ;TBL1=125252
4852  014144  026037  177774  014012            CMP     -4(R0),@#TBL1   ;CHECK RESULT
4853  014152  001404                            BEQ     TS160
4854                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
4855                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4856                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4857                                                                    ;           WHICH FOLLOWS W/ 755           <====
4858  014154  012742  000352                    MOV     #352,-(R2)      ;MOVE TO MAILBOX #  *******  352  *******
4859  014160  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4860  014162  000000                            HALT                    ;RESULT OF MODE 6 INSTS. INCORRECT
4861                                                                    ;  OR SEQUENCE ERROR
4862                                    ;*****************************************************************************
4863                                    ;
4864                                    ;       THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
4865                                    ;THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
4866                                    ;THE MODE 5 TESTS.  THIS TIME THE DATA IS ACCESSED USING MODE 7.
4867                                    ;R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
4868                                    ;TEST.  THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
4869                                    ;IN THE MODE 7 INSTRUCTIONS.  THE DATA RESULTS ARE IDENTICAL TO
4870                                    ;THOSE EXPECTED IN THE MODE 5 TESTS.
4871                                    ;
4872                                    ;*****************************************************************************
4873                                    ;TEST 160       TEST MODE 7 W/ DOP INSTS.
4874                                    ;*****************************************************************************
4875  014164  005212                    TS160:  INC     (R2)            ;UPDATE TEST NUMBER
4876  014166  022712  000160                    CMP     #160,(R2)       ;SEQUENCE ERROR?
4877  014172  001022                            BNE     TS161-10        ;BR TO ERROR HALT ON SEQ ERROR
4878  014174  012700  014072                    MOV     #TBL2-4,R0      ;INITIALIZE R0
4879  014200  017037  000004  014012            MOV     @4(R0),@#TBL1   ;TBL1=125252
4880  014206  067037  000002  014012            ADD     @2(R0),@#TBL1   ;TBL1=000377
4881  014214  147037  000000  014012            BICB    @0(R0),@#TBL1   ;TBL1=000252
4882  014222  157037  177776  014013            BISB    @-2(R0),@#TBL1+1                ;TBL1=125252
4883  014230  027037  177774  014012            CMP     @-4(R0),@#TBL1          ;CHECK RESULT
4884  014236  001404                            BEQ     TS161
4885                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= -
4886                                                                    ;           CONDITIONAL BRANCH INST. AND   <=
```

```
                                                        ;           REPLACE THE MOVE INSTRUCTION   <-- -
                                                        ;           WHICH FOLLOWS W/ 755           <==-=
014240  012742  000353                  MOV     #353,-(R2)      ;MOVE TO MAILBOX #  *******  353  *******
014244  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014246  000000                          HALT                    ;RESULT OF MODE 7 INSTS INCORRECT
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       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 161        TEST ROTATE INSTRUCTIONS OF MODE 0
                                ;*****************************************************************************
014250  005212                  TS161:  INC     (R2)            ;UPDATE TEST NUMBER
014252  022712  000161                  CMP     #161,(R2)       ;SEQUENCE ERROR?
014256  001026                          BNE     TS162-10        ;BR TO ERROR HALT ON SEQ ERROR
014260  012700  125252                  MOV     #125252,R0      ;INITIALIZE DATA
014264  000261                          SEC                     ;SET C-BIT
014266  006100                          ROL     R0              ;TRY ROL W/ MODE 0
014270  102004                          BVC     ROT0A           ;CC=0011
014272  103003                          BCC     ROT0A
014274  022700  052525                  CMP     #052525,R0      ;CHECK DATA
014300  001404                          BEQ     ROT0B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=- -
                                                                ;           CONDITIONAL BRANCH INST. AND   <=-
                                                                ;           REPLACE THE MOVE INSTRUCTION   <--
                                                                ;           WHICH FOLLOWS W/ 766           <==--
014302                          ROT0A:
014302  012742  000354                  MOV     #354,-(R2)      ;MOVE TO MAILBOX #  *******  354  *******
014306  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014310  000000                          HALT                    ;ROL MODE 0 FAILED
014312  012700  125252          ROT0B:  MOV     #125252,R0      ;INITIALIZE DATA
014316  000261                          SEC                     ;SET C-BIT
014320  106100                          ROLB    R0              ;TRY ROL W/ MODE 0 EVEN BYTE
014322  102004                          BVC     ROT0C           ;CC=0011
014324  103003                          BCC     ROT0C
014326  022700  125125                  CMP     #125125,R0      ;CHECK DATA
014332  001404                          BEQ     TS162
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 751           <====
014334                          ROT0C:
014334  012742  000355                  MOV     #355,-(R2)      ;MOVE TO MAILBOX #  *******  355  *******
014340  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014342  000000                          HALT                    ;ROLB MODE 0 FAILED
                                                                ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       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 162       TEST ROTATE INSTRUCTIONS W/ MODE 1
                        ;*****************************************************************************
014344  005212          TS162:  INC     (R2)            ;UPDATE TEST NUMBER
014346  022712  000162          CMP     #162,(R2)       ;SEQUENCE ERROR?
014352  001051                  BNE     TS163-10        ;BR TO ERROR HALT ON SEQ ERROR
014354  005000                  CLR     R0              ;POINT TO LOC. 0
014356  012710  052525          MOV     #52525,(R0)     ;INITIALIZE DATA
014362  000241                  CLC                     ;CLEAR C-BIT
014364  006110                  ROL     (R0)            ;TRY ROL W/ MODE 1
014366  102005                  BVC     ROT1A           ;CC=1010
014370  103404                  BCS     ROT1A
014372  023727  000000  125252  CMP     @#0,#125252     ;CHECK RESULT
014400  001404                  BEQ     ROT1B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
014402                  ROT1A:
014402  012742  000356          MOV     #356,-(R2)      ;MOVE TO MAILBOX #  *******  356  *******
014406  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014410  000000                  HALT                    ;ROL MODE 1 FAILED
014412  000261          ROT1B:  SEC
014414  012710  125252          MOV     #125252,(R0)    ;INITIALIZE DATA
014420  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 EVEN BYTE
014422  102005                  BVC     ROT1C           ;CC=1011
014424  103004                  BCC     ROT1C
014426  022737  125125  000000  CMP     #125125,@#0     ;TEST RESULT
014434  001404                  BEQ     ROT1D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 746           <====
014436                  ROT1C:
014436  012742  000357          MOV     #357,-(R2)      ;MOVE TO MAILBOX #  *******  357  *******
014442  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014444  000000                  HALT                    ;ROLB W/ MODE 1 EVEN BYTE FAILED
014446  012710  125252  ROT1D:  MOV     #125252,(R0)
014452  005000                  CLR     R0              ;POINT TO ODD BYTE
014454  005200                  INC     R0
014456  000261                  SEC                     ;SET C-BIT
014460  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 ODD BYTE
014462  102005                  BVC     ROT1E           ;CC=0011
014464  103004                  BCC     ROT1E
014466  022737  052652  000000  CMP     #052652,@#0     ;CHECK DATA
014474  001404                  BEQ     TS163
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 726           <====
014476                                  ROT1E:
014476  012742  000360                          MOV     #360,-(R2)      ;MOVE TO MAILBOX #  *******  360  *******
014502  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014504  000000                                  HALT                    ;ROLB W/ MODE 1 ODD BYTE FAILED
                                                                        ;  OR SEQUENCE ERROR

                                        ;*****************************************************************************
                                        ;
                                        ;       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 163       TEST ROTATE INSTRUCTIONS W/ MODE 2
                                        ;*****************************************************************************
014506  005212                          TS163:  INC     (R2)            ;UPDATE TEST NUMBER
014510  022712  000163                          CMP     #163,(R2)       ;SEQUENCE ERROR?
014514  001057                                  BNE     TS164-10        ;BR TO ERROR HALT ON SEQ ERROR
014516  005000                                  CLR     R0              ;POINT TO LOC 0
014520  012710  173737                          MOV     #173737,(R0)    ;INITIALIZE DATA
014524  000241                                  CLC                     ;CLEAR C-BIT
014526  006120                                  ROL     (R0)+           ;TRY ROL W/ MODE 2
014530  103007                                  BCC     ROT2A           ;CHECK C-BIT
014532  022737  167676  000000                  CMP     #167676,@#0     ;CHECK DATA
014540  001003                                  BNE     ROT2A           ;BRANCH IF RESULT INCORRECT
014542  005300                                  DEC     R0              ;TEST R0
014544  005300                                  DEC     R0
014546  001404                                  BEQ     ROT2B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
014550                                  ROT2A:
014550  012742  000361                          MOV     #361,-(R2)      ;MOVE TO MAILBOX #  *******  361  *******
014554  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014556  000000                                  HALT                    ;ROL W/ MODE 2 FAILED
014560  005000                          ROT2B:  CLR     R0              ;POINT TO LOC 0
014562  012710  004040                          MOV     #4040,(R0)      ;INITIALIZE DATA
014566  000241                                  CLC                     ;CLEAR C-BIT
014570  106120                                  ROLB    (R0)+           ;TRY ROLB W/ MODE 2 EVEN BYTE
014572  103406                                  BCS     ROT2C           ;CHECK C-BIT
014574  022737  004100  000000                  CMP     #4100,@#0       ;CHECK DATA
014602  001002                                  BNE     ROT2C           ;BRANCH IF DATA INCORRECT
014604  005300                                  DEC     R0              ;CHECK R0
014606  001404                                  BEQ     ROT2D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 742           <====
014610                                  ROT2C:
014610  012742  000362                          MOV     #362,-(R2)      ;MOVE TO MAILBOX #  *******  362  *******
```

```
5050  014614  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5051  014516  000000                          HALT                    ;ROLB W/ MODE 2 EVEN BYTE FAILED
5052  014620  005000                  ROT2D:  CLR     R0              ;POINT TO LOC 0
5053  014622  012710  004040                  MOV     #4040,(R0)      ;INITIALIZE DATA
5054  014626  005200                          INC     R0              ;POINT TO ODD BYTE OF DATA
5055  014630  000261                          SEC                     ;SET C-BIT
5056  014632  106120                          ROLB    (R0)+           ;TRY ROL W/ MODE 2 ODD BYTE
5057  014634  103407                          BCS     ROT2E           ;CHECK C-BIT
5058  014636  022737  010440  000000          CMP     #10440,@#0      ;CHECK DATA
5059  014644  001003                          BNE     ROT2E           ;BRANCH IF DATA INCORRECT
5060  014646  005300                          DEC     R0              ;CHECK R0
5061  014650  005300                          DEC     R0
5062  014652  001404                          BEQ     TS164
5063                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-=-=
5064                                                                  ;           CONDITIONAL BRANCH INST. AND   <-=-
5065                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-=-=
5066                                                                  ;           WHICH FOLLOWS W/ 720           <-=
5067  014654                          ROT2E:
5068  014654  012742  000363                  MOV     #363,-(R2)      ;MOVE TO MAILBOX #  *******  363  *******
5069  014660  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5070  014662  000000                          HALT                    ;ROLB W/ MODE 2 ODD BYTE FAILED
5071                                                                  ;  OR SEQUENCE ERROR
```

```
5072
5073                            ;*****************************************************************************
5074                            ;
5075                            ;       THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
5076                            ;THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
5077                            ;TESTS.  THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
5078                            ;MODE 37.  BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
5079                            ;
5080                            ;*****************************************************************************
5081                            ;TEST 164       TEST ROTATE INSTRUCTIONS /W MODE 3
5082                            ;*****************************************************************************
5083  014664  005212            TS164:  INC     (R2)            ;UPDATE TEST NUMBER
5084  014666  022712  000164            CMP     #164,(R2)       ;SEQUENCE ERROR?
5085  014672  001051                    BNE     TS165-10        ;BR TO ERROR HALT ON SEQ ERROR
5086  014674  012737  052525  000000    MOV     #52525,@#0      ;INITIALIZE DATA IN LOC 0
5087  014702  000261                    SEC                     ;SET C-BIT
5088  014704  006137  000000            ROL     @#0             ;TRY ROL W/ MODE 3
5089  014710  103404                    BCS     ROT3A           ;CHECK C-BIT
5090  014712  022737  125253  000000    CMP     #125253,@#0     ;CHECK DATA
5091  014720  001404                    BEQ     ROT3B
5092                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5093                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5094                                                            ;           REPLACE THE MOVE INSTRUCTION   <-===
5095                                                            ;           WHICH FOLLOWS W/ 764           <-=-=
5096  014722                    ROT3A:
5097  014722  012742  000364            MOV     #364,-(R2)      ;MOVE TO MAILBOX #  *******  364  *******
5098  014726  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5099  014730  000000                    HALT                    ;ROL W/ MODE 3 FAILED
5100  014732  012737  125252  000000  ROT3B:  MOV     #125252,@#0     ;INITIALIZE DATA
5101  014740  000241                    CLC                     ;CLEAR C-BIT
5102  014742  106137  000000            ROLB    @#0             ;TRY ROL W/ MODE 3 EVEN BYTE
5103  014746  103004                    BCC     ROT3C           ;CHECK C-BIT
5104  014750  023727  000000  125124  4$:     CMP     @#0,#125124     ;CHECK DATA
5105  014756  001404                    BEQ     ROT3D
5106                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=  -
5107                                                            ;           CONDITIONAL BRANCH INST. AND   < -==
5108                                                            ;           REPLACE THE MOVE INSTRUCTION   <==-=
5109                                                            ;           WHICH FOLLOWS W/ 745           < - -
5110  014760                    ROT3C:
5111  014760  012742  000365            MOV     #365,-(R2)      ;MOVE TO MAILBOX #  *******  365  *******
5112  014764  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5113  014766  000000                    HALT                    ;ROL W/ MODE 3 EVEN BYTE FAILED
5114  014770  012737  125252  000000  ROT3D:  MOV     #125252,@#0     ;INITIALIZE DATA IN LOC. 0
5115  014776  000261                    SEC                     ;SET C-BIT
5116  015000  106137  000001            ROLB    @#1             ;TRY ROL W/ MODE 3 ODD BYTE
5117  015004  103004                    BCC     ROT3E           ;CHECK C-BIT
5118  015006  022737  052652  000000    CMP     #052652,@#0     ;CHECK DATA
5119  015014  001404                    BEQ     TS165
5120                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5121                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5122                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5123                                                            ;           WHICH FOLLOWS W/ 726           <====
5124  015016                    ROT3E:
5125  015016  012742  000366            MOV     #366,-(R2)      ;MOVE TO MAILBOX #  *******  366  *******
5126  015022  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5127  015024  000000                    HALT                    ;ROL W/ MODE 3 ODD BYTE FAILED
```

```
                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;
                                ;       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 165       TEST MODE 4 W/ ROTATE INSTRUCTIONS
                                ;*******************************************************************************
015026  005212                  TS165:  INC     (R2)            ;UPDATE TEST NUMBER
015030  022712  000165                  CMP     #165,(R2)       ;SEQUENCE ERROR?
015034  001016                          BNE     TS166-10        ;BR TO ERROR HALT ON SEQ ERROR
015036  012737  070707  000000          MOV     #070707,@#0     ;INITIALIZE DATA IN LOC. 0
015044  012700  000002                  MOV     #2,R0           ;INITIALIZE R0 AS POINTER
015050  000261                          SEC                     ;SET C-BIT
015052  006140                          ROL     -(R0)           ;TRY ROL W/ MODE 4
015054  103406                          BCS     ROT4            ;CHECK C-BIT
015056  022737  161617  000000          CMP     #161617,@#0     ;CHECK DATA
015064  001002                          BNE     ROT4            ;BRANCH IF DATA INCORRECT
015066  005700                          TST     R0              ;CHECK MODE 4 REGISTER
015070  001404                          BEQ     TS166
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 761           <====
015072                          ROT4:
015072  012742  000367                  MOV     #367,-(R2)      ;MOVE TO MAILBOX #  *******  367  *******
015076  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015100  000000                          HALT                    ;ROL MODE 4 FAILED
                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;
                                ;       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 166       TEST MODE 5 W/ ROTATE INSTRUCTIONS
                                ;*******************************************************************************
015102  005212                  TS166:  INC     (R2)            ;UPDATE TEST NUMBER
015104  022712  000166                  CMP     #166,(R2)       ;SEQUENCE ERROR?
015110  001021                          BNE     ROT5            ;BR TO ERROR HALT ON SEQ ERROR
015112  012737  015164  000000          MOV     #ROTX,@#0       ;MOVE POINTER TO LOC. 0
015120  012700  000002                  MOV     #2,R0           ;SET MODE 5 REG. TO LOC. 0
015124  012767  107070  000032          MOV     #107070,ROTX    ;INITIALIZE DATA
015132  000241                          CLC                     ;CLEAR C-BIT
015134  006150                          ROL     @-(R0)          ;TRY ROL W/ MODE 5
```

```
5184  015136  103006                          BCC     ROT5            ;CHECK C-BIT
5185  015140  022737  016160  015164          CMP     #016160,@#ROTX  ;CHECK DATA
5186  015146  001002                          BNE     ROT5            ;BRANCH IF DATA INCORRECT
5187  015150  005700                          TST     R0              ;CHECK MODE 5 REGISTER
5188  015152  001405                          BEQ     TS167
5189                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5190                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5191                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5192                                                                  ;           WHICH FOLLOWS W/ 756           <====
5193  015154                          ROT5:
5194  015154  012742  000370                  MOV     #370,-(R2)      ;MOVE TO MAILBOX #  *******  370  *******
5195  015160  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5196  015162  000000                          HALT                    ;ROL MODE 5 FAILED
5197                                                                  ;  OR SEQUENCE ERROR
5198  015164  000000                  ROTX:   0
5199
5200                                  ;*****************************************************************************
5201                                  ;
5202                                  ;       THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.
5203                                  ;IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE
5204                                  ;ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.
5205                                  ;THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).
5206                                  ;
5207                                  ;*****************************************************************************
5208                                  ;TEST 167       TEST MODE 6 W/ ROTATE INSTRUCTIONS
5209                                  ;*****************************************************************************
5210  015166  005212                  TS167:  INC     (R2)            ;UPDATE TEST NUMBER
5211  015170  022712  000167                  CMP     #167,(R2)       ;SEQUENCE ERROR?
5212  015174  001013                          BNE     TS170-10        ;BR TO ERROR HALT ON SEQ ERROR
5213  015176  012737  125252  015164          MOV     #125252,@#ROTX  ;INITIALIZE DATA
5214  015204  000261                          SEC                     ;SET C-BIT
5215  015206  006167  177752                  ROL     ROTX            ;TRY ROL W/ MODE 6
5216  015212  103004                          BCC     ROT6            ;CHECK C-BIT
5217  015214  022737  052525  015164          CMP     #52525,@#ROTX   ;CHECK DATA
5218  015222  001404                          BEQ     TS170
5219                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5220                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5221                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5222                                                                  ;           WHICH FOLLOWS W/ 764           <====
5223  015224                          ROT6:
5224  015224  012742  000371                  MOV     #371,-(R2)      ;MOVE TO MAILBOX #  *******  371  *******
5225  015230  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5226  015232  000000                          HALT                    ;ROL W/ MODE 6 FAILED
5227                                                                  ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;
                                ;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 170        TEST MODE 7 W/ ROTATE INSTRUCTIONS
                                ;*****************************************************************************************
015234  005212                  TS170:  INC     (R2)            ;UPDATE TEST NUMBER
015236  022712  000170                  CMP     #170,(R2)       ;SEQUENCE ERROR?
015242  001016                          BNE     ROT7            ;BR TO ERROR HALT ON SEQ ERROR
015244  012737  052525  015164          MOV     #52525,@#ROTX   ;INITIALIZE DATA
015252  012737  015164  015310          MOV     #ROTX,@#ROTXAD  ;INITIALIZE ADDRESS POINTER
015260  000241                          CLC                     ;CLEAR C-BIT
015262  006177  000022                  ROL     @ROTXAD         ;TRY ROL W/ MODE 7
015266  103404                          BCS     ROT7            ;CHECK C-BIT
015270  023727  015164  125252          CMP     @#ROTX,#125252  ;CHECK DATA
015276  001405                          BEQ     TS171
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 761           <====
015300                          ROT7:
015300  012742  000372                  MOV     #372,-(R2)      ;MOVE TO MAILBOX #  *******  372  *******
015304  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015306  000000                          HALT                    ;ROL W/ MODE 7 FAILED
                                                                ;  OR SEQUENCE ERROR
015310  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 171        TEST MODE 0 W/ SWAB INST.
                                ;*****************************************************************************************
015312  005212                  TS171:  INC     (R2)            ;UPDATE TEST NUMBER
015314  022712  000171                  CMP     #171,(R2)       ;SEQUENCE ERROR?
015320  001013                          BNE     TS172-10        ;BR TO ERROR HALT ON SEQ ERROR
015322  012700  177400                  MOV     #177400,R0      ;MOVE TEST PATTERN TO R0
015326  000300                          SWAB    R0              ;TRY SWAB MODE 0
015330  100404                          BMI     SB0
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 773           <====
015332  012742  000373                  MOV     #373,-(R2)      ;MOVE TO MAILBOX #  *******  373  *******
015336  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015340  000000                          HALT                    ;SWAB DID NOT SET CC'S CORRECT
```

```
5284  015342  022700  000377          SB0:    CMP     #377,R0         ;CHECK RESULT
5285  015346  001404                          BEQ     TS172
5286                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5287                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5288                                                                    ;           REPLACE THE MOVE INSTRUCTION   <-==
5289                                                                    ;           WHICH FOLLOWS W/ 764           <=-==
5290  015350  012742  000374                      MOV     #374,-(R2)      ;MOVE TO MAILBOX #  *******  374  *******
5291  015354  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5292  015356  000000                          HALT                    ;RESULT OF SWAB MODE 0 FAILED
5293                                                                    ;  OR SEQUENCE ERROR
5294
5295                                  ;*****************************************************************************
5296                                  ;
5297                                  ;       THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION.  THE TEST
5298                                  ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE ADDRESSING
5299                                  ;REGISTER IN THE MODE 1 SWAB.  THE DATA RESULTS ARE CHECKED WITH
5300                                  ;A COMPARE.
5301                                  ;
5302                                  ;*****************************************************************************
5303                                  ;TEST 172       TEST MODE 1 W/ SWAB INST
5304                                  ;*****************************************************************************
5305  015360  005212                  TS172:  INC     (R2)            ;UPDATE TEST NUMBER
5306  015362  022712  000172                  CMP     #172,(R2)       ;SEQUENCE ERROR?
5307  015366  001011                          BNE     TS173-10        ;BR TO ERROR HALT ON SEQ ERROR
5308  015370  012737  125652  000000          MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
5309  015376  005000                          CLR     R0              ;R0=0
5310  015400  000310                          SWAB    (R0)            ;TRY SWAB MODE 1
5311  015402  022737  125253  000000          CMP     #125253,@#0     ;CHECK RESULT
5312  015410  001404                          BEQ     TS173
5313                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5314                                                                  ;           CONDITIONAL BRANCH INST. AND   <=-=-
5315                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5316                                                                  ;           WHICH FOLLOWS W/ 766           <===-
5317  015412  012742  000375                  MOV     #375,-(R2)      ;MOVE TO MAILBOX #  *******  375  *******
5318  015416  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5319  015420  000000                          HALT                    ;RESULT OF SWAB MODE 1 FAILED
5320                                                                  ;  OR SEQUENCE ERROR
```

```
5321
5322
5323                            ;*****************************************************************************************
5324                            ;
5325                            ;       THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION.  THE TEST
5326                            ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE MODE
5327                            ;2 ADDRESSING REGISTER.  THE RESULTS ARE CHECKED WITH A COMPARE.
5328                            ;R0 IS CHECKED FOR PROPER DECREMENTING.
5329                            ;
5330                            ;*****************************************************************************************
5331                            ;TEST 173       TEST MODE 2 W/ SWAB INST
5332                            ;*****************************************************************************************
5333  015422  005212            TS173:  INC     (R2)            ;UPDATE TEST NUMBER
5334  015424  022712  000173            CMP     #173,(R2)       ;SEQUENCE ERROR?
5335  015430  001020                    BNE     TS174-10        ;BR TO ERROR HALT ON SEQ ERROR
5336  015432  012737  125152  000000    MOV     #125152,@#0     ;MOVE TEST PATTERN TO LOC. 0
5337  015440  005000                    CLR     R0              ;R0=0
5338  015442  000320                    SWAB    (R0)+           ;TRY SWAB MODE 2
5339  015444  022737  065252  000000    CMP     #65252,@#0      ;CHECK RESULT
5340  015452  001404                    BEQ     SB2
5341                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5342                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5343                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5344                                                            ;           WHICH FOLLOWS W/ 766           <====
5345  015454  012742  000376            MOV     #376,-(R2)      ;MOVE TO MAILBOX #  *******  376  *******
5346  015460  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5347  015462  000000                    HALT                    ;RESULT OF SWAB MODE 0 FAILED
5348  015464  162700  000002    SB2:    SUB     #2,R0           ;CHECK EFFECT OF REG.
5349  015470  001404                    BEQ     TS174
5350                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5351                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5352                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5353                                                            ;           WHICH FOLLOWS W/ 757           <====
5354  015472  012742  000377            MOV     #377,-(R2)      ;MOVE TO MAILBOX #  *******  377  *******
5355  015476  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5356  015500  000000                    HALT                    ;REGISTER VALUE INCORRECT
5357                                                            ;  OR SEQUENCE ERROR
5358
5359
5360                            ;*****************************************************************************************
5361                            ;
5362                            ;       THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION.  THE TEST
5363                            ;PATTERN IS MOVED TO LOC 0.  A MODE 3 SWAB INSTRUCTION IS EXECUTED
5364                            ;USING R7 AS THE ADDRESSING REGISTER.  A COMPARE VERIFIES THE
5365                            ;DATA RESULTS.
5366                            ;
5367                            ;*****************************************************************************************
5368                            ;TEST 174       TEST MODE 3 W/SWAB INST.
5369                            ;*****************************************************************************************
5370  015502  005212            TS174:  INC     (R2)            ;UPDATE TEST NUMBER
5371  015504  022712  000174            CMP     #174,(R2)       ;SEQUENCE ERROR?
5372  015510  001011                    BNE     TS175-10        ;BR TO ERROR HALT ON SEQ ERROR
5373  015512  012737  000377  000000    MOV     #377,@#0        ;MOVE TEST PATTERN TO LOC. 0
5374  015520  000337  000000            SWAB    @#0             ;TRY SWAB W/ MODE 3
5375  015524  022737  177400  000000    CMP     #177400,@#0     ;CHECK RESULT
5376  015532  001404                    BEQ     TS175
```

```
5377                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
5378                                                    ;           CONDITIONAL BRANCH INST. AND   <----=
5379                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5380                                                    ;           WHICH FOLLOWS W/ 766           <----
5381  015534  012742  000400                MOV     #400,-(R2)  ;MOVE TO MAILBOX #  *******  400  *******
5382  015540  005242                       INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
5383  015542  000000                       HALT                ;RESULT OF SWAB INCORRECT
5384                                                            ;  OR SEQUENCE ERROR
```

```
5385
5386
5387                                    ;*******************************************************************************
5388                                    ;
5389                                    ;       THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
5390                                    ;IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
5391                                    ;REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
5392                                    ;FOR PROPER DECREMENTING.
5393                                    ;
5394                                    ;*******************************************************************************
5395                                    ;TEST 175       TEST MODE 4 W/ SWAB INST
5396                                    ;*******************************************************************************
5397  015544  005212                    TS175:  INC     (R2)            ;UPDATE TEST NUMBER
5398  015546  022712  000175                    CMP     #175,(R2)       ;SEQUENCE ERROR?
5399  015552  001020                            BNE     TS176-10        ;BR TO ERROR HALT ON SEQ ERROR
5400  015554  012737  125652  000000            MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
5401  015562  012700  000002                    MOV     #2,R0           ;SET UP REGISTER POINTER
5402  015566  000340                            SWAB    -(R0)           ;TRY SWAB MODE 4
5403  015570  022737  125253  000000            CMP     #125253,@#0     ;CHECK RESULT
5404  015576  001404                            BEQ     SB4
5405                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5406                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5407                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5408                                                                    ;           WHICH FOLLOWS W/ 765           <====
5409  015600  012742  000401                    MOV     #401,-(R2)      ;MOVE TO MAILBOX #  *******  401  *******
5410  015604  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5411  015606  000000                            HALT                    ;RESULT OF SWAB INCORRECT
5412  015610  005700                    SB4:    TST     R0              ;CHECK EFFECT ON REG.
5413  015612  001404                            BEQ     TS176
5414
5415                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5416                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5417                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5418  015614  012742  000402                    MOV     #402,-(R2)      ;           WHICH FOLLOWS W/ 757           <====
5419  015620  005242                            INC     -(R2)           ;MOVE TO MAILBOX #  *******  402  *******
5420  015622  000000                            HALT                    ;SET MSGTYP TO FATAL ERROR
5421                                                                    ;REGISTER VALUE INCORRECT
5422                                                                    ;  OR SEQUENCE ERROR
```

```
5423
5424                            ;*****************************************************************************
5425                            ;
5426                            ;       THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION.  THE TEST USES
5427                            ;TWO LOCATIONS FOLLOWING THE TEST CODE.  SB5X HOLDS THE DATA;
5428                            ;SB5XAD IS A POINTER TO THE DATA LOCATION.  THE DATA IS MOVED TO
5429                            ;SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD.  FOLLOWING
5430                            ;THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA.  R0 IS
5431                            ;CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
5432                            ;
5433                            ;*****************************************************************************
5434                            ;TEST 176       TEST MODE 5 W/ SWAB INST.
5435                            ;*****************************************************************************
5436  015624  005212            TS176:  INC     (R2)            ;UPDATE TEST NUMBER
5437  015626  022712  000176            CMP     #176,(R2)       ;SEQUENCE ERROR?
5438  015632  001021                    BNE     SB5             ;BR TO ERROR HALT ON SEQ ERROR
5439  015634  012700  015712            MOV     #SB5XAD+2,R0    ;SET UP POINTER TO WORK LOCATION
5440  015640  012767  125125  000040    MOV     #125125,SB5X    ;MOVE PATTERN TO WORK LOCATION
5441  015646  000350                    SWAB    @-(R0)          ;TRY SWAB MODE 5
5442  015650  022767  052652  000030    CMP     #52652,SB5X     ;CHECK RESULT
5443  015656  001404                    BEQ     SB5A
5444                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5445                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5446                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5447                                                    ;           WHICH FOLLOWS W/ 765           <====
5448  015660  012742  000403            MOV     #403,-(R2)      ;MOVE TO MAILBOX #  *******  403  *******
5449  015664  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5450  015666  000000                    HALT                    ;RESULT OF SWAB INCORRECT
5451  015670  020027  015710    SB5A:   CMP     R0,#SB5XAD      ;CHECK RESULT OF REG.
5452  015674  001406                    BEQ     TS177
5453                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5454                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5455                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5456                                                    ;           WHICH FOLLOWS W/ 756           <====
5457  015676            SB5:
5458  015676  012742  000404            MOV     #404,-(R2)      ;MOVE TO MAILBOX #  *******  404  *******
5459  015702  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5460  015704  000000                    HALT                    ;REGISTER VALUE INCORRECT
5461                                                            ;  OR SEQUENCE ERROR
5462  015706  000000    SB5X:   0       ;WORK LOCATION
5463  015710  015706    SB5XAD: SB5X
5464
```

```
5465
5466                            ;*****************************************************************************
5467                            ;
5468                            ;       THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION.  THIS TEST
5469                            ;USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE.  TEST DATA
5470                            ;IS LOADED INTO THE WORK LOCATION.  R0, THE ADDRESSING REGISTER
5471                            ;IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
5472                            ;THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET.  THE DATA IS
5473                            ;VERIFIED WITH A COMPARE.
5474                            ;
5475                            ;*****************************************************************************
5476                            ;TEST 177       TEST MODE 6 W/ SWAB INST.
5477                            ;*****************************************************************************
5478  015712  005212            TS177:  INC     (R2)            ;UPDATE TEST NUMBER
5479  015714  022712  000177            CMP     #177,(R2)       ;SEQUENCE ERROR?
5480  015720  001013                    BNE     SB6             ;BR TO ERROR HALT ON SEQ ERROR
5481  015722  012767  125125  000030    MOV     #125125,SB6X    ;MOVE PATTERN TO WORK LOCATION
5482  015730  012700  015752            MOV     #SB6X-6,R0      ;MOVE OFFSET POINTER TO R0
5483  015734  000360  000006            SWAB    6(R0)           ;TRY SWAB W/ MODE 6
5484  015740  022760  052652  000006    CMP     #52652,6(R0)    ;CHECK RESULT
5485  015746  001405                    BEQ     TS200
5486                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5487                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5488                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5489                                                    ;           WHICH FOLLOWS W/ 764           <====
5490  015750                    SB6:
5491  015750  012742  000405            MOV     #405,-(R2)      ;MOVE TO MAILBOX #  *******  405  *******
5492  015754  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5493  015756  000000                    HALT                    ;RESULT OF SWAB INCORRECT
5494                                                            ;  OR SEQUENCE ERROR
5495  015760  000000            SB6X:   0                       ;WORK LOCATION
5496
```

```
5497
5498                                    ;*****************************************************************************
5499                                    ;
5500                                    ;       THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION.  THIS TEST
5501                                    ;USES TWO LOCATIONS FOLLOWING THE TEST CODE:  A WORK LOCATION
5502                                    ;(SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD).  DATA IS MOVED
5503                                    ;TO THE WORK LOCATION.  R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
5504                                    ;OF THE ADDRESS POINTER.  THE DATA IS SWAB'ED USING A MODE 7
5505                                    ;INSTRUCTION WITH AN OFFSET OF +72.  THE DATA IS VERIFIED WITH A
5506                                    ;COMPARE.
5507                                    ;
5508                                    ;*****************************************************************************
5509                                    ;TEST 200       TEST MODE 7 W/ SWAB INST.
5510                                    ;*****************************************************************************
5511  015762  005212                    TS200:  INC     (R2)            ;UPDATE TEST NUMBER
5512  015764  022712  000200                   CMP     #200,(R2)       ;SEQUENCE ERROR?
5513  015770  001013                           BNE     SB7             ;BR TO ERROR HALT ON SEQ ERROR
5514  015772  012767  177400  000030          MOV     #177400,SB7X    ;MOVE PATTERN TO WORK LOCATION
5515  016000  012700  015740                   MOV     #SB7XAD-72,R0   ;MOVE OFFSET POINTER TO R0
5516  016004  000370  000072                   SWAB    @72(R0)         ;TRY SWAB MODE 7
5517  016010  027027  000072  000377          CMP     @72(R0),#377    ;CHECK RESULTS
5518  016016  001406                           BEQ     TS201
5519                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-_-=
5520                                                                   ;           CONDITIONAL BRANCH INST. AND   < ==
5521                                                                   ;           REPLACE THE MOVE INSTRUCTION   <=--
5522                                                                   ;           WHICH FOLLOWS W/ 764           <=-
5523  016020                            SB7:
5524  016020  012742  000406                   MOV     #406,-(R2)      ;MOVE TO MAILBOX #  *******  406  *******
5525  016024  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5526  016026  000000                           HALT                    ;RESULT OF SWAB INCORRECT
5527                                                                   ;  OR SEQUENCE ERROR
5528  016030  000000                    SB7X:   0                       ;WORK LOCATION
5529  016032  016030                    SB7XAD: SB7X                    ;POINTER TO WORK LOCATION
5530
```

```
5531
5532                            ;*****************************************************************************************
5533                            ;
5534                            ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
5535                            ;BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
5536                            ;UTILIZES SEVERAL DIFFERENT TECHNIQUES.  THE CODE IS NOT EXECUTED
5537                            ;IN A LINEAR FASHION.  THE DIFFERENT MODES ARE EXECUTED IN ORDER
5538                            ;FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
5539                            ;FROGS THRU THE TEST CODE.  THE ORDER OF APPEARANCE OF THE CODE
5540                            ;IS:
5541                            ;       JMP MODE 1
5542                            ;       JMP MODE 3
5543                            ;       JMP MODE 2
5544                            ;       JMP MODE 4
5545                            ;       JMP MODE 6
5546                            ;       JMP MODE 5
5547                            ;       JMP MODE 7
5548                            ;AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
5549                            ;JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
5550                            ;       THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE.  EACH CODE
5551                            ;BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
5552                            ;THAT BLOCK.  A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK.  FOR
5553                            ;EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
5554                            ;OF THE PREVIOUS MODE 2 JUMP.  (ANY REGISTER CHANGES ARE VERIFIED
5555                            ;AND THE SEQUENCE CHECK IS MADE).  THEN THE REGISTERS ARE SETUP
5556                            ;FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
5557                            ;CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
5558                            ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
5559                            ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
5560                            ;THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
5561                            ;REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
5562                            ;
5563                            ;*****************************************************************************************
5564                            ;TEST 201       TEST THE JMP INSTRUCTION IN ALL MODES
5565                            ;*****************************************************************************************
5566  016034  005212            TS201:  INC     (R2)            ;UPDATE TEST NUMBER
5567  016036  022712  000201            CMP     #201,(R2)       ;SEQUENCE ERROR?
5568  016042  001150                    BNE     JMPCK+6         ;BR TO ERROR HALT ON SEQ ERROR
5569  016044  005067  000326            CLR     JMPSEQ          ;ESTABLISH A SEQUENCE CHECKER
5570  016050  012700  016130            MOV     #JMP2,R0        ;SET R0=JUMP TARGET
5571  016054  000110                    JMP     (R0)            ;TRY JMP MODE 1
5572  016056  022700  016060    JMP3:   CMP     #.+2,R0         ;CHECK RESULT OF MODE 2 JUMP
5573  016062  001404                    BEQ     JMP3A
5574                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=
5575                                                            ;           CONDITIONAL BRANCH INST. AND   <=-==
5576                                                            ;           REPLACE THE MOVE INSTRUCTION   <=-=-
5577                                                            ;           WHICH FOLLOWS W/ 767           <=-=-
5578  016064  012742  000407                    MOV     #407,-(R2)      ;MOVE TO MAILBOX #  *******  407  *******
5579  016070  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5580  016072  000000                    HALT                    ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
5581  016074  026727  000276  000001    JMP3A:  CMP     JMPSEQ,#1       ;MAKE SURE JMPS ARE IN SEQUENCE: JMPSEQ=1?
5582  016102  001404                    BEQ     JMP3B
5583                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5584                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5585                                                            ;           REPLACE THE MOVE INSTRUCTION   <=-=-
5586                                                            ;           WHICH FOLLOWS W/ 757           <=-=-
```

```
5587  016104  012742  000410          MOV     #410,-(R2)      ;MOVE TO MAILBOX #  *******  410  *******
5588  016110  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5589  016112  000000                  HALT                    ;SHOULD BE HERE FROM JMP MODE 2 ONLY
5590  016114  012700  016126  JMP3B:  MOV     #IJMP4,R0       ;POINT R0 TO INDIRECT JMP ADDR.
5591  016120  005267  000252          INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5592  016124  000130                  JMP     @(R0)+          ;TRY JMP MODE 3
5593  016126  016160  IJMP4:  JMP4                            ;ADDRESS INDIRECT JUMP
5594
5595  016130  005767  000242  JMP2:   TST     JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
5596  016134  001404                  BEQ     JMP2A
5597                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5598                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5599                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5600                                                          ;           WHICH FOLLOWS W/ 742           <====
5601  016136  012742  000411          MOV     #411,-(R2)      ;MOVE TO MAILBOX #  *******  411  *******
5602  016142  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5603  016144  000000                  HALT                    ;SHOULD BE HERE FROM JMP MODE 1 ONLY
5604  016146  005267  000224  JMP2A:  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5605  016152  012700  016056          MOV     #JMP3,R0        ;SET R0=JUMP TARGET
5606  016156  000120                  JMP     (R0)+           ;TRY A JUMP MODE 2 TO ''JMP3''
5607  016160  022700  016130  JMP4:   CMP     #IJMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
5608  016164  001404                  BEQ     JMP4A
5609                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===
5610                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5611                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5612                                                          ;           WHICH FOLLOWS W/ 726           <====
5613  016166  012742  000412          MOV     #412,-(R2)      ;MOVE TO MAILBOX #  *******  412  *******
5614  016172  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5615  016174  000000                  HALT                    ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5616  016176  022767  000002  000172  JMP4A:  CMP     #2,JMPSEQ       ;CHECK JUMP SEQUENCE: JMPSEQ=2?
5617  016204  001404                  BEQ     JMP4B
5618                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5619                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5620                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5621                                                          ;           WHICH FOLLOWS W/ 716           <====
5622  016206  012742  000413          MOV     #413,-(R2)      ;MOVE TO MAILBOX #  *******  413  *******
5623  016212  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5624  016214  000000                  HALT                    ;SHOULD BE ONLY FROM MODE 3 JUMP
5625  016216  012700  016266  JMP4B:  MOV     #JMP5+2,R0      ;SET UP POINTER TO JUMP TARGET
5626  016222  005267  000150          INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5627  016226  000140                  JMP     -(R0)           ;TRY JUMP MODE 4 TO ''JMP4''
5628
5629  016230  022767  000004  000140  JMP6:   CMP     #4,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5630  016236  001404                  BEQ     JMP6A
5631                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5632                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5633                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5634                                                          ;           WHICH FOLLOWS W/ 701           <====
5635  016240  012742  000414          MOV     #414,-(R2)      ;MOVE TO MAILBOX #  *******  414  *******
5636  016244  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5637  016246  000000                  HALT                    ;SHOULD BE HERE ONLY FROM MODE 3 JUMP
5638  016250  012700  016716  JMP6A:  MOV     #IMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
5639  016254  005267  000116          INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5640  016260  000160  177402          JMP     -376(R0)        ;TRY MODE 6 JUMP
5641
5642  016264  022767  000003  000104  JMP5:   CMP     #3,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
```

```
5643  016272  001404                        BEQ     JMP5A
5644                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5645                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5646                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5647                                                  ;           WHICH FOLLOWS W/ 663           <====
5648  016274  012742  000415                  MOV     #415,-(R2)      ;MOVE TO MAILBOX #  *******  415  *******
5649  016300  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5650  016302  000000                        HALT                    ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5651  016304  012700  016320          JMP5A:  MOV     #IJMP5+2,R0     ;SET UP POINTER TO INDIRECT JUMP ADDR.
5652  016310  005267  000062                INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5653  016314  000150                        JMP     @-(R0)          ;TRY JUMP MODE 5 TO "JMP6"
5654  016316  016230          IJMP5:  JMP6                    ;INDIRECT ADDRESS POINTER
5655
5656  016320  022767  000005  000050  JMP7:  CMP     #5,JMPSEQ       ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5657  016326  001404                        BEQ     JMP7A
5658                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5659                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5660                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5661                                                  ;           WHICH FOLLOWS W/ 645           <====
5662  016330  012742  000416                  MOV     #416,-(R2)      ;MOVE TO MAILBOX #  *******  416  *******
5663  016334  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5664  016336  000000                        HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5665  016340  012700  016364          JMP7A:  MOV     #IJMP+10,R0     ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5666  016344  005267  000026                INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5667  016350  000170  177770                JMP     @-10(R0)        ;TRY MODE 7 JUMP
5668  016354  016356          IJMP:   JMPCK                   ;INDIRECT ADDRESS
5669
5670  016356  026727  000014  000006  JMPCK:  CMP     JMPSEQ,#6       ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5671  016364  001405                        BEQ     TS202
5672                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5673                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5674                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5675                                                  ;           WHICH FOLLOWS W/ 626           <====
5676  016366  012742  000417                  MOV     #417,-(R2)      ;MOVE TO MAILBOX #  *******  417  *******
5677  016372  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5678  016374  000000                        HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5679                                                  ;  OR SEQUENCE ERROR
5680  016376  000000          JMPSEQ: 0
```

```
5681
5682                            ;*****************************************************************************************
5683                            ;
5684                            ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
5685                            ;THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
5686                            ;IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST).  EACH
5687                            ;BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
5688                            ;CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
5689                            ;THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
5690                            ;SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
5691                            ;REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
5692                            ;SUCCESSFUL.  R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
5693                            ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
5694                            ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
5695                            ;THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
5696                            ;REGISTER SAVED).
5697                            ;
5698                            ;*****************************************************************************************
5699                            ;TEST 202       TEST JSR INSTRUCTION W/ ALL MODES
5700                            ;*****************************************************************************************
5701  016400  005212            TS202:  INC     (R2)            ;UPDATE TEST NUMBER
5702  016402  022712  000202            CMP     #202,(R2)       ;SEQUENCE ERROR?
5703  016406  001001                    BNE     JSR0            ;BR TO ERROR HALT ON SEQ ERROR
5704  016410  000402                    BR      JSR1
5705  016412  000137  017046    JSR0:   JMP     @#JSRCK1
5706
5707  016416  012706  001000    JSR1:   MOV     #STBOT,R6       ;SET STACK POINTER
5708  016422  012700  016530            MOV     #JSR2,R0        ;SET TARGET ADDRESS
5709  016426  005037  017026            CLR     @#JSRSEQ        ;INITIALIZE SEQUENCE CHECKER
5710  016432  005001                    CLR     R1              ;INITIALIZE R1
5711  016434  005101                    COM     R1
5712  016436  004110                    JSR     R1,(R0)         ;TRY JSR MODE 1
5713                                                            ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
5714                                                            ;           FOLLOWING W/ 774               <====
5715  016440                    JSR1A:
5716  016440  012742  000420            MOV     #420,-(R2)      ;MOVE TO MAILBOX #  *******  420  *******
5717  016444  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5718  016446  000000                    HALT                    ;JSR MODE 1 FAILED
5719
5720  016450  022737  000001  017026    JSR3:   CMP     #1,@# JSRSEQ    ;CHECK SEQUENCE: JSRSEQ=1?
5721  016456  001014                    BNE     JSR3A           ;BRANCH IF OUT OF SEQUENCE
5722  016460  020127  016612            CMP     R1,#JSR4        ;PROPER PC SAVED?
5723  016464  001011                    BNE     JSR3A           ;BRANCH IF PC WRONG
5724  016466  022706  000776            CMP     #STBOT-2,R6     ;STACK POINTER DECREMENTED?
5725  016472  001006                    BNE     JSR3A           ;BRANCH IF SP WRONG
5726  016474  022716  125252            CMP     #125252,(R6)    ;REG SAVED ON STACK?
5727  016500  001003                    BNE     JSR3A           ;BRANCH IF REG. NOT SAVED
5728  016502  022700  016452            CMP     #JSR3+2,R0      ;MODE 2 INCREMENT CORRECT?
5729  016506  001404                    BEQ     JSR3B
5730                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5731                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5732                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5733                                                            ;           WHICH FOLLOWS W/ 737           <====
5734  016510                    JSR3A:
5735  016510  012742  000421            MOV     #421,-(R2)      ;MOVE TO MAILBOX #  *******  421  *******
5736  016514  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
5737  016516  000000                        HALT                    ;JSR MODE 3 MALFUNCTIONED
5738  016520  005237  017026        JSR3B:  INC     @#JSRSEQ        ;JPDATE SEQUENCE CHECKER
5739  016524  004137  016612                JSR     R1,@#JSR4       ;TRY JSR MODE 4
5740
5741  016530  005737  0i7026        JSR2:   TST     @#JSRSEQ        ;CHECK SEQUENCE: JSRSEQ=0?
5742  016534  001011                        BNE     JSR2A           ;BRANCH IF OUT OF SEQUENCE
5743  016536  020127  016440                CMP     R1,#JSR1A       ;PROPER PC SAVED?
5744  016542  001006                        BNE     JSR2A           ;BRANCH IF PC WRONG
5745  016544  022706  000776                CMP     #STBOT-2,R6     ;R6 DECREMENT?
5746  016550  001003                        BNE     JSR2A           ;BRANCH IF R6 IS INCORRECT
5747  016552  021627  177777                CMP     (R6),#-1        ;REGISTER SAVED?
5748  016556  001404                        BEQ     JSR2B
5749                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5750                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5751                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5752                                                                ;           WHICH FOLLOWS W/ 713           <====
5753  016560                        JSR2A:
5754  016560  012742  000422                MOV     #422,-(R2)      ;MOVE TO MAILBOX #  *******  422  *******
5755  016564  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5756  016566  000000                        HALT                    ;JSR MODE 1 MALFUNCTIONED
5757  016570  012706  001000        JSR2B:  MOV     #STBOT,R6       ;INITIALIZE R6
5758  016574  012701  125252                MOV     #125252,R1      ;INITIALIZE R1
5759  016600  005237  017026                INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5760  016604  012700  016450                MOV     #JSR3,R0        ;SET TARGET ADDRESS
5761  016610  004120                        JSR     R1,(R0)+        ;TRY JSR MODE 2
5762
5763  016612  022737  000002  017026 JSR4:   CMP     #2,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=2?
5764  016620  001003                        BNE     JSR4A           ;BRANCH IF OUT OF SEQUENCE
5765  016622  022701  016530                CMP     #JSR2,R1        ;PROPER PC SAVED?
5766  016626  001404                        BEQ     JSR4B
5767                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5768                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5769                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5770                                                                ;           WHICH FOLLOWS W/ 667           <====
5771  016630                        JSR4A:
5772  016630  012742  000423                MOV     #423,-(R2)      ;MOVE TO MAILBOX #  *******  423  *******
5773  016634  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5774  016636  000000                        HALT                    ;JSR MODE 3 MALFUNCTIONFD
5775  016640  005237  017026        JSR4B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5776  016644  012700  016720                MOV     #JSR5+2,R0      ;SET TARGET ADDRESS
5777  016650  004140                        JSR     R1,-(R0)        ;TRY JSR MODE 4
5778
5779  016652  022767  000004  000146 JSR6:   CMP     #4,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=4?
5780  016660  001006                        BNE     JSR6A           ;BRANCH IF OUT OF SEQUENCE
5781  016662  022701  016764                CMP     #JSR7,R1        ;PROPER PC SAVED?
5782  016666  001003                        BNE     JSR6A           ;BRANCH IF PC WRONG
5783  016670  022700  017022                CMP     #JSR6AD,R0      ;MODE 5 REGISTER CORRECT?
5784  016674  001404                        BEQ     JSR6B
5785                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5786                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5787                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5788                                                                ;           WHICH FOLLOWS W/ 644           <====
5789  016676                        JSR6A:
5790  016676  012742  000424                MOV     #424,-(R2)      ;MOVE TO MAILBOX #  *******  424  *******
5791  016702  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5792  016704  000000                        HALT                    ;JSR MODE 5 FAILED
```

```
5793  016706  005237  017026          JSR6B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5794  016712  004167  000046                  JSR     R1,JSR7         ;TRY JSR MODE 6
5795  016716  022767  000003  000102  JSR5:   CMP     #3,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=3?
5796  016724  001006                          BNE     JSR5A           ;BRANCH IF OUT OF SEQUENCE
5797  016726  022701  016652                  CMP     #JSR6,R1        ;PROPER PC SAVED?
5798  016732  001003                          BNE     JSR5A           ;BRANCH IF PC WRONG
5799  016734  022700  016716                  CMP     #JSR5,R0        ;CHECK MODE 4 REGISTER
5800  016740  001404                          BEQ     JSR5B
5801                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5802                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5803                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5804                                                                  ;           WHICH FOLLOWS W/ 622           <====
5805  016742                          JSR5A:
5806  016742  012742  000425                  MOV     #425,-(R2)      ;MOVE TO MAILBOX #  ******  425  *******
5807  016746  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5808  016750  000000                          HALT                    ;JSR MODE 4 MALFUNCTIONED
5809  016752  005237  017026          JSR5B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5810  016756  012700  017024                  MOV     #JSR6AD+2,R0    ;POINT R0 TO TARGET ADDRESS
5811  016762  004150                          JSR     R1,@-(R0)       ;TRY JSR MODE 5
5812
5813  016764  022737  000005  017026  JSR7:   CMP     #5,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=5?
5814  016772  001003                          BNE     JSR7A           ;BRANCH IF OUT OF SEQUENCE
5815  016774  022701  016716                  CMP     #JSR5,R1        ;PROPER PC SAVED?
5816  017000  001404                          BEQ     JSR7B
5817                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5818                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5819                                                                  ;           REPLACE THE MOVE INSTRUCTION   <== =
5820                                                                  ;           WHICH FOLLOWS W/ 602           <-===
5821  017002                          JSR7A:
5822  017002  012742  000426                  MOV     #426,-(R2)      ;MOVE TO MAILBOX #  *******  426  *******
5823  017006  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5824  017010  000000                          HALT                    ;JSR MODE 6 FAILED
5825  017012  005237  017026          JSR7B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5826  017016  004177  000002                  JSR     R1,@JSRCKAD     ;TRY JSR MODE 7
5827
5828  017022  016652                  JSR6AD: JSR6                    ;MODE 5 TARGET ADDRESS
5829  017024  017030                  JSRCKAD:JSRCK                   ;MODE 7 TARGET ADDRESS
5830  017026  000000                  JSRSEQ: 0                       ;SEQUENCE CHECKER
5831
5832  017030  022767  000006  177770  JSRCK:  CMP     #6,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=6?
5833  017036  001003                          BNE     JSRCK1          ;BRANCH IF OUT OF SEQUENCE
5834  017040  022701  017022                  CMP     #JSR6AD,R1      ;PROPER PC SAVED?
5835  017044  001404                          BEQ     TS203
5836                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5837                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5838                                                                  ;           REPLACE THE MOVE INSTRUCTION   <= -
5839                                                                  ;           WHICH FOLLOWS W/ 560           <====
5840  017046                          JSRCK1:
5841  017046  012742  000427                  MOV     #427,-(R2)      ;MOVE TO MAILBOX #  *******  427  *******
5842  017052  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5843  017054  000000                          HALT                    ;JSR MODE 7 MALFUNCTIONED
5844                                                                  ;  OR SEQUENCE ERROR
5845
5846
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 203       TEST RTS INSTRUCTION
                                ;*****************************************************************************
017056  005212                  TS203:  INC     (R2)            ;UPDATE TEST NUMBER
017060  022712  000203                  CMP     #203,(R2)       ;SEQUENCE ERROR?
017064  001016                          BNE     TS204-10        ;BR TO ERROR HALT ON SEQ ERROR
017066  012706  001000                  MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
017072  012746  052525                  MOV     #52525,-(R6)    ;INITIALIZE TOP OF STACK
017076  012700  017114                  MOV     #RTS1,R0        ;INITIALIZE RETURN REGISTER
017102  000200                          RTS     R0              ;TRY RTS THROUGH R0
                                                                ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           FOLLOWING W/ 770               <====
017104  012742  000430                  MOV     #430,-(R2)      ;MOVE TO MAILBOX #  *******  430  *******
017110  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017112  000000                          HALT                    ;RTS FAILED
017114  022700  052525          RTS1:   CMP     #52525,R0       ;CHECK THAT R0 RESTORED FROM STACK
017120  001404                          BEQ     TS204
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 761           <====
017122  012742  000431                  MOV     #431,-(R2)      ;MOVE TO MAILBOX #  *******  431  *******
017126  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017130  000000                          HALT                    ;RTS MALFUNCTIONED
                                                                ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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 VERIFIED WITH
                                ;A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.  THE DATA IS CHOSEN
                                ;TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
                                ;
                                ;*****************************************************************************************
                                ;TEST 204        TEST MOV INSTRUCTION
                                ;*****************************************************************************************
017132  005212                  TS204:  INC     (R2)            ;UPDATE TEST NUMBER
017134  022712  000204                  CMP     #204,(R2)       ;SEQUENCE ERROR?
017140  001022                          BNE     TS205-10        ;BR TO ERROR HALT ON SEQ ERROR
017142  000277                          SCC                     ;CC=0110
017144  000251                          +CLN!CLC
017146  012700  100000                  MOV     #100000,R0      ;CC=1000
017152  101402                          BLOS    MOV1
017154  102401                          BVS     MOV1
017156  100404                          BMI     MOV2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
017160                          MOV1:
017160  012742  000432                  MOV     #432,-(R2)      ;MOVE TO MAILBOX #  *******  432  *******
017164  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017166  000000                          HALT                    ;MOV DID NOT SET CC'S CORRECTLY
017170  000277                  MOV2:   SCC                     ;CC=1011
017172  000244                          CLZ
017174  012700  000000                  MOV     #0,R0           ;CC-0101
017200  101002                          BHI     MOV3            ;C OR Z = 0?
017202  102401                          BVS     MOV3            ;V-1?
017204  100004                          BPL     TS205
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 755           <====
017206                          MOV3:
017206  012742  000433                  MOV     #433,-(R2)      ;MOVE TO MAILBOX #  *******  433  *******
017212  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017214  000000                          HALT                    ;MOV DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************************
                                ;TEST 205        TEST BIT INSTRUCTION
                                ;*****************************************************************************************
017216  005212                  TS205:  INC     (R2)            ;UPDATE TEST NUMBER
017220  022712  000205                  CMP     #205,(R2)       ;SEQUENCE ERROR?
017224  001024                          BNE     TS206-10        ;BR TO ERROR HALT ON SEQ ERROR
```

```
5935  017226  012700  100001          MOV     #100001,R0
5936  017232  000277                  SCC                     ;CC=0110
5937  017234  000251                  +CLN!CLC
5938  017236  032700  100000          BIT     #100000,R0      ;CC=1000
5939  017242  101402                  BLOS    BIT1
5940  017244  102401                  BVS     BIT1
5941  017246  100404                  BMI     BIT2
5942                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5943                                  ;           CONDITIONAL BRANCH INST. AND   <====
5944                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5945                                  ;           WHICH FOLLOWS W/ 766           <====
5946  017250                  BIT1:
5947  017250  012742  000434          MOV     #434,-(R2)      ;MOVE TO MAILBOX #  *******  434  *******
5948  017254  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5949  017256  000000                  HALT                    :BIT DID NOT SET CC'S CORRECTLY
5950
5951  017260  000277          BIT2:   SCC                     ;CC=1011
5952  017262  000244                  CLZ
5953  017264  032700  077776          BIT     #77776,R0       ;CC=0101
5954  017270  101002                  BHI     BIT3
5955  017272  102401                  BVS     BIT3
5956  017274  100004                  BPL     TS206
5957                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5958                                  ;           CONDITIONAL BRANCH INST. AND   <====
5959                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5960                                  ;           WHICH FOLLOWS W/ 753           <====
5961  017276                  BIT3:
5962  017276  012742  000435          MOV     #435,-(R2)      ;MOVE TO MAILBOX #  *******  435  *******
5963  017302  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5964  017304  000000                  HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5965                                                          ;  OR SEQUENCE ERROR
5966  ;*************************************************************************************
5967  ;TEST 206       TEST BIC INSTRUCTION
5968  ;*************************************************************************************
5969  017306  005212          TS206:  INC     (R2)            ;UPDATE TEST NUMBER
5970  017310  022712  000206          CMP     #206,(R2)       ;SEQUENCE ERROR?
5971  017314  001024                  BNE     TS207-10        ;BR TO ERROR HALT ON SEQ ERROR
5972  017316  012700  177777          MOV     #177777,R0
5973  017322  000277                  SCC                     ;CC=0110
5974  017324  000251                  +CLN!CLC
5975  017326  042700  077777          BIC     #77777,R0       ;CC=1000
5976  017332  101402                  BLOS    BIC1
5977  017334  102401                  BVS     BIC1
5978  017336  100404                  BMI     BIC2
5979                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5980                                  ;           CONDITIONAL BRANCH INST. AND   <====
5981                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5982                                  ;           WHICH FOLLOWS W/ 766           <====
5983  017340                  BIC1:
5984  017340  012742  000436          MOV     #436,-(R2)      ;MOVE TO MAILBOX #  *******  436  *******
5985  017344  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5986  017346  000000                  HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5987  017350  000277          BIC2:   SCC                     ;CC=1011
5988  017352  000244                  CLZ
5989  017354  042700  100000          BIC     #100000,R0      ;CC=0101
5990  017360  101002                  BHI     BIC3
```

```
5991  017362  102401                BVS     BIC3
5992  017364  100004                BPL     TS207
5993                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5994                                                ;           CONDITIONAL BRANCH INST. AND   <====
5995                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5996                                                ;           WHICH FOLLOWS W/ 753           <====
5997  017366                    BIC3:
5998  017366  012742  000437          MOV     #437,-(R2)      ;MOVE TO MAILBOX #  *******  437  *******
5999  017372  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6000  017374  000000                  HALT                    ;BIC DID NOT SET CC'S CORRECTLY
6001                                                ;  OR SEQUENCE ERROR
6002                              ;*************************************************************************
6003                              ;TEST 207       TEST BIS INSTRUCTION
6004                              ;*************************************************************************
6005  017376  005212            TS207:  INC     (R2)            ;UPDATE TEST NUMBER
6006  017400  022712  000207           CMP     #207,(R2)       ;SEQUENCE ERROR?
6007  017404  001025                   BNE     TS210-10        ;BR TO ERROR HALT ON SEQ ERROR
6008  017406  005000                   CLR     R0              ;R0=0
6009  017410  000277                   SCC                     ;CC=1010
601C  017412  000251                   +CLN!CLC
6011  017414  052700  000000          BIS     #0,R0           ;CC=0100  R0=0
6012  017420  103403                   BCS     BIS1
6013  017422  102402                   BVS     BIS1
6014  017424  100401                   BMI     BIS1
6015  017426  001404                   BEQ     BIS2
6016                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6017                                                ;           CONDITIONAL BRANCH INST. AND   <====
6018                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6019                                                ;           WHICH FOLLOWS W/ 766           <====
6020  017430                    BIS1:
6021  017430  012742  000440          MOV     #440,-(R2)      ;MOVE TO MAILBUX #  *******  440  *******
6022  017434  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6023  017436  000000                  HALT                    ;BIS DID NOT SET CC'S CORRECTLY
6024  017440  000277            BIS2:   SCC                     ;CC=0111
6025  017442  000250                   CLN
6026  017444  052700  177777          BIS     #177777,R0      ;CC=1001
6027  017450  103003                   BCC     BIS3
6028  017452  102402                   BVS     BIS3
6029  017454  001401                   BEQ     BIS3
6030  017456  100404                   BMI     TS210
6031                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6032                                                ;           CONDITIONAL BRANCH INST. AND   <====
6033                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6034                                                ;           WHICH FOLLOWS W/ 752           <====
6035  017460                    BIS3:
6036  017460  012742  000441          MOV     #441,-(R2)      ;MOVE TO MAILBOX #  *******  441  *******
6037  017464  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6038  017466  000000                  HALT                    ;BIS DID NOT SET CC'S CORRECTLY
6039                                                ;  OR SEQUENCE ERROR
```

```
6040
6041                    ;*****************************************************************************************
6042                    ;
6043                    ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
6044                    ;DEC INSTRUCTIONS.  THESE INSTRUCTIONS BOTH EFFECT THE C AND V
6045                    ;BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
6046                    ;UPON THE DATA RESULTS.  THE SAME PROCEDURE IS USED.  THE CONDITION
6047                    ;CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
6048                    ;RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
6049                    ;THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
6050                    ;DIFFERENT COMBINATIONS OF THE C AND V BITS.
6051                    ;
6052                    ;*****************************************************************************************
6053                    ;TEST 210       TEST INC INSTRUCTION
6054                    ;*****************************************************************************************
6055  017470  005212    TS210:  INC     (R2)            ;UPDATE TEST NUMBER
6056  017472  022712  000210           CMP     #210,(R2)       ;SEQUENCE ERROR?
6057  017476  001037                   BNE     TS211-10        ;BR TO ERROR HALT ON SEQ ERROR
6058  017500  012700  077777           MOV     #077777,R0      ;R0-077777
6059  017504  000257                   CCC                     ;CC-0100
6060  017506  000264                   SEZ
6061  017510  005200                   INC     R0              ;CC=1010   R0-10000
6062  017512  101402                   BLOS    INC1
6063  017514  100001                   BPL     INC1
6064  017516  102404                   BVS     INC2
6065                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <== -
6066                                                    ;           CONDITIONAL BRANCH INST. AND   <== -
6067                                                    ;           REPLACE THE MOVE INSTRUCTION   <   -
6068                                                    ;           WHICH FOLLOWS W/ 767           <- --
6069  017520            INC1:
6070  017520  012742  000442           MOV     #442,-(R2)      ;MOVE TO MAILBOX #  *******  442  *******
6071  017524  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6072  017526  000000                   HALT                    ;INC DID NOT SET CC'S CORRECTLY
6073  017530  052700  077777   INC2:   BIS     #77777,R0       ;R0=177777
6074  017534  000261                   SEC                     ;CC=1011
6075  017536  000244                   CLZ
6076  017540  005200                   INC     R0              ;CC=0101   R0=0
6077  017542  100403                   BMI     INC3
6078  017544  102402                   BVS     INC3
6079  017546  103001                   BCC     INC3
6080  017550  001404                   BEQ     INC4
6081                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6082                                                    ;           CONDITIONAL BRANCH INST. AND   <====
6083                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
6084                                                    ;           WHICH FOLLOWS W/ 752           <====
6085  017552            INC3:
6086  017552  012742  000443           MOV     #443,-(R2)      ;MOVE TO MAILBOX #  *******  443  *******
6087  017556  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6088  017560  000000                   HALT                    ;INC DID NOT SET CC'S CORRECTLY
6089
6090  017562  000277           INC4:   SCC                     ;CC=1110
6091  017564  000241                   CLC
6092  017566  005200                   INC     R0              ;CC=0000   R0=1
6093  017570  101402                   BLOS    INC5
6094  017572  100401                   BMI     INC5
6095  017574  100004                   BPL     TS211
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 740           <====
017576                          INC5:
017576  012742  000444                MOV     #444,-(R2)      ;MOVE TO MAILBOX #  *******  444  *******
017602  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017604  000000                        HALT                    ;INC DID NOT SET CC'S CORRECTLY
                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 211       TEST DEC INSTRUCTION
                                ;*****************************************************************************
017606  005212                  TS211:  INC     (R2)            ;UPDATE TEST NUMBER
017610  022712  000211                CMP     #211,(R2)       ;SEQUENCE ERROR?
017614  001051                        BNE     TS212-10        ;BR TO ERROR HALT ON SEQ ERROR
017616  012700  000002                MOV     #2,R0           ;R0=2
017622  000277                        SCC                     ;CC=1111
017624  005300                        DEC     R0              ;CC=0001   R0-1
017626  100403                        BMI     DEC1
017630  001402                        BEQ     DEC1
017632  102401                        BVS     DEC1
017634  103404                        BCS     DEC2
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 767           <====
017636                          DEC1:
017636  012742  000445                MOV     #445,-(R2)      ;MOVE TO MAILBOX #  *******  445  *******
017642  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017644  000000                        HALT                    ;DEC DID NOT SET CC'S CORRECTLY
017646  000261                  DEC2:   SEC                     ;CC=1011
017650  000244                        CLZ
017652  005300                        DEC     R0              ;CC=0101   R0=0
017654  101002                        BHI     DEC3
017656  100401                        BMI     DEC3
017660  102004                        BVC     DEC4
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 755           <====
017662                          DEC3:
017662  012742  000446                MOV     #446,-(R2)      ;MOVE TO MAILBOX #  *******  446  *******
017666  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017670  000000                        HALT                    ;DEC DID NOT SET CC'S CORRECTLY
017672  000277                  DEC4:   SCC                     ;CC=0110
017674  000251                        +CLN!CLC
017676  005300                        DEC     R0              ;CC=1000   R0=177777
017700  101402                        BLOS    DEC5
017702  102401                        BVS     DEC5
017704  100404                        BMI     DEC6
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 743           <====
017706                          DEC5:
```

```
6152  017706  012742  000447          MOV     #447,-(R2)      ;MOVE TO MAILBOX #  *******  447  *******
6153  017712  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6154  017714  000000                  HALT                    ;DEC DID NOT SET CC'S CORRECTLY
6155  017716  042700  077777  DEC6:   BIC     #77777,R0       ;R0=100000
6156  017722  000277                  SCC                     ;CC=0101
6157  017724  000252                  +CLN!CLV
6158  017726  005300                  DEC     R0              ;CC=1011   R0=77777
6159  017730  100403                  BMI     DEC7            ;CC=0011
6160  017732  001402                  BEQ     DEC7
6161  017734  102001                  BVC     DEC7
6162  017736  103404                  BCS     TS212
6163                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
6164                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6165                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6166                                                          ;           WHICH FOLLOWS W/ 726           <====
6167  017740                  DEC7:
6168  017740  012742  000450          MOV     #450,-(R2)      ;MOVE TO MAILBOX #  *******  450  *******
6169  017744  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6170  017746  000000                  HALT                    ;DEC DID NOT SET CC'S CORRECTLY
6171                                                          ;  OR SEQUENCE ERROR
6172
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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 212        TEST CLR INSTRUCTION
                                ;*****************************************************************************************
017750  005212                  TS212:  INC     (R2)            ;UPDATE TEST NUMBER
017752  022712  000212                  CMP     #212,(R2)       ;SEQUENCE ERROR?
017756  001007                          BNE     TS213-10        ;BR TO ERROR HALT ON SEQ ERROR
017760  000277                          SCC                     ;CC=1011
017762  000244                          CLZ
017764  005000                          CLR     R0              ;CC=0'00   R0=0
017766  100403                          BMI     CLR1
017770  102402                          BVS     CLR1
017772  103401                          BCS     CLR1
017774  001404                          BEQ     TS213
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
017776                          CLR1:
017776  012742  000451                  MOV     #451,-(R2)      ;MOVE TO MAILBOX #  *******  451  *******
020002  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020004  000000                          HALT                    ;CLR DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************************
                                ;TEST 213        TEST TST INSTRUCTION
                                ;*****************************************************************************************
020006  005212                  TS213:  INC     (R2)            ;UPDATE TEST NUMBER
020010  022712  000213                  CMP     #213,(R2)       ;SEQUENCE ERROR?
020014  001022                          BNE     TS214-10        ;BR TO ERROR HALT ON SEQ ERROR
020016  000277                          SCC                     ;CC=1011
020020  000244                          CLZ
020022  005700                          TST     R0              ;CC=0100
020024  100403                          BMI     TEST1
020026  102402                          BVS     TEST1
020030  103401                          BCS     TEST1
020032  001404                          BEQ     TEST2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
020034                          TEST1:
020034  012742  000452                  MOV     #452,-(R2)      ;MOVE TO MAILBOX #  *******  452  *******
020040  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020042  000000                          HALT                    ;TEST DID NOT SET CC'S CORRECTLY
020044  005300                  TEST2:  DEC     R0              ;MAKE R0 NEGATIVE
020046  000277                          SCC                     ;CC=0111
```

```
6229  020050  000250                          CLN
6230  020052  005700                          TST     R0              ;CC=1000
6231  020054  101402                          BLOS    TEST3
6232  020056  102401                          BVS     TEST3
6233  020060  100404                          BMI     TS214
6234                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6235                                          ;           CONDITIONAL BRANCH INST. AND   <====
6236                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6237                                          ;           WHICH FOLLOWS W/ 755           <====
6238  020062                          TEST3:
6239  020062  012742  000453                  MOV     #453,-(R2)      ;MOVE TO MAILBOX #  *******  453  *******
6240  020066  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6241  020070  000000                          HALT                    ;TEST DID NOT SET CC'S CORRECTLY
6242                                          ;  OR SEQUENCE ERROR
6243                            ;*************************************************************************************
6244                            ;TEST 214       TEST SWAB INSTRUCTION
6245                            ;*************************************************************************************
6246  020072  005212            TS214:  INC     (R2)            ;UPDATE TEST NUMBER
6247  020074  022712  000214            CMP     #214,(R2)       ;SEQUENCE ERROR?
6248  020100  001023                    BNE     TS215-10        ;BR TO ERROR HALT ON SEQ ERROR
6249  020102  012700  170000            MOV     #170000,R0      ;R0=170000
6250  020106  000277                    SCC                     ;CC=0111
6251  020110  000250                    CLN
6252  020112  000300                    SWAB    R0              ;CC=1000   R0=360
6253  020114  101402                    BLOS    SWB1
6254  020116  102401                    BVS     SWB1
6255  020120  100404                    BMI     SWB2
6256                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6257                                          ;           CONDITIONAL BRANCH INST. AND   <====
6258                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6259                                          ;           WHICH FOLLOWS W/ 767           <====
6260  020122                          SWB1:
6261  020122  012742  000454                  MOV     #454,-(R2)      ;MOVE TO MAILBOX #  *******  454  *******
6262  020126  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6263  020130  000000                          HALT                    ;SWAB DID NOT SET CC'S CORRECTLY
6264  020132  000277                  SWB2:   SCC                     ;CC=1011
6265  020134  000244                          CLZ
6266  020136  000300                          SWAB    R0              ;CC=0100   R0=170000
6267  020140  102403                          BVS     SWB3
6268  020142  103402                          BCS     SWB3
6269  020144  100401                          BMI     SWB3
6270  020146  001404                          BEQ     TS215
6271                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===
6272                                          ;           CONDITIONAL BRANCH INST. AND   <=
6273                                          ;           REPLACE THE MOVE INSTRUCTION   <
6274                                          ;           WHICH FOLLOWS W/ 754           <
6275  020150                          SWB3:
6276  020150  012742  000455                  MOV     #455,-(R2)      ;MOVE TO MAILBOX #  *******  455  *******
6277  020154  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6278  020156  000000                          HALT                    ;
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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 215       TEST ADD INSTRUCTION
                                ;*****************************************************************************************
020160  005212                  TS215:  INC     (R2)            ;UPDATE TEST NUMBER
020162  022712  000215                  CMP     #215,(R2)       ;SEQUENCE ERROR?
020166  001062                          BNE     TS216-10        ;BR TO ERROR HALT ON SEQ ERROR
020170  012700  040000                  MOV     #40000,R0       ;R0=40000
020174  000277                          SCC                     ;CC=1111
020176  062700  030000                  ADD     #30000,R0       ;CC=0000   R0=70000
020202  101402                          BLOS    ADD1
020204  102401                          BVS     ADD1
020206  100004                          BPL     ADD2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-=-=
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <==-=
020210                          ADD1:
020210  012742  000456                  MOV     #456,-(R2)      ;MOVE TO MAILBOX #  *******  456  *******
020214  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020216  000000                          HALT                    ;ADD DID NOT SET CC'S CORRECT_Y
020220  000264                  ADD2:   SEZ                     ;CC=0100
020222  062700  010000                  ADD     #10000,R0       ;CC=1010   40=100000
020226  101402                          BLOS    ADD3
020230  102001                          BVC     ADD3
020232  100404                          BMI     ADD4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-=
                                                                ;           CONDITIONAL BRANCH INST. AND   <-=  
                                                                ;           REPLACE THE MOVE INSTRUCTION   <-=  -
                                                                ;           WHICH FOLLOWS W/ 755           <  =
020234                          ADD3:
020234  012742  000457                  MOV     #457,-(R2)      ;MOVE TO MAILBOX #  *******  457  *******
020240  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020242  000000                          HALT                    ;ADD DID NOT SET CC'S CORRECTLY
020244  000257                  ADD4:   CCC                     ;CC-1000
020246  000270                          SEN
020250  062700  100000                  ADD     #100000,R0      ;CC=0111   R0=0
020254  101002                          BHI     ADD5
020256  102001                          BVC     ADD5
020260  100004                          BPL     ADD6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=  ==
                                                                ;           CONDITIONAL BRANCH INST. AND   <=-==
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 742           <=  =-
020262                          ADD5:
```

```
6335  020262  012742  000460          MOV     #460,-(R2)      ;MOVE TO MAILBOX #  *******  460  *******
6336  020266  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6337  020270  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6338  020272  062700  177777  ADD6:   ADD     #177777,R0      ;CC=1000   R0=177777
6339  020276  101402                  BLOS    ADD7
6340  020300  102401                  BVS     ADD7
6341  020302  100404                  BMI     ADD8
6342                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6343                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6344                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6345                                                          ;           WHICH FOLLOWS W/ 731           <====
6346  020304          ADD7:
6347  020304  012742  000461          MOV     #461,-(R2)      ;MOVE TO MAILBOX #  *******  461  *******
6348  020310  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6349  020312  000000                  HALT                    ;ADD DID NOT SET CC'S COPRECTLY
6350  020314  000277  ADD8:   SCC                             ;CC-1010
6351  020316  000245                  +CLC!CLZ
6352  020320  062700  000001          ADD     #1,R0           ;CC=0101   R=0
6353  020324  102403                  BVS     ADD9
6354  020326  103002                  BCC     ADD9
6355  020330  100401                  BMI     ADD9
6356  020332  001404                  BEQ     TS216
6357                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6358                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6359                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6360                                                          ;           WHICH FOLLOWS W/ 715           <====
6361  020334          ADD9:
6362  020334  012742  000462          MOV     #462,-(R2)      ;MOVE TO MAILBOX #  *******  462  *******
6363  020340  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6364  020342  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6365                                                          ;  OR SEQUENCE ERROR
6366
6367                    ;*****************************************************************************************
6368                    ;TEST 216       TEST ADC INSTRUCTION
6369                    ;*****************************************************************************************
6370  020344  005212  TS216:  INC     (R2)            ;UPDATE TEST NUMBER
6371  020346  022712  000216          CMP     #216,(R2)       ;SEQUENCE ERROR?
6372  020352  001037                  BNE     TS217-10        ;BR TO ERROR HALT ON SEQ ERROR
6373  020354  012700  077777          MOV     #077777,R0
6374  020360  000277                  SCC                     ;CC=0101
6375  020362  000252                  +CLN!CLV
6376  020364  005500                  ADC     R0              ;CC=1010
6377  020366  101402                  BLOS    ADC1
6378  020370  102001                  BVC     ADC1
6379  020372  100404                  BMI     ADC2
6380                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= =-
6381                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6382                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6383                                                          ;           WHICH FOLLOWS W/ 767           <==--
6384  020374          ADC1:
6385  020374  012742  000463          MOV     #463,-(R2)      ;MOVE TO MAILBOX #  *******  463  *******
6386  020400  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6387  020402  000000                  HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6388  020404  052700  077777  ADC2:   BIS     #77777,R0
6389  020410  000277                  SCC                     ;CC=1011
6390  020412  000244                  CLZ
```

```
6391  020414  005500                        ADC     RC              ;CC=0101   R0=0
6392  020416  101002                        BHI     ADC3
6393  020420  102401                        BVS     ADC3
6394  020422  100004                        BPL     ADC4
6395                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6396                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
6397                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6398                                                                  ;           WHICH FOLLOWS W/ 753           <====
6399  020424                          ADC3:
6400  020424  012742  000464                MOV     #464,-(R2)      ;MOVE TO MAILBOX #  *******  464  *******
6401  020430  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6402  020432  000000                        HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6403  020434  000277                ADC4:   SCC
6404  020436  000245                        +CLZ!CLC                        ;CC=1010
6405  020440  005500                        ADC     R0              ;CC=0100
6406  020442  102403                        BVS     ADC5
6407  020444  103402                        BCS     ADC5
6408  020446  100401                        BMI     ADC5
6409  020450  001404                        BEQ     TS217
6410                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6411                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
6412                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6413                                                                  ;           WHICH FOLLOWS W/ 740           <====
6414  020452                          ADC5:
6415  020452  012742  000465                MOV     #465,-(R2)      ;MOVE TO MAILBOX #  *******  465  *******
6416  020456  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6417  020460  000000                        HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6418                                                                  ;  OR SEQUENCE ERROR
```

```
6419
6420                                    ;*****************************************************************************************
6421                                    ;
6422                                    ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
6423                                    ;CMP, AND COM INSTRUCTIONS.  EACH OF THESE INSTRUCTIONS GENERATE
6424                                    ;THE C AND V BITS IDENTICALLY.  THE CONDITION CODES ARE PRESET,
6425                                    ;THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
6426                                    ;OF CONDITIONAL BRANCH INSTRUCTIONS.  THIS PROCEDURE IS REPEATED
6427                                    ;SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
6428                                    ;COMBINATIONS OF THE C AND V BITS.
6429                                    ;
6430                                    ;*****************************************************************************************
6431                                    ;TEST 217       TEST NEG INSTRUCTION
6432                                    ;*****************************************************************************************
6433  020462  005212                   TS217:  INC     (R2)            ;UPDATE TEST NUMBER
6434  020464  022712  000217                   CMP     #217,(R2)       ;SEQUENCE ERROR?
6435  020470  001042                           BNE     TS220-10        ;BR TO ERROR HALT ON SEQ ERROR
6436  020472  012700  000001                   MOV     #1,R0
6437  020476  000277                           SCC                     ;CC=0110
6438  020500  000251                           +CLN!CLC
6439  020502  005400                           NEG     R0              ;CC=1001   R0=177777
6440  020504  103003                           BCC     NEG1
6441  020506  102402                           BVS     NEG1
6442  020510  001401                           BEQ     NEG1
6443  020512  100404                           BMI     NEG2
6444                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6445                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
6446                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
6447                                                                   ;           WHICH FOLLOWS W/ 766           <====
6448  020514                           NEG1:
6449  020514  012742  000466                   MOV     #466,-(R2)      ;MOVE TO MAILBOX #  *******  466  *******
6450  020520  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6451  020522  000000                           HALT                    ;NEG DID NOT SET CC'S CORRECTLY
6452  020524  042700  077777           NEG2:   BIC     #77777,R0
6453  020530  000257                           CCC                     ;CC=0100
6454  020532  000264                           SEZ
6455  020534  005400                           NEG     R0              ;CC=1011   R0=100000
6456  020536  102003                           BVC     NEG3
6457  020540  103002                           BCC     NEG3
6458  020542  001401                           BEQ     NEG3
6459  020544  100404                           BMI     NEG4
6460                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6461                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
6462                                                                   ;           REPLACE THE MOVE INSTRUCTION   <=--=
6463                                                                   ;           WHICH FOLLOWS W/ 751           <--==
6464  020546                           NEG3:
6465  020546  012742  000467                   MOV     #467,-(R2)      ;MOVE TO MAILBOX #  *******  467  *******
6466  020552  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6467  020554  000000                           HALT                    ;NEG DID NOT SET CC'S CORRECTLY
6468  020556  005000           NEG4:   CLR     R0
6469  020560  000277                           SCC                     ;CC=1011
6470  020562  000244                           CLZ
6471  020564  005400                           NEG     R0              ;CC=0100   R0=0
6472  020566  102403                           BVS     NEG5
6473  020570  103402                           BCS     NEG5
6474  020572  001001                           BNE     NEG5
```

```
6475  020574  100004                  BPL     TS220
6476                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6477                                                  ;           CONDITIONAL BRANCH INST. AND   <==-=
6478                                                  ;           REPLACE THE MOVE INSTRUCTION   <==-=
6479                                                  ;           WHICH FOLLOWS W/ 735           <===-
6480  020576                  NEG5:
6481  020576  012742  000470          MOV     #470,-(R2)      ;MOVE TO MAILBOX #  *******  470  *******
6482  020602  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6483  020604  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
6484                                                          ;  OR SEQUENCE ERROR
6485
6486                          ;*****************************************************************************
6487                          ;TEST 220       TEST CMP INSTRUCTION
6488                          ;*****************************************************************************
6489  020606  005212          TS220:  INC     (R2)            ;UPDATE TEST NUMBER
6490  020610  022712  000220          CMP     #220,(R2)       ;SEQUENCE ERROR?
6491  020614  001060                  BNE     TS221-10        ;BR TO ERROR HALT ON SEQ ERROR
6492  020616  012700  000005          MOV     #5,R0
6493  020622  000257                  CCC                     ;CC=1010
6494  020624  000271                  +SEN.SEC
6495  020626  022700  000005          CMP     #5,R0           ;CC=0101
6496  020632  101002                  BHI     CMP1
6497  020634  102401                  BVS     CMP1
6498  020636  100004                  BPL     CMP2
6499                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=-
6500                                                  ;           CONDITIONAL BRANCH INST. AND   <= ==
6501                                                  ;           REPLACE THE MOVE INSTRUCTION   <-_--
6502                                                  ;           WHICH FOLLOWS W/ 766           <-==-
6503  020640                  CMP1:
6504  020640  012742  000471          MOV     #471,-(R2)      ;MOVE TO MAILBOX #  *******  471  *******
6505  020644  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6506  020646  000000                  HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6507  020650  012700  100000  CMP2:   MOV     #100000,R0
6508  020654  000277                  SCC                     ;CC-1101
6509  020656  000242                  CLV
6510  020660  020027  077777          CMP     R0,#77777       ;CC=0010
6511  020664  101402                  BLOS    CMP3
6512  020666  102001                  BVC     CMP3
6513  020670  100004                  BPL     CMP4
6514                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6515                                                  ;           CONDITIONAL BRANCH INST. AND   <====
6516                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6517                                                  ,           WHICH FOLLOWS W/ 751           <====
6518  020672                  CMP3:
6519  020672  012742  000472          MOV     #472,-(R2)      ;MOVE TO MAILBOX #  *******  472  *******
6520  020676  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6521  020700  000000                  HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6522  020702  052700  040000  CMP4:   BIS     #40000,R0       ;R0=140000
6523  020706  000257                  CCC                     ;CC=0100
6524  020710  000264                  SEZ
6525  020712  022700  040000          CMP     #40000,R0       ;CC=1011
6526  020716  102003                  BVC     CMP5
6527  020720  103002                  BCC     CMP5
6528  020722  001401                  BEQ     CMP5
6529  020724  100404                  BMI     CMP6
6530                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 733           <====
020726                          CMP5:
020726 012742 000473                MOV     #473,-(R2)      ;MOVE TO MAILBOX #  *******  473  *******
020732 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020734 000000                       HALT                    ;CMP DID NOT SET CC'S CORRECTLY
020736 042700 040000        CMP6:   BIC     #40000,R0
020742 000277                       SCC                     ;CC=1111
020744 022700 177777                CMP     #-1,R0          ;CC=0000
020750 101402                       BLOS    CMP7
020752 102401                       BVS     CMP7
020754 100004                       BPL     TS221
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 717           <====
020756                          CMP7:
020756 012742 000474                MOV     #474,-(R2)      ;MOVE TO MAILBOX #  *******  474  *******
020762 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020764 000000                       HALT                    ;CMP DID NOT SET CC'S CORRECTLY
                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 221       TEST COM INSTRUCTION
                                ;*******************************************************************************
020766 005212                   TS221:  INC     (R2)            ;UPDATE TEST NUMBER
020770 022712 00022?                    CMP     #221,(R2)       ;SEQUENCE ERROR?
020774 001010                           BNE     TS222-10        ;BR TO ERROR HALT ON SEQ ERROR
020776 012700 177777                    MOV     #-1,R0
021002 000257                           CCC                     ;CC-1010
021004 000265                           +SEC!SEZ
021006 005100                           COM     R0              ;CC=0101
021010 101002                           BHI     COM1
021012 102401                           BVS     COM1
021014 100004                           BPL     TS222
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <= --
                                                ;           CONDITIONAL BRANCH INST. AND   < --=
                                                ;           REPLACE THE MOVE INSTRUCTION   <- --
                                                ;           WHICH FOLLOWS W/ 767           <-
021016                          COM1:
021016 012742 000475                MOV     #475,-(R2)      ;MOVE TO MAILBOX #  *******  475  *******
021022 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021024 000000                       HALT                    ;COM DID NOT SET CC'S CORRECTLY
                                                ;  OR SEQUENCE ERROR
```

```
6577
6578                            ;*****************************************************************************
6579                            ;
6580                            ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
6581                            ;AND SBC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE
6582                            ;C AND V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
6583                            ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
6584                            ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
6585                            ;BRANCHES.  THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
6586                            ;DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
6587                            ;
6588                            ;*****************************************************************************
6589                            ;TEST 222       TEST SUB INSTRUCTION
6590                            ;*****************************************************************************
6591  021026  005212            TS222:  INC     (R2)            ;UPDATE TEST NUMBER
6592  021030  022712  000222            CMP     #222,(R2)       ;SEQUENCE ERROR?
6593  021034  001055            BNE     TS223-10        ;BR TO ERROR HALT ON SEQ ERROR
6594  021036  012700  125252            MOV     #125252,R0
6595  021042  000257            CCC                     ;CC=1010
6596  021044  000271            +SEN!SEC
6597  021046  162700  125252            SUB     #125252,R0      ;CC=0101   R0=0
6598  021052  101002            BHI     SUB1
6599  021054  102401            BVS     SUB1
6600  021056  100004            BPL     SUB2
6601                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-
6602                                                    ;           CONDITIONAL BRANCH INST. AND   <  -
6603                                                    ;           REPLACE THE MOVE INSTRUCTION   <=
6604                                                    ;           WHICH FOLLOWS W/ 766           <---
6605  021060            SUB1:
6606  021060  012742  000476            MOV     #476,-(R2)      ;MOVE TO MAILBOX #  *******  476  *******
6607  021064  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6608  021066  000000            HALT                    ;SUB DID NOT SET CC'S CORRECTLY
6609  021070  052700  100000  SUB2:    BIS     #100000,R0
6610  021074  000277            SCC                     ;CC=1101
6611  021076  000242            CLV
6612  021100  162700  077777            SUB     #77777,R0       ;CC=0010   R0=1
6613  021104  101402            BLOS    SUB3
6614  021106  102001            BVC     SUB3
6615  021110  100004            BPL     SUB4
6616                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
6617                                                    ;           CONDITIONAL BRANCH INST. AND   <  ==
6618                                                    ;           REPLACE THE MOVE INSTRUCTION   <=-==
6619                                                    ;           WHICH FOLLOWS W/ 751           <==--
6620  021112            SUB3:
6621  021112  012742  000477            MOV     #477,-(R2)      ;MOVE TO MAILBOX #  *******  477  *******
6622  021116  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6623  021120  000000            HALT                    ;
6624  021122  005100  SUB4:    COM     R0              ;R0=177777
6625  021124  000277            SCC                     ;CC=11111
6626
6627  021126  162700  100000            SUB     #100000,R0      ;CC=0000   R0=77777
6628  021132  101402            BLOS    SUB5
6629  021134  102401            BVS     SUB5
6630  021136  100004            BPL     SUB6
6631                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
6632                                                    ;           CONDITIONAL BRANCH INST. AND   <==-
```

```
6633                                                    ;           REPLACE THE MOVE INSTRUCTION   <-- --
6634                                                    ;           WHICH FOLLOWS W/ 736           <--
6635  021140                            SUB5:
6636  021140  012742  000500                  MOV     #500,-(R2)    ;MOVE TO MAILBOX #  *******  500  *******
6637  021144  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6638  021146  000000                        HALT                  ;SUB DID NOT SET CC'S CORRECTLY
6639  021150  000257                  SUB6: CCC                   ;CC=0100
6640  021152  000264                        SEZ
6641  021154  162700  140000              SUB     #140000,R0    ;CC=1011
6642  021160  102003                        BVC     SUB7
6643  021162  103002                        BCC     SUB7
6644  021164  001401                        BEQ     SUB7
6645  021166  100404                        BMI     TS223
6646                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
6647                                                    ;           CONDITIONAL BRANCH INST. AND   <---
6648                                                    ;           REPLACE THE MOVE INSTRUCTION   <---
6649                                                    ;           WHICH FOLLOWS W/ 722           <---
6650  021170                            SUB7:
6651  021170  012742  000501                  MOV     #501,-(R2)    ;MOVE TO MAILBOX #  *******  501  *******
6652  021174  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6653  021176  000000                        HALT                  ;
6654
6655                                    ;*****************************************************************************
6656                                    ;TEST 223       TEST SBC INSTRUCTION
6657                                    ;*****************************************************************************
6658  021200  005212                  TS223: INC     (R2)          ;UPDATE TEST NUMBLR
6659  021202  022712  000223              CMP     #223,(R2)     ;SEQUENCE ERROR?
6660  021206  001053                        BNE     TS224-10      ;BR TO ERROR HALT ON SEQ ERROR
6661  021210  012700  000001              MOV     #1,R0
6662  021214  000277                        SCC                   ;CC=1011
6663  021216  000244                        CLZ
6664  021220  005600                        SBC     R0            ;CC=0100   R-0
6665  021222  103403                        BCS     SBC1
6666  021224  102402                        BVS     SBC1
6667  021226  100401                        BMI     SBC1
6668  021230  001404                        BEQ     SBC2
6669                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <---
6670                                                    ;           CONDITIONAL BRANCH INST. AND   <---
6671                                                    ;           REPLACE THE MOVE INSTRUCTION   <---
6672                                                    ;           WHICH FOLLOWS W/ 766           <---
6673  021232                            SBC1:
6674  021232  012742  000502                  MOV     #502,-(R2)    ;MOVE TO MAILBOX #  *******  502  *******
6675  021236  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6676  021240  000000                        HALT                  ;SBC DID NOT SET CC'S CORRECTLY
6677  021242  000277                  SBC2: SCC                   ;CC-1010
6678  021244  000245                        +CLZ!CLC
6679  021246  005600                        SBC     R0            ;CC-0100   R=0
6680  021250  103403                        BCS     SBC3
6681  021252  102402                        BVS     SBC3
6682  021254  100401                        BMI     SBC3
6683  021256  001404                        BEQ     SBC4
6684                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <---
6685                                                    ;           CONDITIONAL BRANCH INST. AND   <---
6686                                                    ;           REPLACE THE MOVE INSTRUCTION   <---
6687                                                    ;           WHICH FOLLOWS W/ 753           <---
6688  021260                            SBC3:
```

```
6689  021260  012742  000503          MOV     #503,-(R2)      ;MOVE TO MAILBOX #  *******  503  *******
6690  021264  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6691  021266  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6692  021270  000277          SBC4:   SCC                     ;CC=0111
6693  021272  000250                  CLN
6694  021274  005600                  SBC     R0              ;CC=1001   R0=177777
6695  021276  103003                  BCC     SBC5
6696  021300  102402                  BVS     SBC5
6697  021302  001401                  BEQ     SBC5
6698  021304  100404                  BMI     SBC6
6699                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6700                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6701                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6702                                                          ;           WHICH FOLLOWS W/ 740           <====
6703  021306          SBC5:
6704  021306  012742  000504          MOV     #504,-(R2)      ;MOVE TO MAILBOX #  *******  504  *******
6705  021312  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6706  021314  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6707  021316  042700  077777  SBC6:   BIC     #77777,R0       ;R0=100000
6708  021322  000277                  SCC                     ;CC=1101
6709  021324  000242                  CLV
6710  021326  005600                  SBC     R0              ;CC=0010
6711  021330  101402                  BLOS    SBC7
6712  021332  102001                  BVC     SBC7
6713  021334  100004                  BPL     TS224
6714                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6715                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6716                                                          ;           REPLACE THE MOVE INSTRUCTION   <----
6717                                                          ;           WHICH FOLLOWS W/ 724           <====
6718  021336          SBC7:
6719  021336  012742  000505          MOV     #505,-(R2)      ;MOVE TO MAILBOX #  *******  505  *******
6720  021342  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6721  021344  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6722                                                          ;  OR SEQUENCE ERROR
6723
```

```
6724                                ;*****************************************************************************
6725                                ;
6726                                ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
6727                                ;ROR, ASL AND ASR INSTRUCTIONS.  SPECIAL DATA PATTERNS ARE LOADED
6728                                ;AND ROTATED SEVERAL TIMES FOR EACH TEST.  THE CONDITION CODES
6729                                ;ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
6730                                ;CHECKED AFTER EACH ROTATION.  THE FINAL CHECK IN EACH TEST IS
6731                                ;TO VERIFY THE COMMULATIVE DATA RESULT.  THE DATA PATTERNS HAVE
6732                                ;BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.
6733                                ;
6734                                ;*****************************************************************************
6735                                ;TEST 224       TEST ROL INSTRUCTION
6736                                ;*****************************************************************************
6737  021346  005212                T224:   INC     (R2)            ;UPDATE TEST NUMBER
6738  021350  022712  000224                CMP     #224,(R2)       ;SEQUENCE ERROR?
6739  021354  001053                        BNE     TS225-10        ;BR TO ERROR HALT ON SEQ ERROR
6740  021356  012700  144000                MOV     #144000,R0      ;R0=144000
6741  021362  000257                        CCC                     ;CC=0110
6742  021364  000266                        +SEZ!SEV
6743  021366  006100                        ROL     R0              ;CC=1001   R0=110000
6744  021370  103003                        BCC     ROL1
6745  021372  102402                        BVS     ROL1
6746  021374  001401                        BEQ     ROL1
6747  021376  100404                        BMI     ROL2
6748                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6749                                                                ;           CONDITIONAL BRANCH INST. AND   <====
6750                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6751                                                                ;           WHICH FOLLOWS W/ 766           <====
6752  021400                        ROL1:
6753  021400  012742  000506                MOV     #506,-(R2)      ;MOVE TO MAILBOX #  *******  506  *******
6754  021404  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6755  021406  000000                        HALT                    ;
6756  021410  000277                ROL2:   SCC                     ;CC=1100
6757  021412  000243                        +CLV!CLC
6758  021414  006100                        ROL     R0              ;CC-0011   R0=020000
6759  021416  103003                        BCC     ROL3
6760  021420  102002                        BVC     ROL3
6761  021422  001401                        BEQ     ROL3
6762  021424  100004                        BPL     ROL4
6763                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6764                                                                ;           CONDITIONAL BRANCH INST. AND   <====
6765                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6766                                                                ;           WHICH FOLLOWS W/ 753           <====
6767  021426                        ROL3:
6768  021426  012742  000507                MOV     #507,-(R2)      ;MOVE TO MAILBOX #  *******  507  *******
6769  021432  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6770  021434  000000                        HALT                    ;ROL DID NOT SET CC'S CORRECTLY
6771  021436  000277                ROL4:   SCC                     ;CC=0111
6772  021440  000250                        CLN
6773  021442  006100                        ROL     R0              ;CC=0000   R0=040001
6774  021444  101402                        BLOS    ROL5
6775  021446  102401                        BVS     ROL5
6776  021450  100004                        BPL     ROL6
6777                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6778                                                                ;           CONDITIONAL BRANCH INST. AND   <====
6779                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                ;           WHICH FOLLOWS W/ 741          <====
021452                          ROL5:
021452  012742  000510                MOV     #510,-(R2)      ;MOVE TO MAILBOX #  *******  510  *******
021456  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021460  000000                      HALT                    ;ROL DID NOT SET CC'S CORRECTLY
021462  000257                  ROL6: CCC                     ;CC=0101
021464  000265                      +SEZ!SEC
021466  006100                      ROL     R0              ;CC=1010   R0=100003
021470  101405                      BLOS    ROL7
021472  102004                      BVC     ROL7
021474  100003                      BPL     ROL7
021476  022700  100003              CMP     #100003,R0
021502  001404                      BEQ     TS225
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 724           <====
021504                          ROL7:
021504  012742  000511                MOV     #511,-(R2)      ;MOVE TO MAILBOX #  *******  511  *******
021510  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021512  000000                      HALT                    ;ROL MALFUNCTIONED
                                                ;  OR SEQUENCE ERROR
                                ;*******************************************************************************
                                ;TEST 225       TEST ROR INSTRUCTION
                                ;*******************************************************************************
021514  005212                  TS225: INC      (R2)            ;UPDATE TEST NUMBER
021516  022712  000225              CMP     #225,(R2)       ;SEQUENCE ERROR?
021522  001051                      BNE     TS226-10        ;BR TO ERROR HALT ON SEQ ERROR
021524  012700  000023              MOV     #23,R0          ;R0=23
021530  000277                      SCC                     ;CC=0111
021532  000250                      CLN
021534  006000                      ROR     R0              ;CC=1001   R0=100011
021536  102403                      BVS     ROR1
021540  103002                      BCC     ROR1
021542  001401                      BEQ     ROR1
021544  100404                      BMI     ROR2
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 766           <====
021546                          ROR1:
021546  012742  000512                MOV     #512,-(R2)      ;MOVE TO MAILBOX #  *******  512  *******
021552  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021554  000000                      HALT                    ;ROR DID NOT SET CC'S CORRECTLY
021556  000257                  ROR2: CCC                     ;CC=1100
021560  000274                      +SEN!SEZ
021562  006000                      ROR     R0              ;CC=0011   R0=040004
021564  102003                      BVC     ROR3
021566  103002                      BCC     ROR3
021570  001401                      BEQ     ROR3
021572  100004                      BPL     ROR4
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 753           <====
021574                          ROR3:
```

```
6836  021574  012742  000513          MOV     #513,-(R2)      ;MOVE TO MAILBOX #  *******  513  *******
6837  021600  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6838  021602  000000                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6839  021604  000277          ROR4:   SCC                     ;CC=1110
6840  021606  000241                  CLC
6841  021610  006000                  ROR     R0              ;CC=0000   R0=020002
6842  021612  101403                  BLOS    ROR5
6843  021614  102402                  BVS     ROR5
6844  021616  001401                  BEQ     ROR5
6845  021620  100004                  BPL     ROR6
6846                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6847                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6848                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6849                                                          ;           WHICH FOLLOWS W/ 740           <====
6850  021622                  ROR5:
6851  021622  012742  000514          MOV     #514,-(R2)      ;MOVE TO MAILBOX #  *******  514  *******
6852  021626  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6853  021630  000000                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6854  021632  000257          ROR6:   CCC                     ;CC=0101
6855  021634  000265                  +SEC!SEZ
6856  021636  006000                  ROR     R0              ;CC=1010   R0=110001
6857  021640  101402                  BLOS    ROR7
6858  021642  102001                  BVC     ROR7
6859  021644  100404                  BMI     TS226
6860                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6861                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6862                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6863                                                          ;           WHICH FOLLOWS W/ 726           <====
6864  021646                  ROR7:
6865  021646  012742  000515          MOV     #515,-(R2)      ;MOVE TO MAILBOX #  *******  515  *******
6866  021652  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6867  021654  000000                  HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
6868                                                          ;  OR SEQUENCE ERROR
6869  ;*****************************************************************************************
6870  ;TEST 226       TEST ASL INSTRUCTION
6871  ;*****************************************************************************************
6872  021656  005212          TS226:  INC     (R2)            ;UPDATE TEST NUMBER
6873  021660  022712  000226          CMP     #226,(R2)       ;SEQUENCE ERROR?
6874  021664  001054                  BNE     TS227-10        ;BR TO ERROR HALT ON SEQ ERROR
6875  021666  012700  144000          MOV     #144000,R0      ;R0=14000
6876  021672  000257                  CCC                     ;CC=0110
6877  021674  000271                  +SEN:SEC
6878  021676  006300                  ASL     R0              ;CC=1001   R0=110000
6879  021700  103003                  BCC     ASL1
6880  021702  102402                  BVS     ASL1
6881  021704  001401                  BEQ     ASL1
6882  021706  100404                  BMI     ASL2
6883                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6884                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6885                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6886                                                          ;           WHICH FOLLOWS W/ 766           <====
6887  021710                  ASL1:
6888  021710  012742  000516          MOV     #516,-(R2)      ;MOVE TO MAILBOX #  *******  516  *******
6889  021714  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6890  021716  000000                  HALT                    ;
6891  021720  000277          ASL2:   SCC                     ;CC=1100
```

```
6892  021722  000243                        *CLV!CLC
6893  021724  006300                        ASL     R0              ;CC-0011   R0=020000
6894  021726  103003                        BCC     ASL3
6895  021730  102002                        BVC     ASL3
6896  021732  001401                        BEQ     ASL3
6897  021734  100004                        BPL     ASL4
6898                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6899                                        ;           CONDITIONAL BRANCH INST. AND   <====
6900                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6901                                        ;           WHICH FOLLOWS W/ 753           <====
6902  021736                        ASL3:
6903  021736  012742  000517                MOV     #517,-(R2)      ;MOVE TO MAILBOX #  *******  517  *******
6904  021742  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6905  021744  000000                        HALT                    ;ASL DID NOT SET CC'S CORRECTLY
6906  021746  000277                ASL4:   SCC                     ;CC=0111
6907  021750  000250                        CLN
6908  021752  006300                        ASL     R0              ;CC=0000   R0=040000
6909  021754  101402                        BLOS    ASL5
6910  021756  102401                        BVS     ASL5
6911  021760  100004                        BPL     ASL6
6912                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6913                                        ;           CONDITIONAL BRANCH INST. AND   <====
6914                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6915                                        ;           WHICH FOLLOWS W/ 741           <====
6916  021762                        ASL5:
6917  021762  012742  000520                MOV     #520,-(R2)      ;MOVE TO MAILBOX #  *******  520  *******
6918  021766  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6919  021770  000000                        HALT                    ;ASL DID NOT SET CC'S CORRECTLY
6920  021772  000257                ASL6:   CCC                     ;CC=0101
6921  021774  000265                        *SEZ!SEC
6922  021776  006300                        ASL     R0              ;CC=1010   R0=100000
6923  022000  103406                        BCS     ASL7
6924  022002  001405                        BEQ     ASL7
6925  022004  102004                        BVC     ASL7
6926  022006  100003                        BPL     ASL7
6927  022010  022700  100000                CMP     #100000,R0
6928  022014  001404                        BEQ     TS227
6929                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6930                                        ;           CONDITIONAL BRANCH INST. AND   <====
6931                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6932                                        ;           WHICH FOLLOWS W/ 723           <====
6933  022016                        ASL7:
6934  022016  012742  000521                MOV     #521,-(R2)      ;MOVE TO MAILBOX #  *******  521  *******
6935  022022  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6936  022024  000000                        HALT                    ;ASL MALFUNCTIONED
6937                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;TEST 227       TEST ASR INSTRUCTION
                                ;*****************************************************************************************
022026  005212                  S227:   INC     (R2)            ;UPDATE TEST NUMBER
022030  022712  000227                  CMP     #227,(R2)       ;SEQUENCE ERROR?
022034  001060                          BNE     TS230-10        ;BR TO ERROR HALT ON SEQ ERROR
022036  012700  100023                  MOV     #100023,R0      ;R0=100023
022042  000277                          SCC                     ;CC=0110
022044  000250                          CLN
022046  006200                          ASR     R0              ;CC=1001   RP=140011
022050  102403                          BVS     ASR1
022052  103002                          BCC     ASR1
022054  001401                          BEQ     ASR1
022056  100404                          BMI     ASR2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
                                                                ;           CONDITIONAL BRANCH INST. AND   <-
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
                                                                ;           WHICH FOLLOWS W/ 766           <
022060                          ASR1:
022060  012742  000522                  MOV     #522,-(R2)      ;MOVE TO MAILBOX #  *******  522  *******
022064  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022066  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022070  042700  100000          ASR2:   BIC     #100000,R0      ;R0=40011
022074  000277                          SCC                     ;CC=1100
022076  000243                          +CLV!CLC
022100  006200                          ASR     R0              ;CC=0011   R0-020004
022102  102003                          BVC     ASR3
022104  103002                          BCC     ASR3
022106  001401                          BEQ     ASR3
022110  100004                          BPL     ASR4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <= ==
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 751           <====
022112                          ASR3:
022112  012742  000523                  MOV     #523,-(R2)      ;MOVE TO MAILBOX #  *******  523  *******
022116  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022120  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022122  000277                  ASR4:   SCC                     ;CC=1111
022124  006200                          ASR     R0              ;CC=0000   R0=010002
022126  101403                          BLOS    ASR5
022130  102402                          BVS     ASR5
022132  001401                          BEQ     ASR5
022134  100004                          BPL     ASR6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 737           <====
022136                          ASR5:
022136  012742  000524                  MOV     #524,-(R2)      ;MOVE TO MAILBOX #  *******  524  *******
022142  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022144  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022146  052700  100000          ASR6:   BIS     #100000,R0      ;R0=110002
022152  000257                          CCC                     ;CC=0101
022154  000265                          +SEZ.SEC
```

```
6994  022156  006200                  ASR     R0              ;C=1010   R0=144001
6995  022160  101406                  BLOS    ASR7
6996  022162  102005                  BVC     ASR7
6997  022164  100004                  BPL     ASR7
6998  022166  001403                  BEQ     ASR7
6999  022170  022700  144001          CMP     #144001,R0      ;CHECK RESULT OF ASR'S
7000  022174  001404                  BEQ     TS230
7001                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <---
7002                                          ;           CONDITIONAL BRANCH INST. AND   < -
7003                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7004                                          ;           WHICH FOLLOWS W/ 717           <=-=-
7005  022176                  ASR7:
7006  022176  012742  000525          MOV     #525,-(R2)      ;MOVE TO MAILBOX #  *******  525  *******
7007  022202  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7008  022204  000000                  HALT                    ;ASR DID NOT FUNCTION CORRECTLY
7009                                                          ;  OR SEQUENCE ERROR
7010
7011
7012
7013                    ;*****************************************************************************************
7014                    ;
7015                    ;       THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION CODES
7016                    ;ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
7017                    ;THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
7018                    ;CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
7019                    ;IS VERIFIED BY CONDITIONAL BRANCHES.
7020                    ;
7021                    ;*****************************************************************************************
7022                    ;TEST 230       TEST THE SXT INSTRUCTION
7023                    ;*****************************************************************************************
7024  022206  005212    TS230:  INC     (R2)            ;UPDATE TEST NUMBER
7025  022210  022712  000230          CMP     #230,(R2)       ;SEQUENCE ERROR?
7026  022214  001033                  BNE     TS231-10        ;BR TO ERROR HALT ON SEQ ERROR
7027  022216  005000                  CLR     R0
7028  022220  000277                  SCC                     ;SET CC=1011
7029  022222  000244                  CLZ
7030  022224  006700                  SXT     R0              ;TRY SXT
7031  022226  100006                  BPL     SXT0            ;TEST CC=1001
7032  022230  001405                  BEQ     SXT0
7033  022232  102404                  BVS     SXT0
7034  022234  103003                  BCC     SXT0
7035  022236  022700  177777          CMP     #-1,R0          ;CHECK DATA RESULT
7036  022242  001404                  BEQ     SXT1
7037                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < -
7038                                          ;           CONDITIONAL BRANCH INST. AND   < - -
7039                                          ;           REPLACE THE MOVE INSTRUCTION   <  -=
7040                                          ;           WHICH FOLLOWS W/ 764           <=  -
7041  022244                  SXT0:
7042  022244  012742  000526          MOV     #526,-(R2)      ;MOVE TO MAILBOX #  *******  526  *******
7043  022250  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7044  022252  000000                  HALT                    ;RESULTS OF SXT INCORRECT
7045  022254  005000    SXT1:   CLR     R0              ;R0=0
7046  022256  005010                  CLR     (R0)            ;LOC. 0=0
7047  022260  005110                  COM     (R0)            ;LOC. 0=177777
7048  022262  000257                  CCC                     ;SET CC=0110
7049  022264  000266                  +SEZ!SEV
```

```
 7050  022266  006710                        SXT     (R0)
 7051  022270  001005                        BNE     SXT2            ;TEST CC=0100
 7052  022272  103404                        BCS     SXT2
 7053  022274  102403                        BVS     SXT2
 7054  022276  100402                        BMI     SXT2
 7055  022300  005710                        TST     (R0)
 7056  022302  001404                        BEQ     TS231
 7057                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7058                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 7059                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 7060                                                  ;           WHICH FOLLOWS W/ 744           <====
 7061  022304                          SXT2:
 7062  022304  012742  000527                 MOV     #527,-(R2)      ;MOVE TO MAILBOX #  *******  527  *******
 7063  022310  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7064  022312  000000                        HALT                    ;RESULTS OF SXT INCORRECT
 7065                                                  ;  OR SEQUENCE ERROR
```

```
7066                                    ;*****************************************************************************
7067                                    ;
7068                                    ;       THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
7069                                    ;OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.
7070                                    ;AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
7071                                    ;EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
7072                                    ;REPRODUCE THE ORIGINAL VALUE IF R0=31525.
7073                                    ;
7074                                    ;*****************************************************************************
7075                                    ;TEST 231       TEST THE XOR INSTRUCTION
7076                                    ;*****************************************************************************
7077  022314  005212                    TS231:  INC     (R2)            ;UPDATE TEST NUMBER
7078  022316  022712  000231                    CMP     #231,(R2)       ;SEQUENCE ERROR?
7079  022322  001035                            BNE     TS232-10        ;BR TO ERROR HALT ON SEQ ERROR
7080  022324  012700  007463                    MOV     #7463,R0        ;SET UP R0
7081  022330  012701  031525                    MOV     #31525,R1       ;SET UP R1
7082  022334  000277                            SCC                     ;SET CC=1110
7083  022336  000241                            CLC
7084  022340  074100                            XOR     R1,R0           ;TRY XOR
7085  022342  101406                            BLOS    XOR1            ;CC=0000?
7086  022344  102405                            BVS     XOR1
7087  022346  001404                            BEQ     XOR1
7088  022350  100403                            BMI     XOR1
7089  022352  022700  036146                    CMP     #36146,R0       ;DATA RESULT CORRECT?
7090  022356  001404                            BEQ     XOR2
7091                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7092                                                                    ;           CONDITIONAL BRANCH INST. AND   <=- -
7093                                                                    ;           REPLACE THE MOVE INSTRUCTION   <==-=
7094                                                                    ;           WHICH FOLLOWS W/ 761           <=---=
7095  022360                            XOR1:
7096  022360  012742  000530                    MOV     #530,-(R2)      ;MOVE TO MAILBOX #  *******  530  *******
7097  022364  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7098  022366  000000                            HALT                    ;
7099  022370  010104                    XOR2:   MOV     R1,R4
7100  022372  000261                            SEC                     ;CC=1110
7101  022374  000241                            CLC
7102  022376  074400                            XOR     R4,R0           ;TRY XOR MODE 0,0
7103  022400  101406                            BLOS    XOR3            ;CC=0000?
7104  022402  102405                            BVS     XOR3
7105  022404  001404                            BEQ     XOR3
7106  022406  100403                            BMI     XOR3
7107  022410  022700  007463                    CMP     #7463,R0
7108  022414  001404                            BEQ     TS232
7109                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7110                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7111                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7112                                                                    ;           WHICH FOLLOWS W/ 742           <====
7113  022416                            XOR3:
7114  022416  012742  000531                    MOV     #531,-(R2)      ;MOVE TO MAILBOX #  *******  531  *******
7115  022422  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7116  022424  000000                            HALT                    ;RESULT OF XOR INCORRECT
7117                                                                    ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 232        TEST SOB INSTRUCTION
                                ;*****************************************************************************
022426  005212                TS232:  INC     (R2)            ;UPDATE TEST NUMBER
022430  022712  000232                CMP     #232,(R2)       ;SEQUENCE ERROR?
022434  001023                        BNE     TS233-10        ;BR TO ERROR HALT ON SEQ ERROR
022436  012700  000525                MOV     #525,R0
022442  010004                        MOV     R0,R4
022444  000277                        SCC                     ;SET CC=1111
022446  101002                SOB1:   BHI     SOB2            ;CC=1111?
022450  100001                        BPL     SOB2
022452  102404                        BVS     SOB3
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=
                                                        ;           CONDITIONAL BRANCH INST. AND   <=  =
                                                        ;           REPLACE THE MOVE INSTRUCTION   <= =-
                                                        ;           WHICH FOLLOWS W/ 770           <- -
022454                          SOB2:
022454  012742  000532                MOV     #532,-(R2)      ;MOVE TO MAILBOX #  *******  532  *******
022460  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022462  000000                        HALT                    ;
022464  005304                SOB3:   DEC     R4              ;COUNT ITERATIONS
022466  000277                        SCC                     ;CC=1111
022470  077012                        SOB     R0,SOB1         ;DO SOB W/ R0
022472  101004                        BHI     SOB4            ;CHECK CC-1111
022474  100003                        BPL     SOB4
022476  102002                        BVC     SOB4
022500  005704                        TST     R4              ;ITERATION COUNT OK?
022502  001404                        BEQ     TS233
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <===-
                                                        ;           WHICH FOLLOWS W/ 754           <===-
022504                          SOB4:
022504  012742  000533                MOV     #533,-(R2)      ;MOVE TO MAILBOX #  *******  533  *******
022510  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022512  000000                        HALT                    ;INCORRECT # OF BRANCHES OR CC'S CHANGED
                                                        ;  OR SEQUENCE ERROR
```

```
7162                                    ;*****************************************************************************
7163                                    ;
7164                                    ;       THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
7165                                    ;OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
7166                                    ;THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
7167                                    ;OF THE TWO ROUTINES IN THE TEST.
7168                                    ;
7169                                    ;*****************************************************************************
7170                                    ;TEST 233       TEST MARK INSTRUCTION
7171                                    ;*****************************************************************************
7172  022514  005212                    TS233:  INC     (R2)            ;UPDATE TEST NUMBER
7173  022516  022712  000233                    CMP     #233,(R2)       ;SEQUENCE ERROR?
7174  022522  001061                            BNE     TS234-10        ;BR TO ERROR HALT ON SEQ ERROR
7175  022524  012706  001000                    MOV     #STBOT,SP
7176  022530  012746  125252                    MOV     #125252,-(SP)   ;PUT R5 VALUE ON STACK
7177  022534  162706  000074                    SUB     #74,SP          ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
7178  022540  012705  022564                    MOV     #MRK1,R5        ;SET NEW PC IN R5
7179  022544  012746  006436                    MOV     #6436,-(SP)     ;PUT MARK 36 INST. ON STACK
7180  022550  000277                            SCC                     ;SET CC=1111
7181  022552  000116                            JMP     (SP)            ;XFER CONTL TO MARK 36 INST. ON STACK
7182  022554  012742  000534                    MOV     #534,-(R2)      ;MOVE TO MAILBOX #  *******  534  *******
7183  022560  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7184  022562  000000                            HALT                    ;MARK INST. SHOULD HAVE JUMPED TO MRK1
7185  022564  101010                    MRK1:   BHI     MRK2            ;TEST CC UNAFFECTED
7186  022566  100007                            BPL     MRK2            ;IE. CC=1111
7187  022570  102006                            BVC     MRK2
7188  022572  020527  125252                    CMP     R5,#125252      ;CHECK R5 RESTORED FROM STACK
7189  022576  001003                            BNE     MRK2
7190  022600  022706  001000                    CMP     #STBOT,R6       ;CHECK STACK POINTER READJUSTED CORRECTLY.
7191  022604  001404                            BEQ     MRK3
7192                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7193                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7194                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7195                                                                    ;           WHICH FOLLOWS W/ 746           <====
7196  022606                            MRK2:
7197  022606  012742  000535                    MOV     #535,-(R2)      ;MOVE TO MAILBOX #  *******  535  *******
7198  022612  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7199  022614  000000                            HALT                    ;RESULTS OF MARK INCORRECT
7200  022616  012746  052525            MRK3:   MOV     #52525,-(SP)
7201  022622  012746  006400                    MOV     #6400,-(SP)     ;PUT MARK 0 INST. ON STACK
7202  022626  010605                            MOV     SP,R5           ;SET ADDR. OF MARK INST. IN R5
7203  022630  004737  022640                    JSR     PC,@#MRK4       ;DO JSR
7204  022634  000137  022652                    JMP     @#MRK5          ;
7205  022640  000205                    MRK4:   RTS     R5              ;DO RTS WITH R5 TO MARK INST ON STACK
7206  022642  012742  000536                    MOV     #536,-(R2)      ;MOVE TO MAILBOX #  *******  536  *******
7207  022646  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7208  022650  000000                            HALT                    ;RTS,MARK SEQUENCE FAILED
7209  022652  022706  001000            MRK5:   CMP     #STBOT,R6       ;STACK ADJUSTED CORRECTLY
7210  022656  001003                            BNE     MRK6            ;IF NOT:  BR
7211  022660  022705  052525                    CMP     #52525,R5       ;CHECK IF R5 RESTORED FROM STACK
7212  022664  001404                            BEQ     TS234
7213                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=---
7214                                                                    ;           CONDITIONAL BRANCH INST. AND   <=  =
7215                                                                    ;           REPLACE THE MOVE INSTRUCTION   <-=--
7216                                                                    ;           WHICH FOLLOWS W/ 716           <-
7217  022666                            MRK6:
```

```
 7218  022666  012742  000537                MOV     #537,-(R2)      ;MOVE TO MAILBOX #  *******  537  *******
 7219  022672  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7220  022674  000000                        HALT                    ;RESULTS OF MARK INCORRECT
 7221                                                                  ;  OR SEQUENCE ERROR
```

```
                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 234       TEST MTPS INSTRUCTION
                                        ;*****************************************************************************
022676  005212                  TS234:  INC     (R2)            ;UPDATE TEST NUMBER
022700  022712  000234                  CMP     #234,(R2)       ;SEQUENCE ERROR?
022704  001024                          BNE     TS235-10        ;BR TO ERROR HALT ON SEQ ERROR
022706  012700  000377                  MOV     #377,R0
022712  000257                          CCC
022714  106400                          MTPS    R0
022716  022767  000357  155052          CMP     #357,PS
022724  001404                          BEQ     MTPS1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
022726  012742  000540                  MOV     #540,-(R2)      ;MOVE TO MAILBOX #  *******  540  *******
022732  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022734  000000                          HALT                    ;MTPS FAILED
022736  005000                  MTPS1:  CLR     R0
022740  005010                          CLR     (R0)
022742  000277                          SCC                     ;CC=1111
022744  106410                          MTPS    (R0)            ;TRY MTPS MODE 1
022746  100403                          BMI     MTPS1A          ;CHECK PS
022750  102402                          BVS     MTPS1A
022752  103401                          BCS     MTPS1A
022754  001004                          BNE     TS235
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 753           <====
022756                  MTPS1A:
022756  012742  000541                  MOV     #541,-(R2)      ;MOVE TO MAILBOX #  *******  541  *******
022762  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022764  000000                          HALT                    ;MTPS FAILED
                                                                ;  OR SEQUENCE ERROR
                                        ;*****************************************************************************
                                        ;TEST 235       TEST MTPS MODE 2
                                        ;*****************************************************************************
022766  005212                  TS235:  INC     (R2)            ;UPDATE TEST NUMBER
022770  022712  000235                  CMP     #235,(R2)       ;SEQUENCE ERROR?
022774  001021                          BNE     TS236-10        ;BR TO ERROR HALT ON SEQ ERROR
022776  005000                          CLR     R0              ;R0=0
023000  012710  177777                  MOV     #-1,(R0)        ;LOC. 0=-1
023004  005037  177776                  CLR     @#PS            ;PS=0
023010  106420                          MTPS    (R0)+           ;TRY MTPS W/MODE 2
```

```
7278  023012  022737  000357  177776          CMP     #357,@#PS       ;CHECK DATA
7279  023020  001404                          BEQ     MTPS2
7280                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7281                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7282                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7283                                                                  ;           WHICH FOLLOWS W/ 765           <====
7284  023022  012742  000542                MOV     #542,-(R2)      ;MOVE TO MAILBOX #  *******  542  *******
7285  023026  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7286  023030  000000                          HALT                    ;DEST. DATA INCORRECT
7287  023032  022700  000001          MTPS2:  CMP     #1,R0           ;CHECK DEST. REGISTER.
7288  023036  001404                          BEQ     TS236
7289                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7290                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7291                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7292                                                                  ;           WHICH FOLLOWS W/ 756           <====
7293  023040  012742  000543                MOV     #543,-(R2)      ;MOVE TO MAILBOX #  *******  543  *******
7294  023044  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7295  023046  000000                          HALT                    ;DEST REGISTER NOT INCREMENTED BY 1
7296                                                                  ;  OR SEQUENCE ERROR
7297
7298                          ;*****************************************************************************************
7299                          ;TEST 236       TEST MTPS MODE 3
7300                          ;*****************************************************************************************
7301  023050  005212          TS236:  INC     (R2)            ;UPDATE TEST NUMBER
7302  023052  022712  000236          CMP     #236,(R2)       ;SEQUENCE ERROR?
7303  023056  001024                  BNE     TS237-10        ;BR TO ERROR HALT ON SEQ ERROR
7304  023060  012700  000402          MOV     #402,R0         ;R0=402
7305  023064  005010                  CLR     (R0)            ;LOC. 402=0
7306  023066  012737  052652  000000  MOV     #52652,@#0      ;LOC. 0=52652
7307  023074  005037  177776          CLR     @#PS            ;PS=0
7308  023100  106430                  MTPS    @(R0)+          ;TRY MTPS W/MODE 3
7309  023102  022737  000252  177776  CMP     #252,@#PS       ;CHECK DEST. DATA
7310  023110  001404                  BEQ     MTPS3
7311                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7312                                                          ;           CONDITIONAL BRANCH INST. AND   <====
7313                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7314                                                          ;           WHICH FOLLOWS W/ 762           <====
7315  023112  012742  000544          MOV     #544,-(R2)      ;MOVE TO MAILBOX #  *******  544  *******
7316  023116  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7317  023120  000000                  HALT                    ;DEST. DATA INCORRECT
7318  023122  022700  000404  MTPS3:  CMP     #404,R0         ;CHECK MODE 3 REGISTER.
7319  023126  001404                  BEQ     TS237
7320                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7321                                                          ;           CONDITIONAL BRANCH INST. AND   <====
7322                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7323                                                          ;           WHICH FOLLOWS W/ 753           <====
7324  023130  012742  000545          MOV     #545,-(R2)      ;MOVE TO MAILBOX #  *******  545  *******
7325  023134  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7326  023136  000000                  HALT                    ;MODE 3 REGISTER INCORRECT
7327                                                          ;  OR SEQUENCE ERROR
7328
7329                          ;*****************************************************************************************
7330                          ;TEST 237       TEST MTPS MODE 4
7331                          ;*****************************************************************************************
7332  023140  005212          TS237:  INC     (R2)            ;UPDATE TEST NUMBER
7333  023142  022712  000237          CMP     #237,(R2)       ;SEQUENCE ERROR?
```

```
7334  023146  001022                          BNE     TS240-10        ;BR TO ERROR HALT ON SEQ ERROR
7335  023150  012700  000001                  MOV     #1,R0           ;R0=1
7336  023154  012737  125125  000000          MOV     #125125,@#0     ;LOC. 0 = 125125
7337  023162  005037  177776                  CLR     @#PS            ;PS=0
7338  023166  106440                          MTPS    -(R0)           ;TRY MTPS W/MODE 4
7339  023170  022737  000105  177776          CMP     #105,@#PS       ;CHECK DEST. DATA
7340  023176  001404                          BEQ     MTPS4
7341                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7342                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7343                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7344                                                                  ;           WHICH FOLLOWS W/ 763           <====
7345  02320C  012742  000546                  MOV     #546,-(R2)      ;MOVE TO MAILBOX #  *******  546  *******
7346  023204  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7347  023206  000000                          HALT                    ;DEST. DATA INCORRECT
7348  023210  005700                  MTPS4:  TST     R0              ;CHECK MODE 4 REGISTER
7349  023212  001404                          BEQ     TS240
7350                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7351                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7352                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7353                                                                  ;           WHICH FOLLOWS W/ 755           <====
7354  023214  012742  000547                  MOV     #547,-(R2)      ;MOVE TO MAILBOX #  *******  547  *******
7355  023220  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7356  023222  000000                          HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
7357                                                                  ;  OR SEQUENCE ERROR
7358
7359                          ;*****************************************************************************
7360                          ;TEST 240       TEST MTPS MODE 5
7361                          ;*****************************************************************************
7362  023224  005212          TS240:  INC     (R2)            ;UPDATE TEST NUMBER
7363  023226  022712  000240          CMP     #240,(R2)       ;SEQUENCE ERROR?
7364  023232  001021                  BNE     TS241-10        ;BR TO ERROR HALT ON SEQ ERROR
7365  023234  012700  000404          MOV     #404,R0         ;R0=404
7366  023240  012737  177400  000000  MOV     #177400,@#0     ;LOC. 0=177400
7367  023246  000277                  SCC                     ;SET ALL COND. CODES
7368  023250  106450                  MTPS    @-(R0)          ;TRY MTPS W/MODE 5
7369  023252  005737  177776          TST     @#PS            ;CHECK DEST. DATA.
7370  023256  001404                  BEQ     MTPS5
7371                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
7372                                                          ;           CONDITIONAL BRANCH INST. AND   <----
7373                                                          ;           REPLACE THE MOVE INSTRUCTION   <----
7374                                                          ;           WHICH FOLLOWS W/ 765           <----
7375  023260  012742  000550          MOV     #550,-(R2)      ;MOVE TO MAILBOX #  *******  550  *******
7376  023264  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7377  023266  000000                  HALT                    ;DESTINATION DATA INCORRECT
7378  023270  022700  000402  MTPS5:  CMP     #402,R0         ;CHECK MODE 5 REGISTER
7379  023274  001404                  BEQ     TS241
7380                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
7381                                                          ;           CONDITIONAL BRANCH INST. AND   <----
7382                                                          ;           REPLACE THE MOVE INSTRUCTION   <----
7383                                                          ;           WHICH FOLLOWS W/ 756           <----
7384  023276  012742  000551          MOV     #551,-(R2)      ;MOVE TO MAILBOX #  *******  551  *******
7385  023302  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7386  023304  000000                  HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
7387                                                          ;  OR SEQUENCE ERROR
7388
7389                          ;*****************************************************************************
```

```
7390                                        ;TEST 241       TEST MTPS MODE 6
7391                                        ;*****************************************************************************
7392  023306  005212                        TS241:  INC     (R2)            ;UPDATE TEST NUMBER
7393  023310  022712  000241                        CMP     #241,(R2)       ;SEQUENCE ERROR?
7394  023314  001024                                BNE     TS242-10        ;BR TO ERROR HALT ON SEQ ERROR
7395  023316  012737  052652  000000                MOV     #52652,@#0      ;LOC. 0=52652
7396  023324  012700  000406                        MOV     #406,R0         ;R0=406
7397  023330  005037  177776                        CLR     @#PS            ;PS=0
7398  023334  106460  177372                        MTPS    -406(R0)        ;TRY MTPS W/MODE 6
7399  023340  022737  000252  177776                CMP     #252,@#PS       ;CHECK DEST. DATA
7400  023346  001404                                BEQ     MTPS6
7401                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7402                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
7403                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
7404                                                                        ;           WHICH FOLLOWS W/ 762           <====
7405  023350  012742  000552                        MOV     #552,-(R2)      ;MOVE TO MAILBOX #  *******  552  *******
7406  023354  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7407  023356  000000                                HALT                    ;DEST. DATA INCORRECT
7408  023360  022700  000406                MTPS6:  CMP     #406,R0         ;CHECK MODE 6 REGISTER
7409  023364  001404                                BEQ     TS242
7410                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7411                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
7412                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
7413                                                                        ;           WHICH FOLLOWS W/ 753           <====
7414  023366  012742  000553                        MOV     #553,-(R2)      ;MOVE TO MAILBOX #  *******  553  *******
7415  023372  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7416  023374  000000                                HALT                    ;MODE 6 REGISTER MODIFIED
7417                                                                        ;  OR SEQUENCE ERROR
7418
7419                                        ;*****************************************************************************
7420                                        ;TEST 242       TEST MTPS MODE 7
7421                                        ;*****************************************************************************
7422  023376  005212                        TS242:  INC     (R2)            ;UPDATE TEST NUMBER
7423  023400  022712  000242                        CMP     #242,(R2)       ;SEQUENCE ERROR?
7424  023404  001024                                BNE     TS243-10        ;BR TO ERROR HALT ON SEQ ERROR
7425  023406  012737  052652  000000                MOV     #52652,@#0      ;LOC. 0=52652
7426  023414  012700  000410                        MOV     #410,R0         ;R0=410
7427  023420  005037  177776                        CLR     @#PS            ;PS-0
7428  023424  106470  177776                        MTPS    @-2(R0)         ;TRY MTPS W/MODE 7
7429  023430  022737  000105  177776                CMP     #105,@#PS       ;CHECK DEST. DATA
7430  023436  001404                                BEQ     MTPS7
7431                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7432                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
7433                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
7434                                                                        ;           WHICH FOLLOWS W/ 762           <====
7435  023440  012742  000554                        MOV     #554,-(R2)      ;MOVE TO MAILBOX #  *******  554  *******
7436  023444  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7437  023446  000000                                HALT                    ;DESTINATION DATA INCORRECT
7438  023450  022700  000410                MTPS7:  CMP     #410,R0         ;CHECK MODE 7 REGISTER
7439  023454  001404                                BEQ     TS243
7440                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7441                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
7442                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
7443                                                                        ;           WHICH FOLLOWS W/ 753           <====
7444  023456  012742  000555                        MOV     #555,-(R2)      ;MOVE TO MAILBOX #  *******  555  *******
7445  023462  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
7446  023464  000000                    HALT                    ;MODE 7 REGISTER MODIFIED
7447                                                            ;  OR SEQUENCE ERROR
7448
```

```
7449                                    ;*****************************************************************************
7450                                    ;
7451                                    ;       THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
7452                                    ;MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES IS MOVED TO THE
7453                                    ;PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
7454                                    ;BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
7455                                    ;USED TO CHECK PROPER ADDRESSING AND DATA.
7456                                    ;
7457                                    ;*****************************************************************************
7458                                    ;TEST 243       TEST MFPS INSTRUCTION
7459                                    ;*****************************************************************************
7460  023466  005212                    TS243:  INC     (R2)            ;UPDATE TEST NUMBER
7461  023470  022712  000243                    CMP     #243,(R2)       ;SEQUENCE ERROR?
7462  023474  001025                            BNE     TS244-10        ;BR TO ERROR HALT ON SEQ ERROR
7463  023476  012737  000377  177776            MOV     #377,@#PS
7464  023504  106700                            MFPS    R0
7465  023506  022700  177757                    CMP     #177757,R0
7466  023512  001404                            BEQ     MFPS1
7467                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
7468                                                                    ;           CONDITIONAL BRANCH INST. AND   <--
7469                                                                    ;           REPLACE THE MOVE INSTRUCTION   <--
7470                                                                    ;           WHICH FOLLOWS W/ 770           <--
7471  023514  012742  000556                    MOV     #556,-(R2)      ;MOVE TO MAILBOX #  *******  556  *******
7472  023520  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7473  023522  000000                            HALT                    ;MFPS FAILED
7474
7475  023524  005000                    MFPS1:  CLR     R0
7476  023526  012737  177777  000000            MOV     #-1,@#0
7477  023534  005037  177776                    CLR     @#PS
7478  023540  106710                            MFPS    (R0)
7479  023542  105737  000000                    TSTB    @#0
7480  023546  001404                            BEQ     TS244
7481                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7482                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7483                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7484                                                                    ;           WHICH FOLLOWS W/ 752           <====
7485  023550  012742  000557                    MOV     #557,-(R2)      ;MOVE TO MAILBOX #  *******  557  *******
7486  023554  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7487  023556  000000                            HALT                    ;MFPS FAILED
7488                                                                    ;  OR SEQUENCE ERROR
7489
7490                                    ;*****************************************************************************
7491                                    ;TEST 244       TEST MFPS MODE 2
7492                                    ;*****************************************************************************
7493  023560  005212                    TS244:  INC     (R2)            ;UPDATE TEST NUMBER
7494  023562  022712  000244                    CMP     #244,(R2)       ;SEQUENCE ERROR?
7495  023566  001031                            BNE     TS245-10        ;BR TO ERROR HALT ON SEQ ERROR
7496  023570  005000                            CLR     R0              ;R0=0
7497  023572  005010                            CLR     (R0)            ;LOC. 0=0
7498  023574  012737  000377  177776            MOV     #377,@#PS       ;SET PS=357
7499  023602  106720                            MFPS    (R0)+           ;TRY MFPS W/MODE 2
7500  023604  103003                            BCC     MFPS2A          ;BR TO ERROR IF C BIT CLEAR
7501  023606  102402                            BVS     MFPS2A          ;BR TO ERROR IF V BIT SET
7502  023610  001401                            BEQ     MFPS2A          ;BR TO ERROR IF Z BIT SET
7503  023612  100404                            BMI     MFPS2B
7504                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=
```

```
7505                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7506                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7507                                                    ;           WHICH FOLLOWS W/ 765           <====
7508  023614                            MFPS2A:
7509  023614  012742  000560                   MOV     #560,-(R2)      ;MOVE TO MAILBOX #  *******  560  *******
7510  023620  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7511  023622  000000                     HALT                    ;COND. CODES INCORRECT
7512  023624  022737  000357  000000 MFPS2B: CMP     #357,@#0        ;CHECK DEST. DATA
7513  023632  001404                     BEQ     MFPS2C
7514                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7515                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7516                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7517                                                    ;           WHICH FOLLOWS W/ 755           <====
7518  023634  012742  000561                   MOV     #561,-(R2)      ;MOVE TO MAILBOX #  *******  561  *******
7519  023640  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7520  023642  000000                     HALT                    ;DEST. DATA INCORRECT
7521  023644  022700  000001 MFPS2C: CMP     #1,R0           ;CHECK MODE 2 REGISTER
7522  023650  001404                     BEQ     TS245
7523                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7524                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7525                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7526                                                    ;           WHICH FOLLOWS W/ 746           <====
7527  023652  012742  000562                   MOV     #562,-(R2)      ;MOVE TO MAILBOX #  *******  562  *******
7528  023656  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7529  023660  000000                     HALT                    ;MODE 2 REGISTER NOT INCREMENTED 1
7530                                                    ;  OR SEQUENCE ERROR
7531
7532                                    ;*******************************************************************************
7533                                    ;TEST 245       TEST MFPS MODE 3
7534                                    ;*******************************************************************************
7535  023662  005212                TS245:  INC     (R2)            ;UPDATE TEST NUMBER
7536  023664  022712  000245                   CMP     #245,(R2)       ;SEQUENCE ERROR?
7537  023670  001033                     BNE     TS246-10        ;BR TO ERROR HALT ON SEQ ERROR
7538  023672  012700  000406                   MOV     #406,R0         ;R0=406
7539  023676  005037  000000                   CLR     @#0             ;LOC. 0=0
7540  023702  012737  000252  177776           MOV     #252,@#PS       ;PS=252
7541  023710  106730                     MFPS    @(R0)+          ;TRY MFPS WITH MODE 3
7542  023712  103403                     BCS     MFPS3A          ;BR TO ERROR IF C-BIT SET
7543  023714  102402                     BVS     MFPS3A          ;BR TO ERROR IF V-BIT SET
7544  023716  001401                     BEQ     MFPS3A          ;BR TO ERROR IF Z-BIT SET
7545  023720  100404                     BMI     MFPS3B
7546                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- -
7547                                                    ;           CONDITIONAL BRANCH INST. AND   <   -
7548                                                    ;           REPLACE THE MOVE INSTRUCTION   <=
7549                                                    ;           WHICH FOLLOWS W/ 763           <
7550  023722                            MFPS3A:
7551  023722  012742  000563                   MOV     #563,-(R2)      ;MOVE TO MAILBOX #  *******  563  *******
7552  023726  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7553  023730  000C00                     HALT                    ;CONDITION CODES INCORRECT
7554  023732  022737  125000  000000 MFPS3B: CMP     #125000,@#0     ;CHECK DEST. DATA
7555  023740  001404                     BEQ     MFPS3C
7556                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=
7557                                                    ;           CONDITIONAL BRANCH INST. AND   <=-=-=
7558                                                    ;           REPLACE THE MOVE INSTRUCTION   <-= -
7559                                                    ;           WHICH FOLLOWS W/ 753           <--
7560  023742  012742  000564                   MOV     #564,-(R2)      ;MOVE TO MAILBOX #  *******  564  *******
```

```
7561  023746  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7562  023750  000000                        HALT                    ;DEST DATA INCORRECT
7563  023752  020027  000410        MFPS3C: CMP     R0,#410         ;CHECK MODE 3 REGISTER.
7564  023756  001404                        BEQ     TS246
7565                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===
7566                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7567                                                                ;           REPLACE THE MOVE INSTRUCTION   <=====
7568                                                                ;           WHICH FOLLOWS W/ 744           <====
7569  023760  012742  000565                MOV     #565,-(R2)      ;MOVE TO MAILBOX #  *******  565  *******
7570  023764  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7571  023766  000000                        HALT                    ;MODE 3 REGISTER NOT INCREMENTED BY 2
7572                                                                ;  OR SEQUENCE ERROR
7573
7574                                ;*****************************************************************************
7575                                ;TEST 246       TEST MFPS MODE 4
7576                                ;*****************************************************************************
7577  023770  005212                TS246:  INC     (R2)            ;UPDATE TEST NUMBER
7578  023772  022712  000246                CMP     #246,(R2)       ;SEQUENCE ERROR?
7579  023776  001033                        BNE     TS247-10        ;BR TO ERROR HALT ON SEQ ERROR
7580  024000  012700  000002                MOV     #2,R0           ;R0=2
7581  024004  005037  000000                CLR     @#0             ;LOC. 0=0
7582  024010  012737  000125  177776        MOV     #125,@#PS       ;PS=125
7583  024016  106740                        MFPS    -(R0)           ;TRY MFPS W/MODE 4
7584  024020  103003                        BCC     MFPS4A          ;BR TO ERROR IF C-BIT CLEAR
7585  024022  102402                        BVS     MFPS4A          ;BR TO ERROR IF V-BIT SET
7586  024024  001401                        BEQ     MFPS4A          ;BR TO ERROR IF Z-BIT SET
7587  024026  100004                        BPL     MFPS4B
7588                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7589                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7590                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
7591                                                                ;           WHICH FOLLOWS W/ 763           <====
7592  024030                        MFPS4A:
7593  024030  012742  000566                MOV     #566,-(R2)      ;MOVE TO MAILBOX #  *******  566  *******
7594  024034  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7595  024036  000000                        HALT                    ;COND. CODES INCORRECT
7596  024040  022737  042400  000000  MFPS4B: CMP     #42400,@#0      ;CHECK DEST. DATA
7597  024046  001404                        BEQ     MFPS4C
7598                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7599                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7600                                                                ;           REPLACE THE MOVE INSTRUCTION   <-=-
7601                                                                ;           WHICH FOLLOWS W/ 753           <=-==
7602  024050  012742  000567                MOV     #567,-(R2)      ;MOVE TO MAILBOX #  *******  567  *******
7603  024054  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7604  024056  000000                        HALT                    ;DEST. DATA INCORRECT
7605  024060  020027  000001        MFPS4C: CMP     R0,#1           ;CHECK MODE 4 REGISTER
7606  024064  001404                        BEQ     TS247
7607                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=
7608                                                                ;           CONDITIONAL BRANCH INST. AND   <-
7609                                                                ;           REPLACE THE MOVE INSTRUCTION   <
7610                                                                ;           WHICH FOLLOWS W/ 744           <  -
7611  024066  012742  000570                MOV     #570,-(R2)      ;MOVE TO MAILBOX #  *******  570  *******
7612  024072  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7613  024074  000000                        HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
7614                                                                ;  OR SEQUENCE ERROR
7615
7616                                ;*****************************************************************************
```

```
                                        ;TEST 247       TEST MFPS MODE 5
                                        ;*****************************************************************************************
024076  005212                  TS247:  INC     (R2)            ;UPDATE TEST NUMBER
024100  022712  000247                  CMP     #247,(R2)       ;SEQUENCE ERROR?
024104  001033                          BNE     TS250-10        ;BR TO ERROR HALT ON SEQ ERROR
024106  012700  000410                  MOV     #410,R0         ;R0=410
024112  012737  177777  000000          MOV     #-1,@#0         ;LOC. 0=-1
024120  005037  177776                  CLR     @#PS            ;PS=0
024124  106750                          MFPS    @-(R0)          ;TRY MFPS W/MODE 5
024126  103403                          BCS     MFPS5A          ;BR TO ERROR IF C-BIT SET
024130  102402                          BVS     MFPS5A          ;BR TO ERROR IF V-BIT SET
024132  100401                          BMI     MFPS5A          ;BR TO ERROR IF N-BIT SET
024134  001404                          BEQ     MFPS5B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 763           <====
024136                  MFPS5A:
024136  012742  000571                  MOV     #571,-(R2)      ;MOVE TO MAILBOX #  *******  571  *******
024142  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024144  000000                          HALT                    ;COND. CODES INCORRECT
024146  022737  000377  000000  MFPS5B: CMP     #377,@#0        ;CHECK DEST. DATA
024154  001404                          BEQ     MFPS5C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 753           <====
024156  012742  000572                  MOV     #572,-(R2)      ;MOVE TO MAILBOX #  *******  572  *******
024162  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024164  000000                          HALT                    ;DEST DATA INCORRECT
024166  020027  000406          MFPS5C: CMP     R0,#406         ;CHECK MODE 5 REGISTER
024172  001404                          BEQ     TS250
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 744           <====
024174  012742  000573                  MOV     #573,-(R2)      ;MOVE TO MAILBOX #  *******  573  *******
024200  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024202  000000                          HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
                                                                ;  OR SEQUENCE ERROR

                                        ;*****************************************************************************************
                                        ;TEST 250       TEST MFPS MODE 6
                                        ;*****************************************************************************************
024204  005212                  TS250:  INC     (R2)            ;UPDATE TEST NUMBER
024206  022712  000250                  CMP     #250,(R2)       ;SEQUENCE ERROR?
024212  001034                          BNE     TS251-10        ;BR TO ERROR HALT ON SEQ ERROR
024214  012700  000401                  MOV     #401,R0         ;R0=410
024220  005037  000000                  CLR     @#0             ;LOC. 0=0
024224  012737  000252  177776          MOV     #252,@#PS       ;PS=252
024232  106760  177377                  MFPS    -401(R0)        ;TRY MFPS W/MODE 6
024236  102403                          BVS     MFPS6A          ;BR TO ERROR IF V-BIT SET
024240  103402                          BCS     MFPS6A          ;BR TO ERROR IF C-BIT SET
024242  001401                          BEQ     MFPS6A          ;BR TO ERROR IF Z-BIT SET
024244  100404                          BMI     MFPS6B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
7673                                                    ;           CONDITIONAL BRANCH INST. AND   <=-=-
7674                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7675                                                    ;           WHICH FOLLOWS W/ 762           <==-=
7676  024246                            MFPS6A:
7677  024246  012742  000574                  MOV     #574,-(R2)      ;MOVE TO MAILBOX #  *******  574  *******
7678  024252  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7679  024254  000000                    HALT                    ;COND. CODES INCORRECT
7680  024256  022737  000252  000000    MFPS6B: CMP     #252,@#0        ;CHECK DEST. DATA
7681  024264  001404                    BEQ     MFPS6C
7682                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < -=-
7683                                                    ;           CONDITIONAL BRANCH INST. AND   <- --
7684                                                    ;           REPLACE THE MOVE INSTRUCTION   < ---
7685                                                    ;           WHICH FOLLOWS W/ 752           < = -
7686  024266  012742  000575                  MOV     #575,-(R2)      ;MOVE TO MAILBOX #  *******  575  *******
7687  024272  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7688  024274  000000                    HALT                    ;DEST. DATA INCORRECT
7689  024276  022700  000401    MFPS6C: CMP     #401,R0         ;CHECK DEST. REGISTER
7690  024302  001404                    BEQ     TS251
7691                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=-
7692                                                    ;           CONDITIONAL BRANCH INST. AND   <=-==
7693                                                    ;           REPLACE THE MOVE INSTRUCTION   <-==
7694                                                    ;           WHICH FOLLOWS W/ 743           <====
7695  024304  012742  000576                  MOV     #576,-(R2)      ;MOVE TO MAILBOX #  *******  576  *******
7696  024310  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7697  024312  000000                    HALT                    ;DEST. DATA INCORRECT
7698                                                    ;  OR SEQUENCE ERROR
7699
7700                                    ;*****************************************************************************
7701                                    ;TEST 251       TEST MFPS MODE 7                                          .
7702                                    ;*****************************************************************************
7703  024314  005212                    TS251:  INC     (R2)            ;UPDATE TEST NUMBER
7704  024316  022712  000251                    CMP     #251,(R2)       ;SEQUENCE ERROR?
7705  024322  001034                    BNE     TS252-10        ;BR TO ERROR HALT ON SEQ ERROR
7706  024324  012700  000777                    MOV     #777,R0         ;R0=777
7707  024330  005037  000000                    CLR     @#0             ;LOC. 0=0
7708  024334  012737  000125  177776          MOV     #125,@#PS       ;PS=125
7709  024342  106770  177407                  MFPS    @-371(R0)       ;TRY MFPS W/MODE 7
7710  024346  102403                    BVS     MFPS7A          ;BR TO ERROR IF V-BIT SET
7711  024350  103002                    BCC     MFPS7A          ;BR TO ERROR IF C-BIT SET
7712  024352  001401                    BEQ     MFPS7A          ;BR TO ERROR IF Z-BIT SET
7713  024354  100004                    BPL     MFPS7B
7714                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7715                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7716                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7717                                                    ;           WHICH FOLLOWS W/ 762           <====
7718  024356                            MFPS7A:
7719  024356  012742  000577                  MOV     #577,-(R2)      ;MOVE TO MAILBOX #  *******  577  *******
7720  024362  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7721  024364  000000                    HALT                    ;CONDITION CODE INCORRECT
7722  024366  022737  042400  000000    MFPS7B: CMP     #42400,@#0      ;CHECK DESTINATION DATA
7723  024374  001404                    BEQ     MFPS7C
7724                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7725                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7726                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7727                                                    ;           WHICH FOLLOWS W/ 752           <====
7728  024376  012742  000600                  MOV     #600,-(R2)      ;MOVE TO MAILBOX #  *******  600  *******
```

```
7729  024402  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7730  024404  000000                          HALT                    ;DEST. DATA INCORRECT
7731  024406  022700  000777        MFPS7C: CMP     #777,R0         ;CHECK MODE 7 REGISTER
7732  024412  001404                          BEQ     TS252
7733                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
7734                                                                  ;           CONDITIONAL BRANCH INST. AND   <  --
7735                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-=-=
7736                                                                  ;           WHICH FOLLOWS W/ 743           <  =--
7737  024414  012742  000601                  MOV     #601,-(R2)      ;MOVE TO MAILBOX #  *******  601  *******
7738  024420  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7739  024422  000000                          HALT                    ;MODE 7 REGISTER MODIFIED
7740                                                                  ;  OR SEQUENCE ERROR
7741
7742                          ;*****************************************************************************
7743                          ;
7744                          ;       THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
7745                          ;THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
7746                          ;CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
7747                          ;CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 240 (DECIMAL)
7748                          ;ITERATIONS OF PROGRAM.
7749                          ;
7750                          ;*****************************************************************************
7751                          ;TEST 252       TEST THAT RESET DOES NOT CLEAR PSW
7752                          ;*****************************************************************************
7753  024424  005212          TS252:  INC     (R2)            ;UPDATE TEST NUMBER
7754  024426  022712  000252          CMP     #252,(R2)       ;SEQUENCE ERROR?
7755  024432  001010                  BNE     TS253-10        ;BR TO ERROR HALT ON SEQ ERROR
7756  024434  012737  000357  177776          MOV     #357,@#PS               ;MOV ONES TO PSW
7757  024442  000005                          RESET                           ;
7758  024444  022737  000357  177776          CMP     #357,@#PS               ;PSW CORRECT?
7759  024452  001404                          BEQ     TS253
7760                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-= -
7761                                                                  ;           CONDITIONAL BRANCH INST. AND   <-= =
7762                                                                  ;           REPLACE THE MOVE INSTRUCTION   <==-
7763                                                                  ;           WHICH FOLLOWS W/ 767           <  -=
7764  024454  012742  000602                  MOV     #602,-(R2)      ;MOVE TO MAILBOX #  *******  602  *******
7765  024460  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7766  024462  000000                          HALT                    ;RESET ALTERED PSW
7767                                                                  ;  OR SEQUENCE ERROR
7768  024464                          REST:
7769
7770                          ;*****************************************************************************
7771                          ;
7772                          ;       THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
7773                          ;DATA PATH COMPONENTS WITH USER MODE SET.
7774                          ;
7775                          ;*****************************************************************************
7776                          ;TEST 253       TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
7777                          ;*****************************************************************************
7778  024464  005212          TS253:  INC     (R2)            ;UPDATE TEST NUMBER
7779  024466  022712  000253          CMP     #253,(R2)       ;SEQUENCE ERROR?
7780  024472  001017                  BNE     TS254-10        ;BR TO ERROR HALT ON SEQ ERROR
7781  024474  052767  140000  153274          BIS     #USRM,PS                ;SET USER MODE
7782  024502  012706  000001                  MOV     #1,R6                   ;SET BIT0
7783  024506  000241                          CLC                             ;CLEAR C-BIT
7784  024510  006106                  USP1:   ROL     R6                      ;ROTATE 1 POSITION
```

```
7785  024512  103376                        BCC     USP1                    ;BR IF NOT ALL DONE
7786  024514  001407                        BEQ     USP1A                   ;BR IF NO BITS PICKED
7787  024516  042767  140000  153252        BIC     #USRM,PS                ;CLEAR USER MODE
7788  024524  012742  000603                MOV     #603,-(R2)      ;MOVE TO MAILBOX #  *******  603  *******
7789  024530  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7790  024532  000000                        HALT                    ;USER MODE R6 PICKED A BIT
7791  024534  042767  140000  153234  USP1A: BIC     #USRM,PS                ;CLEAR USER MODE
7792
7793
7794                                  ;*************************************************************************************
7795                                  ;
7796                                  ;       THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
7797                                  ;AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
7798                                  ;OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
7799                                  ;OF EACH OTHER.
7800                                  ;
7801                                  ;*************************************************************************************
7802                                  ;TEST 254       TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
7803                                  ;*************************************************************************************
7804  024542  005212                  TS254:  INC     (R2)            ;UPDATE TEST NUMBER
7805  024544  022712  000254                  CMP     #254,(R2)       ;SEQUENCE ERROR?
7806  024550  001043                        BNE     USP4-10         ;BR TO ERROR HALT ON SEQ ERROR
7807  024552  052767  140000  153216        BIS     #USRM,PS                ;SET USER MODE
7808  024560  012706  177777                MOV     #-1,R6                  ;SET USER R6 TO ALL ONES
7809  024564  022706  177777                CMP     #-1,R6                  ;READ AND CHECK USER R6
7810  024570  001407                        BEQ     USP2                    ;BR IF NO ERROR
7811  024572  042767  140000  153176        BIC     #USRM,PS                ;CLEAR USER MODE
7812  024600  012742  000604                MOV     #604,-(R2)      ;MOVE TO MAILBOX #  *******  604  *******
7813  024604  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7814  024606  000000                        HALT                    ;USER R6 WILL NOT HOLD ALL ONES
7815  024610  042767  140000  153160  USP2: BIC     #USRM,PS                ;SET KERNEL MODE
7816  024616  022706  177777                CMP     #-1,R6                  ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
7817  024622  001004                        BNE     USP3
7818                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7819                                                          ;           CONDITIONAL BRANCH INST. AND   <====
7820                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7821                                                          ;           WHICH FOLLOWS W/ 752           <====
7822  024624  012742  000605                        MOV     #605,-(R2)      ;MOVE TO MAILBOX #  *******  605  *******
7823  024630  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7824  024632  000000                        HALT                    ;DUAL ADDRESSING ERROR USER/KERNEL R6
7825  024634  005006                  USP3: CLR     R6                      ;CLEAR KERNEL MODE SP
7826  024636  052767  140000  153132        BIS     #USRM,PS                ;SET USER MODE
7827  024644  022706  177777                CMP     #-1,R6                  ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7828  024650  042767  140000  153120        BIC     #USRM,PS                ;CLEAR USER MODE
7829  024656  001404                        BEQ     USP4                    ;BR IF NO ERROR
7830  024660  012742  000606                MOV     #606,-(R2)      ;MOVE TO MAILBOX #  *******  606  *******
7831  024664  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7832  024666  000000                        HALT                    ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7833  024670  012706  001000          USP4: MOV     #STBOT,R6               ;RESTORE SP USER
7834  024674  042767  140000  153074        BIC     #USRM,PS                ;SET KERNEL MODE
7835  024702  012706  001000                MOV     #STBOT,R6               ;RESTORE SP KERNEL
7836
7837                                  ;*************************************************************************************
7838                                  ;
7839                                  ;       THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
7840                                  ;WITH R6 IN MODE 0.
```

```
                                        ;
                                        ;*********************************************************************************
                                        ;TEST 255       TEST MFPI WITH R6 IN MODE 0
                                        ;*********************************************************************************
024706  005212                  TS255:  INC     (R2)            ;UPDATE TEST NUMBER
024710  022712  000255                  CMP     #255,(R2)       ;SEQUENCE ERROR?
024714  001032                          BNE     TS256-10        ;BR TO ERROR HALT ON SEQ ERROR
024716  012706  001000                  MOV     #STBOT,R6               ;INITIALIZE KERNEL STACK POINTER
024722  012767  140000  153046          MOV     #USRM,PS                ;SET USER MODE,PREVIOUS KERNEL
024730  012706  027300                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
024734  006506                          MFPI    R6                      ;TRY MFPI WITH MODE 0
024736  022767  140000  153032          CMP     #140000,PS              ;CHECK PSW
024744  001407                          BEQ     MFPI0                   ;BR IF NO ERROR
024746  042767  140000  153022          BIC     #USRM,PS                ;CLEAR USER MODE
024754  012742  000607                  MOV     #607,-(R2)      ;MOVE TO MAILBOX #  *******  607  *******
024760  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024762  000000                          HALT                    ;INCORRECT PSW FROM MFPI
024764  042767  140000  153004  MFPI0:  BIC     #USRM,PS                ;CLEAR USER MODE
024772  022767  001000  002276          CMP     #STBOT,USTBOT-2         ;CHECK DATA ON STACK
025000  001404                          BEQ     MFPI0A                  ;BR IF NO ERROR
025002  012742  000610                  MOV     #610,-(R2)      ;MOVE TO MAILBOX #  *******  610  *******
025006  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025010  000000                          HALT                    ;INCORRECT DATA FROM MFPI
025012                          MFPI0A:
                                        ;
                                        ;*********************************************************************************
                                        ;TEST 256       TEST MTPI WITH R6 IN MODE 0
                                        ;*********************************************************************************
025012  005212                  TS256:  INC     (R2)            ;UPDATE TEST NUMBER
025014  022712  000256                  CMP     #256,(R2)       ;SEQUENCE ERROR?
025020  001033                          BNE     TS257-10        ;BR TO ERROR HALT ON SEQ ERROR
025022  005067  152750                  CLR     PS                      ;SET KERNEL MODE
025026  005006                          CLR     R6                      ;INITIALIZE KERNEL R6
025030  012767  140000  152740          MOV     #USRM,PS                ;SET USER MODE/PREVIOUS KERNEL
025036  012706  027300                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
025042  012746  001000                  MOV     #STBOT,-(R6)            ;SET UP TARGET DATA
025046  006606                          MTPI    R6                      ;TRY MODE 0 MTPI
025050  022767  140000  152720          CMP     #USRM,PS                ;CHECK PSW
025056  001407                          BEQ     MTPI0                   ;BR IF NO ERROR
025060  042767  140000  152710          BIC     #USRM,PS                ;CLEAR USER MODE
025066  012742  000611                  MOV     #611,-(R2)      ;MOVE TO MAILBOX #  *******  611  *******
025072  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025074  000000                          HALT                    ;PS INCORRECT FOLLOWING MTPI
025076  005067  152674          MTPI0:  CLR     PS                      ;SET KERNEL MODE
025102  020627  001000                  CMP     R6,#STBOT               ;CHECK TARGET DATA
025106  001404                          BEQ     TS257
                                        ;
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <---
                                        ;           CONDITIONAL BRANCH INST. AND   <----
                                        ;           REPLACE THE MOVE INSTRUCTION   <---
                                        ;           WHICH FOLLOWS W/ 744           <---
025110  012742  000612                  MOV     #612,-(R2)      ;MOVE TO MAILBOX #  *******  612  *******
025114  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025116  000000                          HALT                    ;DATA INCORRECT FOLLOWING MTPI
                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 (YNTAB) 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 257       TEST THE BRANCH ROM
                                ;*****************************************************************************
025120  005212                  TS257:  INC     (R2)            ;UPDATE TEST NUMBER
025122  022712  000257                  CMP     #257,(R2)       ;SEQUENCE ERROR?
025126  001062                          BNE     ER              ;BR TO ERROR HALT ON SEQ ERROR
025130  012700  027170          SETUP:  MOV     #BRTAB,R0       ;INITIALIZE BRANCH TABLE POINTER
025134  012704  027226                  MOV     #YNTAB,R4       ;INITIALIZE YES/NO BRANCH MAP POINTER
025140  012767  000017  000142          MOV     #15.,BRCT       ;INITIALIZE BRANCH TABLE COUNT
025146  012067  000110          SETBR:  MOV     (R0)+,BRH       ;GET NEXT BRANCH INST.
025152  012401                          MOV     (R4)+,R1        ;GET NEXT BRANCH MAP
025154  012767  177777  000074          MOV     #-1,CC1         ;INITIALIZE CONDITION CODE VALUE
025162  012703  000020                  MOV     #16.,R3         ;INITIALIZE CONDITION CODE COUNT
025166  005267  000064          SETCC:  INC     CC1             ;SET FOR NEXT CC VALUE
025172  032701  100000                  BIT     #100000,R1      ;SEE IF SHOULD BR W/ THESE CC'S
025176  013705  177776                  MOV     @#177776,R5     ;SIMULATE A JNE
025202  042705  177773                  BIC     #177773,R5      ;       (JUMP NOT EQUAL)
025206  000165  025212                  JMP     .+4(R5)         ;       TO SET2BR
025212  000167  000020                  JMP     SET2BR
025216  012767  025312  000042          MOV     #CONT,NBR       ;SET TO CONTINUE IF NO BRANCH
025224  012767  025274  000040          MOV     #ER,YBR         ;SET TO REPORT ERROR IF BRANCH
025232  000167  000014                  JMP     AROUN           ;GO AROUND OPPOSITE CONDITION
025236  012767  025274  000022  SET2BR: MOV     #ER,NBR         ;SET TO REPORT ERROR IF NO BRANCH
025244  012767  025312  000020          MOV     #CONT,YBR       ;SET TO CONTINUE IF BRANCH
025252  006101                  AROUN:  ROL     R1              ;UPDATE BIT MAP
                                
025254  012737                          MOV     (PC)+,@(PC)+    ;SET CONDITION CODE
025256  000000          CC1:    0                               ;NEW CC VALUE GOES HERE
025260  177776                          177776
025262  000000          BRH:    0                               ;BRANCH INST. GOES HERE
025264  000137                  .JMP    @(PC)+                  ;THIS JUMP IF NO BRANCH
025266  000000          NBR:    0                               ;WHERE TO GO IF NO BRANCH OCCURS
025270  000137                  JMP     @(PC)+                  ;THIS JUMP IF BRANCH OCCURS
```

```
7952  025272  000000            YBR:    0
7953  025274  012702  000304    ER:     MOV     #$TESTN,R2      ;RESTORE POINTER
7954  025300  012742  000613            MOV     #613,-(R2)      ;MOVE TO MAILBOX #  *******  613  *******
7955  025304  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7956  025306  000000                    HALT                    ;
7957  025310  000000            BRCT:   0
7958  025312  005303            CONT:   DEC     R3              ;CC'S DONE?
7959  025314  013705  177776            MOV     @#177776,R5     ;SIMULATE A JNE
7960  025320  042705  177773            BIC     #177773,R5      ;       (JUMP NOT EQUAL)
7961  025324  000165  025330            JMP     .+4(R5)         ;       TO SETCC
7962  025330  000167  177632            JMP     SETCC
7963  025334  005367  177750            DEC     BRCT            ;BR'S DONE?
7964  025340  013705  177776            MOV     @#177776,R5     ;SIMULATE A JNE
7965  025344  042705  177773            BIC     #177773,R5      ;       (JUMP NOT EQUAL)
7966  025350  000165  025354            JMP     .+4(R5)         ;       TO SETBR
7967  025354  000167  177566            JMP     SETBR
```

```
7968
7969                                    ;*****************************************************************************
7970                                    ;
7971                                    ;THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL
7972                                    ;REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET
7973                                    ;IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH
7974                                    ;REGISTER.
7975                                    ;
7976                                    ;*****************************************************************************
7977                                    ;TEST 260       DUAL REGISTER ADDRESSING TEST
7978                                    ;*****************************************************************************
7979  025360  005212                    TS260:  INC     (R2)            ;UPDATE TEST NUMBER
7980  025362  022712  000260                    CMP     #260,(R2)       ;SEQUENCE ERROR?
7981  025366  001052                            BNE     DAERR           ;BR TO ERROR HALT ON SEQ ERROR
7982  025370  005000                    BITCLR: CLR     R0              ;INITIALIZE ALL REGISTERS
7983  025372  005001                            CLR     R1
7984  025374  005002                            CLR     R2
7985  025376  005003                            CLR     R3
7986  025400  005004                            CLR     R4
7987  025402  005005                            CLR     R5
7988  025404  005006                            CLR     R6
7989  025406  052700  000001          BITSET: BIS     #1,R0           ;SET R0=1
7990  025412  052701  000002                  BIS     #2,R1           ;R1-2
7991  025416  052702  000004                  BIS     #4,R2           ;R2=4
7992  025422  052703  000010                  BIS     #10,R3          ;R3-10
7993  025426  052704  000020                  BIS     #20,R4          ;R4=20
7994  025432  052705  000040                  BIS     #40,R5          ;R5=40
7995  025436  052706  000100                  BIS     #100,R6         ;R6=100
7996  025442  022706  000100          BITCHK: CMP     #100,R6         ;TEST THAT NO DUAL ADDRESSING OCCURRED
7997  025446  001022                            BNE     DAERR           ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
7998  025450  022705  000040                  CMP     #40,R5
7999  025454  001017                            BNE     DAERR
8000  025456  022704  000020                  CMP     #20,R4
8001  025462  001014                            BNE     DAERR
8002  025464  022703  000010                  CMP     #10,R3
8003  025470  001011                            BNE     DAERR
8004  025472  022702  000004                  CMP     #4,R2
8005  025476  001006                            BNE     DAERR
8006  025500  022701  000002                  CMP     #2,R1
8007  025504  001003                            BNE     DAERR
8008  025506  022700  000001                  CMP     #1,R0
8009  025512  001404                            BEQ     BITCON
8010                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
8011                                                                    ;           CONDITIONAL BRANCH INST. AND    <====
8012                                                                    ;           REPLACE THE MOVE INSTRUCTION    <====
8013                                                                    ;           WHICH FOLLOWS W/ 725            <====
8014  025514                    DAERR:
8015  025514  012742  000614                  MOV     #614,-(R2)      ;MOVE TO MAILBOX #  ***:***  614  *******
8016  025520  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8017  025522  000000                            HALT                    ;DUAL ADDRESSING ERROR
8018  025524  012702  000304          BITCON: MOV     #$TESTN,R2      ;RESTORE POINTER
8019
```

```
                                ;*****************************************************************************
                                ;       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 261       TEST BYTE INSTRUCTION ON PSW
                                ;*****************************************************************************
025530  005212                TS261:  INC     (R2)            ;UPDATE TEST NUMBER
025532  022712  000261              CMP     #261,(R2)       ;SEQUENCE ERROR?
025536  001012                      BNE     BTERR           ;BR TO ERROR HALT ON SEQ ERROR
025540  052737  170357  177776      BIS     #170357,@#PS    ;SET ALL POSSIBLE BITS IN PSW
025546  105037  177776              CLRB    @#PS            ;CLR PR LEVEL AND CC'S
025552  013700  177776              MOV     @#PS,R0         ;COPY CONTENTS OF PSW
025556  032700  170000              BIT     #170000,R0      ;TEST THAT UPPER BYTE IS UNAFFECTED
025562  001006                      BNE     BTCON           ;CONTINUE IF OK
025564  005037  177776        BTERR:  CLR     @#PS            ;RETURN TO KERNEL MODE
025570  012742  000615              MOV     #615,-(R2)      ;MOVE TO MAILBOX #  *******  615  *******
025574  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025576  000000                      HALT                    ;BYTE INSTRUCTION ALTERED PSW
025600  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 262       TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES
                                ;*****************************************************************************
025604  005212                TS262:  INC     (R2)            ;UPDATE TEST NUMBER
025606  022712  000262              CMP     #262,(R2)       ;SEQUENCE ERROR?
025612  001010                      BNE     TS263-10        ;BR TO ERROR HALT ON SEQ ERROR
025614  000277                      SCC
025616  000252                      +CLN!CLV                ;CC=0101
025620  000167  000000              JMP     JMPT            ;JUMP TO TEST PSW
025624  100403                JMPT:   BMI     JMPERR          ;BR TO ERROR HALT IF N-BIT IS SET
025626  001002                      BNE     JMPERR          ;BR TO ERROR HALT IF Z-BIT IS CLEAR
025630  102401                      BVS     JMPERR          ;BR TO ERROR HALT IF V-BIT IF SET
025632  103404                      BCS     TS263
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <----
                                                            ;           CONDITIONAL BRANCH INST. AND   <===
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 767           <---
025634                          JMPERR:
025634  012742  000616              MOV     #616,-(R2)      ;MOVE TO MAILBOX #  *******  616  *******
025640  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025642  000000                      HALT                    ;JMP INSTRUCTION AFFECTED CC'S
                                                            ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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 263        TEST SET CC AND CLEAR CC INSTRUCTIONS
                                ;*****************************************************************************
025644  005212                  TS263:  INC     (R2)            ;UPDATE TEST NUMBER
025646  022712  000263                  CMP     #263,(R2)       ;SEQUENCE ERROR?
025652  001062                          BNE     CCERR           ;BR TO ERROR HALT ON SEQ ERROR
025654  012767  000240  000024          MOV     #240,CC3        ;INITIALIZE CLR CC INSTRUCTION CODES
025662  012767  000017  000032          MOV     #17,CC2         ;INITIALIZE OCTAL MAP
025670  012767  000261  000102          MOV     #261,SC3        ;INITIALIZE SET CC INSTRUCTION CODES
025676  012767  000001  000110          MOV     #1,SC4          ;INITIALIZE OCTAL MAP
025704  000277                  CLRCD:  SCC                     ;SET ALL CONDITION CODES
025706  000000                  CC3:    0                       ;CONDITION CODE INSTRUCTION
025710  013704  177776                  MOV     @#PS,R4         ;COPY THE PSW
025714  042704  177760                  BIC     #177760,R4      ;ISOLATE CONDITION CODES
025720  022704                          CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE CLEARED
025722  000000                  CC2:    0                       ;OCTAL REPRESENTATION OF CC'S
025724  001404                          BEQ     CON1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 752           <====
025726  012742  000617                  MOV     #617,-(R2)      ;MOVE TO MAILBOX #  *******  617  *******
025732  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025734  000000                          HALT                    ;CLEAR CC INSTRUCTION FAILED
025736  005367  177760          CON1:   DEC     CC2             ;SET NEXT OCTAL MAP OF CC'S
025742  005267  177740                  INC     CC3             ;GET NEXT CLEAR CC INSTRUCTION
025746  026727  177734  000257          CMP     CC3,#257        ;TEST FOR CCC INSTRUCTION
025754  003753                          BLE     CLRCD           ;GO TEST NEXT INSTRUCTION IF NOT FOUND
025756  026727  177724  000260          CMP     CC3,#260        ;CHECK FOR NOP=260
025764  001004                          BNE     SETCD           ;GO TEST SET CC INSTRUCTIONS
025766  012767  000017  177726          MOV     #17,CC2         ;SET OCTAL MAP TO TEST NOP
025774  000743                          BR      CLRCD           ;GO TEST NOP
025776  000257                  SETCD:  CCC                     ;CLEAR ALL CONDITION CODES
026000  000000                  SC3:    0                       ;CONDITION CODE INSTRUCTION
026002  013704  177776                  MOV     @#PS,R4         ;COY PSW
026006  042704  177760                  BIC     #177760,R4      ;CLEAR AWAY UNWANTED BITS
026012  022704                          CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE SET
026014  000000                  SC4:    0                       ;OCTAL REPRESENTATION OF CC'S
026016  001404                          BEQ     CON2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <    -
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
```

```
8127                                                    ;           WHICH FOLLOWS W/ 715          <---
8128  026020                            CCERR:
8129  026020  012742  000620                  MOV     #620,-(R2)    ;MOVE TO MAILBOX #  *******  620  *******
8130  026024  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
8131  026026  000000                            HALT                ;SET CC FAILED OR SEQUENCE ERROR
8132  026030  005267  177760           CON2:   INC     SC4         ;SET NEXT OCTAL MAP
8133  026034  005267  177740                    INC     SC3         ;PREPARE NEXT SET CC INSTRUCTION
8134  026040  026727  177734  000277            CMP     SC3,#277    ;FINISHED?
8135  026046  003753                            BLE     SETCD       ;BR IF NO
8136  026050  000167  000006                    JMP     MORO        ;JUMP TO NEXT TESTS
```

```
8137                                    ;*****************************************************************************
8138                                    .SBTTL  TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION
8139                                    ;
8140                                    ;IN AUTO INCREMENT (DECREMENT) MODES AND
8141                                    ;AUTO INCREMENT (DECREMENT) DEFERRED MODES,
8142                                    ;CONTENTS OF THE REGISTER IN USED ARE
8143                                    ;INCREMENTED (DECREMENTED) BY 2
8144                                    ;BEFORE USED AS THE SOURCE OPERAND.
8145                                    ;
8146  026054  000000  000000  000000  A:      .WORD 0,0,0
8147  026062                          MOR0:
8148                                    ;*****************************************************************************
8149                                    ;TEST 264       TEST AUTO-INCREMENT MODE, USING R0
8150                                    ;*****************************************************************************
8151  026062  005212                  TS264:  INC     (R2)            ;UPDATE TEST NUMBER
8152  026064  022712  000264                  CMP     #264,(R2)       ;SEQUENCE ERROR?
8153  026070  001020                          BNE     TS265-10        ;BR TO ERROR HALT ON SEQ ERROR
8154  026072  005037  026054                  CLR     @#A             ;CLEAR LOC A
8155  026076  012700  026054                  MOV     #A,R0           ;R0 STORES ADDR OF A
8156  026102  060020                          ADD     R0,(R0)+        ;CHECK THAT R0 IS INCR BY 2 BEFORE
8157                                                                  ;BEING USED AS THE SOURCE OPERAND
8158  026104  022700  026056                  CMP     #A+2,R0         ;R0 INCR BY 2?
8159  026110  001404                          BEQ     MOR1
8160                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- -
8161                                                                  ;           CONDITIONAL BRANCH INST. AND   <- -
8162                                                                  ;           REPLACE THE MOVE INSTRUCTION   <==-
8163                                                                  ;           WHICH FOLLOWS W/ 767           <-
8164  026112  012742  000621                  MOV     #621,-(R2)      ;MOVE TO MAILBOX #  *******  621  *******
8165  026116  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8166  026120  000000                          HALT                    ;R0 WAS NOT INCREMENTED BY 2
8167                                                                  ;
8168  026122  022737  026056  026054  MOR1:  CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8169                                                                  ;BEING USED IN THE ''ADD'' INSTR
8170                                                                  ;LOC A CONTAINS (A+2)?
8171  026130  001404                          BEQ     TS265
8172                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-==-
8173                                                                  ;           CONDITIONAL BRANCH INST. AND   <-=-
8174                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-==-
8175                                                                  ;           WHICH FOLLOWS W/ 757           <==-=
8176  026132  012742  000622                  MOV     #622,-(R2)      ;MOVE TO MAILBOX #  *******  622  *******
8177  026136  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8178  026140  000000                          HALT                    ;WRONG SUM IN LOC A
8179                                                                  ;  OR SEQUENCE ERROR
8180
8181                                    ;*****************************************************************************
8182                                    ;TEST 265       AUTO-DECREMENT MODE, USING R0
8183                                    ;*****************************************************************************
8184  026142  005212                  TS265:  INC     (R2)            ;UPDATE TEST NUMBER
8185  026144  022712  000265                  CMP     #265,(R2)       ;SEQUENCE ERROR?
8186  026150  001020                          BNE     TS266-10        ;BR TO ERROR HALT ON SEQ ERROR
8187  026152  005037  026054                  CLR     @#A             ;CLEAR LOC A
8188  026156  012700  026056                  MOV     #A+2,R0         ;R0 STORES ADDR OF A+2
8189  026162  060040                          ADD     R0,-(R0)        ;CHECK THAT R0 IS DECR BY 2 BEFORE
8190                                                                  ;BEING USED AS THE SOURCE OPERAND
8191  026164  022700  026054                  CMP     #A,R0           ;R0 DECR BY 2?
8192  026170  001404                          BEQ     MOR2
```

```
8193                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8194                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8195                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8196                                                    ;           WHICH FOLLOWS W/ 767           <====
8197  026172  012742  000623                MOV   #623,-(R2)    ;MOVE TO MAILBOX #  *******  623  *******
8198  026176  005242                        INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8199  026200  000000                        HALT                ;R0 WAS NOT DECREMENTED BY 2
8200                                                            ;
8201  026202  022737  026054  026054 MOR2:  CMP   #A,@#A        ;CONTENT OF R0 WAS DECR BY 2 BEFORE
8202                                                            ;BEING USED IN THE ''ADD'' INSTR
8203                                                            ;LOC A CONTAINS (R0)
8204  026210  001404                        BEQ   TS266
8205                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8206                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8207                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8208                                                    ;           WHICH FOLLOWS W/ 757           <====
8209  026212  012742  000624                MOV   #624,-(R2)    ;MOVE TO MAILBOX #  *******  624  *******
8210  026216  005242                        INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8211  026220  000000                        HALT                ;WRONG SUM IN LOC A
8212                                                            ;  OR SEQUENCE ERROR
8213                                                            ;
8214                                ;*******************************************************************************
8215                                ;TEST 266       TEST AUTO-INCREMENT DEFERRED MODE, USING R0
8216                                ;*******************************************************************************
8217  026222  005212                TS266: INC    (R2)          ;UPDATE TEST NUMBER
8218  026224  022712  000266               CMP   #266,(R2)     ;SEQUENCE ERROR?
8219  026230  001044                       BNE   TS267-10      ;BR TO ERROR HALT ON SEQ ERROR
8220  026232  005037  026054               CLR   @#A           ;CLEAR LOC A
8221  026236  005037  026060               CLR   @#A+4         ;CLEAR LOC A+4
8222  026242  012737  026054  026056       MOV   #A,@#A+2      ;STORE ADDR A IN LOC A+2
8223  026250  012700  026056               MOV   #A+2,R0       ;R0 STORES ADDR A+2
8224  026254  060030                       ADD   R0,@(R0)+     ;CHECK THAT R0 IS INCR BY 2 BEFORE
8225                                                            ;BEING USED AS THE SOURCE OPERAND
8226  026256  022700  026060               CMP   #A+4,R0       ;R0 INCR BY 2?
8227  026262  001404                       BEQ   MOR3
8228                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8229                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8230                                                    ;           REPLACE THE MOVE INSTRUCTION   <==-=
8231                                                    ;           WHICH FOLLOWS W/ 762           <=-==
8232  026264  012742  000625                MOV   #625,-(R2)    ;MOVE TO MAILBOX #  *******  625  *******
8233  026270  005242                        INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8234  026272  000000                        HALT                ;R0 WAS NOT INCREMENTED BY 2
8235                                                            ;
8236  026274  022737  026054  026056 MOR3:  CMP   #A,@#A+2      ;LOC A+2 STILL STORES ADDR A?
8237  026302  001404                        BEQ   MOR4
8238                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==--
8239                                                    ;           CONDITIONAL BRANCH INST. AND   <==--
8240                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8241                                                    ;           WHICH FOLLOWS W/ 752           <===-
8242  026304  012742  000626                MOV   #626,-(R2)    ;MOVE TO MAILBOX #  *******  626  *******
8243  026310  005242                        INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8244  026312  000000                        HALT                ;LOC A+2 STORES WRONG DATA
8245                                                            ;
8246  026314  022737  026060  026054 MOR4:  CMP   #A+4,@#A      ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8247                                                            ;BEING USED IN THE ''ADD'' INSTR
8248  026322  001404                        BEQ   MOR5
```

```
8249                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8250                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8251                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8252                                                    ;           WHICH FOLLOWS W/ 742           <====
8253  026324  012742  000627                 MOV     #627,-(R2)      ;MOVE TO MAILBOX #  *******  627  *******
8254  026330  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8255  026332  000000                         HALT                    ;LOC A STORES WRONG DATA
8256                                                                  ;
8257  026334  005737  026060       MOR5:     TST     @#A+4           ;LOC A+4 STILL STORES 0?
8258  026340  001404                         BEQ     TS267
8259                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8260                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8261                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8262                                                    ;           WHICH FOLLOWS W/ 733           <====
8263  026342  012742  000630                 MOV     #630,-(R2)      ;MOVE TO MAILBOX #  *******  630  *******
8264  026346  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8265  026350  000000                         HALT                    ;LOC A+4 DID NOT STAY CLEAR
8266                                                                  ;  OR SEQUENCE ERROR
8267
8268                                    ;*************************************************************************************
8269                                    ;TEST 267       TEST AUTO-DECREMENT DEFERRED, USING R0
8270                                    ;*************************************************************************************
8271  026352  005212                    TS267:  INC     (R2)            ;UPDATE TEST NUMBER
8272  026354  022712  000267                    CMP     #267,(R2)       ;SEQUENCE ERROR?
8273  026360  001044                         BNE     TS270-10        ;BR TO ERROR HALT ON SEQ ERROR
8274  026362  005037  026054                CLR     @#A             ;CLEAR LOC A
8275  026366  005037  026060                CLR     @#A+4           ;CLEAR LOC A+4
8276  026372  012700  026060                MOV     #A+4,R0         ;R0 STORES ADDR A+4
8277  026376  012737  026054  026056        MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
8278  026404  060050                         ADD     R0,@-(R0)       ;CHECK THAT R0 IS DECR BY 2 BEFORE
8279                                                                  ;BEING USED AS THE SOURCE OPERAND
8280  026406  022700  026056                CMP     #A+2,R0         ;R0 DECREMENTFD BY 2?
8281  026412  001404                         BEQ     MOR6
8282                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-=
8283                                                    ;           CONDITIONAL BRANCH INST. AND   <-- -
8284                                                    ;           REPLACE THE MOVE INSTRUCTION   <   =
8285                                                    ;           WHICH FOLLOWS W/ 762           <= -=
8286  026414  012742  000631                 MOV     #631,-(R2)      ;MOVE TO MAILBOX #  *******  631  *******
8287  026420  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8288  026422  000000                         HALT                    ;R0 WAS NOT DECREMENTED BY 2
8289                                                                  ;
8290  026424  022737  026056  026054 MOR6:   CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
8291                                                                  ;BEING USED IN THE "ADD" INSTR
8292  026432  001404                         BEQ     MOR7
8293                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-= =
8294                                                    ;           CONDITIONAL BRANCH INST. AND   <=-==
8295                                                    ;           REPLACE THE MOVE INSTRUCTION   <===-
8296                                                    ;           WHICH FOLLOWS W/ 752           <====
8297  026434  012742  000632                 MOV     #632,-(R2)      ;MOVE TO MAILBOX #  *******  632  *******
8298  026440  005242                         INC     -(R2)           ;SET MSGTYP TC FATAL ERROR
8299  026442  000000                         HALT                    ;LOC A STORES WRONG DATA
8300
8301                                                                  ;
8302  026444  022737  026054  026056 MOR7:   CMP     #A,@#A+2        ;LOC A+2 STILL STORES A?
8303  026452  001404                         BEQ     MOR8
8304                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
```

```
8305                                                    ;           CONDITIONAL BRANCH INST. AND    <====
8306                                                    ;           REPLACE THE MOVE INSTRUCTION    <====
8307                                                    ;           WHICH FOLLOWS W/ 742            <====
8308  026454  012742  000633          MOV     #633,-(R2)  ;MOVE TO MAILBOX #  *******  633  *******
8309  026460  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
8310  026462  000000                  HALT                ;LOC A+2 STORES WRONG DATA
8311                                                    ;
8312  026464  005737  026060  MOR8:   TST     @#A+4       ;LOC A+4 STILL STORES 0?
8313  026470  001404                  BEQ     TS270
8314                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
8315                                                    ;           CONDITIONAL BRANCH INST. AND    <====
8316                                                    ;           REPLACE THE MOVE INSTRUCTION    <====
8317                                                    ;           WHICH FOLLOWS W/ 733            <====
8318  026472  012742  000634          MOV     #634,-(R2)  ;MOVE TO MAILBOX #  *******  634  *******
8319  026476  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
8320  026500  000000                  HALT                ;LOC A+4 DID NOT STAY CLEAR
8321                                                    ;  OR SEQUENCE ERROR
8322                                                    ;
```

```
                                ;*****************************************************************************
                                .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 270       TEST PC AS SOURCE IN MODE 0, USING R0
                                ;*****************************************************************************
026502  005212                  TS270:  INC     (R2)            ;UPDATE TEST NUMBER
026504  022712  000270                  CMP     #270,(R2)       ;SEQUENCE ERROR?
026510  001006                          BNE     TS271-10        ;BR TO ERROR HALT ON SEQ ERROR
026512  012700  177777                  MOV     #-1,R0          ;SET ALL 1 IN R0
026516  010700                  PCN01:  MOV     PC,R0           ;STORES PC IN R0
026520  022700  026520                  CMP     #PCN01+2,R0     ;R0 STORES PC+2?
026524  001404                          BEQ     TS271
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
026526  012742  000635                  MOV     #635,-(R2)      ;MOVE TO MAILBOX #  *******  635  *******
026532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026534  000000                          HALT                    ;R0 STORED WRONG VALUE
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 271       TEST PC AS SOURCE IN MODE 6, USING R0
                                ;*****************************************************************************
026536  005212                  TS271:  INC     (R2)            ;UPDATE TEST NUMBER
026540  022712  000271                  CMP     #271,(R2)       ;SEQUENCE ERROR?
026544  001010                          BNE     TS272-10        ;BR TO ERROR HALT ON SEQ ERROR
026546  012700  026054                  MOV     #A,R0           ;R0 STORES ADDR A
026552  010760  000004          PCN2:   MOV     PC,4(R0)                ;EFFECTIVE ADDR IS A+4
026556  022737  026556  026060        CMP     #PCN2+4,@#A+4   ;LOC A+4 STORES PC+4?
026564  001404                          BEQ     TS272
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <==  -
                                                                ;           REPLACE THE MOVE INSTRUCTION   <==--
                                                                ;           WHICH FOLLOWS W/ 767           <   -=
026566  012742  000636                  MOV     #636,-(R2)      ;MOVE TO MAILBOX #  *******  636  *******
026572  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026574  000000                          HALT                    ;LOC A+4 STORED WRONG VALUE
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 272       TEST PC AS SOURCE IN MODE 7, USING R0
                                ;*****************************************************************************
026576  005212                  TS272:  INC     (R2)            ;UPDATE TEST NUMBER
026600  022712  000272                  CMP     #272,(R2)       ;SEQUENCE ERROR?
026604  001013                          BNE     TS273-10        ;BR TO ERROR HALT ON SEQ ERROR
026606  012737  026054  026060        MOV     #A,@#A+4        ;LOC A+4 STORES ADDR A
026614  012700  026054                  MOV     #A,R0           ;R0 STORES ADDR A
026620  010770  000004          PCN3:   MOV     PC,@4(R0)       ;EFFECTIVE ADDR IS A
026624  022737  026624  026054        CMP     #PCN3+4,@#A     ;LOC A STORES PC+4?
026632  001404                          BEQ     TS273
```

```
8379                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
8380                                                    ;           CONDITIONAL BRANCH INST. AND   <-==
8381                                                    ;           REPLACE THE MOVE INSTRUCTION   <-===
8382                                                    ;           WHICH FOLLOWS W/ 764           <-
8383  026634  012742  000637                  MOV     #637,-(R2)      ;MOVE TO MAILBOX #  *******  637  *******
8384  026640  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8385  026642  000000                          HALT                    ;LOC A STORED WRONG VALUE
8386                                                    ;  OR SEQUENCE ERROR
8387
8388                            ;*****************************************************************************
8389                            ;TEST 273       TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0
8390                            ;*****************************************************************************
8391  026644  005212            TS273:  INC     (R2)            ;UPDATE TEST NUMBER
8392  026646  022712  000273            CMP     #273,(R2)       ;SEQUENCE ERROR?
8393  026652  001011                    BNE     TS274-10        ;BR TO ERROR HALT ON SEQ ERROR
8394  026654  012737  026056  026054    MOV     #A+2,@#A        ;LOC A STORES ADDR A+2
8395  026662  010777  177166    PCN4:   MOV     PC,@A           ;EFFECTIVE ADDR IS A+2
8396  026666  022737  026666  026056    CMP     #PCN4+4,@#A+2   ;LOC A+2 STORES PC+4?
8397  026674  001404                    BEQ     TS274
8398                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
8399                                                    ;           CONDITIONAL BRANCH INST. AND   <
8400                                                    ;           REPLACE THE MOVE INSTRUCTION   <-
8401                                                    ;           WHICH FOLLOWS W/ 766           <
8402  026676  012742  000640                  MOV     #640,-(R2)      ;MOVE TO MAILBOX #  *******  640  *******
8403  026702  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8404  026704  000000                          HALT                    ;LOC A+2 STORED WRONG VALUE
8405                                                    ;  OR SEQUENCE ERROR
8406
8407                            ;*****************************************************************************
8408                            ;TEST 274       TEST PC AS SOURCE IN RELATIVE MODE ,USING R0
8409                            ;*****************************************************************************
8410  026706  005212            TS274:  INC     (R2)            ;UPDATE TEST NUMBER
8411  026710  022712  000274            CMP     #274,(R2)       ;SEQUENCE ERROR?
8412  026714  001010                    BNE     TS275-10        ;BR TO ERROR HALT ON SEQ ERROR
8413  026716  005037  026054            CLR     @#A             ;CLEAR A
8414  026722  010767  177126    PCN5:   MOV     PC,A            ;EFFECTIVE ADDR IS A
8415  026726  022737  026726  026054    CMP     #PCN5+4,@#A     ;LOC A STORES PC+4?
8416  026734  001404                    BEQ     TS275
8417                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
8418                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8419                                                    ;           REPLACE THE MOVE INSTRUCTION   <-===
8420                                                    ;           WHICH FOLLOWS W/ 767           <====
8421  026736  012742  000641                  MOV     #641,-(R2)      ;MOVE TO MAILBOX #  *******  641  *******
8422  026742  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8423  026744  000000                          HALT                    ;LOCATION A STORED WRONG VALUE
8424                                                    ;  OR SEQUENCE ERROR
8425
8426                            ;*****************************************************************************
8427                            .SBTTL  THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.
8428                            ;*****************************************************************************
8429                            ;TEST 275       TEST SUB INSTRUCTION, SM=0, DM=2
8430                            ;*****************************************************************************
8431  026746  005212            TS275:  INC     (R2)            ;UPDATE TEST NUMBER
8432  026750  022712  000275            CMP     #275,(R2)       ;SEQUENCE ERROR?
8433  026754  001013                    BNE     TS276-10        ;BR TO ERROR HALT ON SEQ ERROR
8434  026756  012737  052525  000000    MOV     #052525,@#0     ;SET UP LOC 0
```

```
8435  026764  012701  050505                  MOV     #050505,R1      ;SET UP R1
8436  026770  005000                          CLR     R0              ;CLEAR R0
8437  026772  160120                          SUB     R1,(R0)+        ;SUBTRACTION, SM=0,DM-2
8438  026774  022737  002020  000000          CMP     #2020,@#0       ;CHECK DIFFERENCE AT LOC 0
8439  027002  001404                          BEQ     TS276
8440                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=--
8441                                                                  ;           CONDITIONAL BRANCH INST. AND   <=---
8442                                                                  ;           REPLACE THE MOVE INSTRUCTION   <===
8443                                                                  ;           WHICH FOLLOWS W/ 764           <
8444  027004  012742  000642                  MOV     #642,-(R2)      ;MOVE TO MAILBOX #  *******  642  *******
8445  027010  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8446  027012  000000                          HALT                    ;WRONG RESULT FROM SUBTRACTION
8447                                                                  ;  OR SEQUENCE ERROR
8448
8449                                  ;*****************************************************************************
8450                                  ;TEST 276       TEST MFPD WITH R0, IN MODE 2
8451                                  ;*****************************************************************************
8452  027014  005212                  TS276:  INC     (R2)            ;UPDATE TEST NUMBER
8453  027016  022712  000276                  CMP     #276,(R2)       ;SEQUENCE ERROR?
8454  027022  001020                          BNE     TS277-10        ;BR TO ERROR HALT ON SEQ ERROR
8455  027024  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
8456  027032  005000                          CLR     R0              ;CLEAR R0
8457  027034  012767  170000  150734          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
8458  027042  012706  027300                  MOV     #USTBOT,R6      ;SET USER STACK POINTER
8459  027046  106520                          MFPD    (R0)+           ;MODE 2, MFPD
8460  027050  005067  150722                  CLR     PS              ;SET KERNEL MODE
8461  027054  022767  052525  000214          CMP     #052525,USTBOT-2        ;CHECK DATA ON STACK
8462  027062  001404                          BEQ     BRMFPD          ;BR IF NO ERROR
8463  027064  012742  000643                  MOV     #643,-(R2)      ;MOVE TO MAILBOX #  *******  643  *******
8464  027070  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8465  027072  000000                          HALT                    ;INCORRECT DATA FROM MFPD
8466  027074                          BRMFPD:
8467
8468
8469                                  ;*****************************************************************************
8470                                  ;TEST 277       TEST MTPD WITH R0, IN MODE 2
8471                                  ;*****************************************************************************
8472  027074  005212                  TS277:  INC     (R2)            ;UPDATE TEST NUMBER
8473  027076  022712  000277                  CMP     #277,(R2)       ;SEQUENCE ERROR?
8474  027102  001026                          BNE     END1            ;BR TO ERROR HALT ON SEQ ERROR
8475  027104  012767  170000  150664          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
8476  027112  012706  027300                  MOV     #USTBOT,R6      ;SET USER STACK POINTER
8477  027116  012746  125252                  MOV     #125252,-(R6)   ;PUSH DATA IN USER STACK
8478  027122  012737  000000  000000          MOV     #0,@#0          ;CLEAR LOC 0
8479  027130  005000                          CLR     R0              ;CLEAR R0
8480  027132  106620                          MTPD    (R0)+           ;MODE 2, MTPD
8481  027134  005067  150636                  CLR     PS              ;SET KERNEL MODE
8482  027140  022737  125252  000000          CMP     #125252,@#0     ;CHECK DATA ON LOC 0
8483  027146  001514                          BEQ     SECPRT          ;BR TO TRAP TEST IF NO ERROR
8484  027150  012742  000644                  MOV     #644,-(R2)      ;MOVE TO MAILBOX #  *******  644  *******
8485  027154  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8486  027156  000000                          HALT                    ;INCORRECT DATA FROM MTPD
8487  027160                          END1:
8488  027160  012742  000645                  MOV     #645,-(R2)      ;MOVE TO MAILBOX #  *******  645  *******
8489  027164  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8490  027166  000000                          HALT                    ;SEQUENCE ERROR
```

```
8491
8492  027170  000402                BRTAB:  BR      .+6
8493  027172  001002                        BNE     .+6
8494  027174  001402                        BEQ     .+6
8495  027176  002002                        BGE     .+6
8496  027200  002402                        BLT     .+6
8497  027202  003002                        BGT     .+6
8498  027204  003402                        BLE     .+6
8499  027206  100002                        BPL     .+6
8500  027210  100402                        BMI     .+6
8501  027212  101002                        BHI     .+6
8502  027214  101402                        BLOS    .+6
8503  027216  102002                        BVC     .+6
8504  027220  102402                        BVS     .+6
8505  027222  103002                        BCC     .+6             ;SAME AS BHIS
8506  027224  103402                        BCS     .+6             ;SAME AS BLO
8507
8508          000002                .RADIX  2
8509  027226  177777                YNTAB:  1111111111111111        ;BR
8510  027230  170360                        1111000011110000        ;BNE:  Z=0
8511  027232  007417                        0000111100001111        ;BEQ:  Z=1
8512  027234  146063                        1100110000110011        ;BGE:  N XOR V =0
8513  027236  031714                        0011001111001100        ;BLT:  N XOR V =1
8514  027240  140060                        1100000000110000        ;BGT:  Z+(N XOR V) =0
8515  027242  037717                        0011111111001111        ;BLE:  Z+(N XOR V) =1
8516
8517  027244  177400                        1111111100000000        ;BPL:  N=0
8518  027246  000377                        0000000011111111        ;BMI:  N=1
8519  027250  120240                        1010000010100000        ;BHI:  C+Z=0
8520  027252  057537                        0101111101011111        ;BLOS: C+Z=1
8521  027254  146314                        1100110011001100        ;BVC:  V-0
8522  027256  031463                        0011001100110011        ;BVS:  V=1
8523  027260  125252                        1010101010101010        ;BCC:  C=0
8524  027262  052525                        0101010101010101        ;BCS:  C=1
8525          000010                .RADIX  8
8526
8527                                .EVEN
8528  027264  000006                .BLKW   6
8529  027300                        USTBOT:
8530                                ;*****************************************************************************
8531                                ; THE FOLLOWING ARE SPECIAL CPU TRAP
8532                                ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
8533                                ;
8534                                ;*****************************************************************************
8535
8536  027300                        TO4:
8537  027300  012742  000646               MOV     #646,-(R2)      ;MOVE TO MAILBOX #  *******  646  *******
8538  027304  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8539  027306  000000                       HALT                    ;TRAPPED THRU LOC. 4
8540  027310                        TO10:
8541  027310  012742  000647               MOV     #647,-(R2)      ;MOVE TO MAILBOX #  *******  647  *******
8542  027314  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8543  027316  000000                       HALT                    ;TRAPPED THRU LOC. 10
8544  027320                        TO14:
8545  027320  012742  000650               MOV     #650,-(R2)      ;MOVE TO MAILBOX #  *******  650  *******
8546  027324  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
8547  027326  000000                        HALT                    ;TRAPPED THRU LOC. 14
8548  027330                        T030:
8549  027330  012742  000651                MOV     #651,-(R2)      ;MOVE TO MAILBOX #  *******  651  *******
8550  027334  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8551  027336  000000                        HALT                    ;TRAPPED THRU LOC. 30
8552  027340                        T034:
8553  027340  012742  000652                MOV     #652,-(R2)      ;MOVE TO MAILBOX #  *******  652  *******
8554  027344  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8555  027346  000000                        HALT                    ;TRAPPED THRU LOC. 34
8556  027350                        T0114:
8557  027350  012742  000653                MOV     #653,-(R2)      ;MOVE TO MAILBOX #  *******  653  *******
8558  027354  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8559  027356  000000                        HALT                    ;TRAPPED THRU LOC. 114
8560  027360                        T0244:
8561  027360  012742  000654                MOV     #654,-(R2)      ;MOVE TO MAILBOX #  *******  654  *******
8562  027364  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8563  027366  000000                        HALT                    ;TRAPPED THRU LOC. 244
8564  027370                        T0250:
8565  027370  012742  000655                MOV     #655,-(R2)      ;MOVE TO MAILBOX #  *******  655  *******
8566  027374  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8567  027376  000000                        HALT                    ;TRAPPED THRU LOC. 250
8568                                .SBTTL ** STARTING OF TRAP TEST **
8569  027400                        SECPRT:
```

```
8570
8571            000000                  .REPT 0
8572
8573
8574                                    PART TWO:               F11 TRAP TEST, THIS IS THE SECOND
8575                                                            PART OF THE MAIN PROGRAM.
8576
8577
8578                                    ABSTRACT
8579
8580                                    THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE
8581                                    TRAPS.  ALSO TESTED ARE TRAP OVERFLOW CONDITIONS,
8582                                    ODDITIES OF REGISTER 6, INTERRUPTS , THE RESET AND WAIT INSTRUCTIONS.
8583
8584                                    .ENDR
8585
8586                                    .LIST ME
8587                                    .NLIST MC,MD,CND
8588                                    .ABS
8589            000006                  SP %6
8590            000006                  R6-%6
8591            000003                  TAB=%3
8592            000001                  LAST-%1
8593            000005                  FIRST %5
8594            000002                  R2-%2
8595            000000                  HLT-HALT
8596            000003                  TRT-3
8597            000004                  ITRAP5 4
8598            000004                  RTRAP5 4                ;RESERVED INST AND ILLEGAL ADDRESSES
8599            000014                  RTRAP4=14               ;FOR TRACE TRAP
8600            000030                  RTRAP3 30               ;FOR EMULATOR TRAP
8601            000020                  RTRAP2-20               ;FOR IOT TRAP
8602            000034                  RTRAP1=34               ;FOR TRAP INST
8603            177564                  TTCSR=177564
8604            177560                  TRCSR-177560
3605            177564                  TPS=177564
8606            177566                  TPB=177566
8607            000240                  BELL=240
8608            000240                  NOP=240
8609            177776                  STATUS=177776
8610            000077                  TRAPA-77
8611            000010                  RTRAP=10
8612            004700                  ILLA-004700
8613            000100                  ILLB-100
8614            177776                  CC=177776
8615            001000                  BUFF=STBOT
8616
8617
8618
```

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

027400  000167  000024                          JMP     TESTN1
027404  000000                          K1:     0
027406  000000                          K2:     0
027410  000000                          K3:     0
027412  000000                          K4:     0
027414  000000                          K5:     0
027416  000000                          K6:     0
027420  052525                          K7:     052525
027422  052400                          K10:    052400
027424  000000                          K11:    0
027426  000000                          K12:    0

027430                                  TESTN1:
                                        ;*****************************************************************************************
                                        ;TEST 300       TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
                                        ;*****************************************************************************************
027430  005212                          TS300:  INC     (R2)            ;UPDATE TEST NUMBER
027432  022712  000300                          CMP     #300,(R2)       ;SEQUENCE ERROR?
027436  001127                          BNE     TS301-10        ;BR TO ERROR HALT ON SEQ ERROR
027440  005006                          CLR     %6
027442  112667  150332                  MOVB    (6)+,HERE       ;SIX SHOULD INCREMENT BY TWO
027446  020627  000002                  CMP     %6,#2
027452  001404                          BEQ     BR1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
027454  012742  000656                  MOV     #656,-(R2)      ;MOVE TO MAILBOX #  *******  656  *******
027460  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
027462  000000                          HALT                    ;R6 DID NOT AUTO INCREMENT BY TWO

027464  012706  001000          BR1:    MOV     #1000,%6
027470  114627  000000                  MOVB    -(6),#HERE      ;SHOULD DECREMENT BY TWO
027474  020627  000776                  CMP     %6,#776
027500  001404                          BEQ     BR2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
                                                                ;           CONDITIONAL BRANCH INST. AND   <-
                                                                ;           REPLACE THE MOVE INSTRUCTION   <-
                                                                ;           WHICH FOLLOWS W/ 756           <
027502  012742  000657                  MOV     #657,-(R2)      ;MOVE TO MAILBOX #  *******  657  *******
027506  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
027510  000000                          HALT                    ;R6 DID NOT AUTO DECREMENT BY 2

027512  005006          BR2:    CLR     %6
027514  112626                  MOVB    (6)+,(6)+       ;DOUBLES AUTO INCREMENT OF R6
027516  020627  000004                  CMP     %6,#4
027522  001404                          BEQ     BR3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
                                                                ;           CONDITIONAL BRANCH INST. AND   <
                                                                ;           REPLACE THE MOVE INSTRUCTION   <
                                                                ;           WHICH FOLLOWS W/ 745           <
027524  012742  000660                  MOV     #660,-(R2)      ;MOVE TO MAILBOX #  *******  660  *******
```

```
8675  027530  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8676  027532  000000                      HALT                    ;WRONG AUTO INCREMENT OF R6
8677
8678  027534  005006              BR3:    CLR     %6
8679  027536  005004                      CLR     %4
8680  027540  122624                      CMPB    (6)+,(4)+       ;TEST INCREMENT OF R6
8681  027542  020627  000002              CMP     %6,#2
8682  027546  001404                      BEQ     BR4
8683                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8684                                                              ;           CONDITIONAL BRANCH INST. AND   <====
8685                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
8686                                                              ;           WHICH FOLLOWS W/ 733           <====
8687  027550  012742  000661              MOV     #661,-(R2)      ;MOVE TO MAILBOX #  *******  661  *******
8688  027554  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8689  027556  000000                      HALT                    ;WRONG INCREMENT OF R6
8690
8691  027560  005006              BR4:    CLR     %6
8692  027562  005004                      CLR     %4
8693  027564  122426                      CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
8694  027566  020627  000002              CMP     %6,#2
8695  027572  001404                      BEQ     BR5
8696                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8697                                                              ;           CONDITIONAL BRANCH INST. AND   <====
8698                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
8699                                                              ;           WHICH FOLLOWS W/ 721           <====
8700  027574  012742  000662              MOV     #662,-(R2)      ;MOVE TO MAILBOX #  *******  662  *******
8701  027600  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8702  027602  000000                      HALT                    ;WRONG INCREMENT OF R6
8703
8704  027604  005006              BR5:    CLR     %6
8705  027606  005004                      CLR     %4
8706  027610  122624                      CMPB    (6)+,(4)+       ;TEST INCREMENT OF R4
8707  027612  020427  000001              CMP     %4,#1
8708  027616  001404                      BEQ     BR6
8709                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8710                                                              ;           CONDITIONAL BRANCH INST. AND   <====
8711                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
8712                                                              ;           WHICH FOLLOWS W/ 707           <====
8713  027620  012742  000663              MOV     #663,-(R2)      ;MOVE TO MAILBOX #  *******  663  *******
8714  027624  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8715  027626  000000                      HALT                    ;WRONG INCREMENT OF R4
8716  027630  005006              BR6:    CLR     %6
8717  027632  005004                      CLR     %4
8718  027634  122426                      CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
8719  027636  020627  000002              CMP     %6,#2
8720  027642  001404                      BEQ     BR7
8721                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8722                                                              ;           CONDITIONAL BRANCH INST. AND   <====
8723                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
8724                                                              ;           WHICH FOLLOWS W/ 675           <====
8725  027644  012742  000664              MOV     #664,-(R2)      ;MOVE TO MAILBOX #  *******  664  *******
8726  027650  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8727  027652  000000                      HALT                    ;WRONG INCREMENT OF R6
8728
8729  027654  005006              BR7:    CLR     %6
8730  027656  005004                      CLR     %4
```

```
8731  027660  122426                          CMPB    (4)+,(6)+       ;TEST INCREMENT OF R4
8732  027662  020427  000001                  CMP     %4,#1
8733  027666  001404                          BEQ     BR10
8734                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8735                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8736                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8737                                                                  ;           WHICH FOLLOWS W/ 663           <====
8738  027670  012742  000665                  MOV     #665,-(R2)      ;MOVE TO MAILBOX #  *******  665  *******
8739  027674  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8740  027676  000000                          HALT                    ;WRONG INCREMENT OF R4
8741
8742  027700  012706  001000          BR10:   MOV     #1000,%6
8743  027704  124627  000000                  CMPB    -(6),#HERE      ;TEST DECREMENT OF R6
8744  027710  022706  000776                  CMP     #776,%6
8745  027714  001404                          BEQ     TS301
8746                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8747                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8748                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8749                                                                  ;           WHICH FOLLOWS W/ 650           <====
8750  027716  012742  000666                  MOV     #666,-(R2)      ;MOVE TO MAILBOX #  *******  666  *******
8751  027722  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8752  027724  000000                          HALT                    ;WRONG DECREMENT OF R6,OR WRONG $TSTNM
8753                                                                  ;  OR SEQUENCE ERROR
8754                                  ;*****************************************************************************
8755                                  ;TEST 301       TEST TRANSFER OF .BYTE USING R6
8756                                  ;*****************************************************************************
8757  027726  005212                  TS301:  INC     (R2)            ;UPDATE TEST NUMBER
8758  027730  022712  000301                  CMP     #301,(R2)       ;SEQUENCE ERROR?
8759  027734  001133                          BNE     TS302-10        ;BR TO ERROR HALT ON SEQ ERROR
8760  027736  012767  123456  177450          MOV     #123456,K5
8761  027744  012767  050505  177432          MOV     #050505,K1
8762  027752  012705  027404                  MOV     #K1,%5          ;%5=(050505)K1
8763  027756  012706  027414                  MOV     #K5,%6          ;%6=(123456)K5
8764  027762  112625                          MOVB    (6)+,(5)+       ;LOW .BYTE OF R6 TO R5
8765  027764  022767  050456  177412          CMP     #050456,K1
8766  027772  001404                          BEQ     BR11
8767                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8768                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8769                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8770                                                                  ;           WHICH FOLLOWS W/ 760           <====
8771  027774  012742  000667                  MOV     #667,-(R2)      ;MOVE TO MAILBOX #  *******  667  *******
8772  030000  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8773  030002  000000                          HALT                    ;FALSE TRANSFER OF .BYTE
8774
8775  030004  012767  123456  177402  BR11:   MOV     #123456,K5
8776  030012  012767  050505  177364          MOV     #050505,K1
8777  030020  012705  027404                  MOV     #K1,%5          ;%5(050505)K1
8778  030024  012706  027416                  MOV     #K6,%6          ;%6(123456)K5
8779  030030  114625                          MOVB    -(6),(5)+       ;LOW .BYTE OF R6 TO R5 (DECREMENT)
8780  030032  026727  177346  050456          CMP     K1,#050456
8781  030040  001404                          BEQ     BR12
8782                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8783                                                                  ;           CONDITIONAL BRANCH INST. AND   <==
8784                                                                  ;           REPLACE THE MOVE INSTRUCTION   <==
8785                                                                  ;           WHICH FOLLOWS W/ 735           <==
8786  030042  012742  000670                  MOV     #670,-(R2)      ;MOVE TO MAILBOX #  *******  670  *******
```

```
 8787  030046  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8788  030050  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
 8789
 8790  030052  012767  123456  177324  BR12:   MOV     #123456,K1
 8791  030060  012767  050505  177326          MOV     #050505,K5
 8792  030066  012705  027404                  MOV     #K1,%5          ;(123456)
 8793  030072  012706  027414                  MOV     #K5,%6          ;(050505)
 8794  030076  112526                          MOVB    (5)+,(6)+       ;LOW OF R5 TO LOW OF R6
 8795  030100  022767  050456  177306          CMP     #050456,K5
 8796  030106  001404                          BEQ     BR13
 8797                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8798                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 8799                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 8800                                                                  ;           WHICH FOLLOWS W/ 712           <====
 8801  030110  012742  000671                  MOV     #671,-(R2)      ;MOVE TO MAILBOX #  *******  671  *******
 8802  030114  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8803  030116  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
 8804
 8805  030120  012767  123456  177256  BR13:   MOV     #123456,K1
 8806  030126  012767  050505  177260          MOV     #050505,K5
 8807  030134  012705  027405                  MOV     #K1+1,%5        ;123456
 8808  030140  012706  027414                  MOV     #K5,%6          ;050505
 8809  030144  112526                          MOVB    (5)+,(6)+       ;HIGH OF R5 TO LOW OF R6
 8810  030146  026727  177242  050647          CMP     K5,#050647
 8811  030154  001404                          BEQ     BR14
 8812                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8813                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 8814                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 8815                                                                  ;           WHICH FOLLOWS W/ 667           <====
 8816  030156  012742  000672                  MOV     #672,-(R2)      ;MOVE TO MAILBOX #  *******  672  *******
 8817  030162  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8818  030164  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
 8819
 8820  030166  012767  123456  177210  BR14:   MOV     #123456,K1
 8821  030174  012767  050505  177212          MOV     #050505,K5
 8822  030202  012705  027405                  MOV     #K1+1,%5        ;R5-123456-ODD ADDRESS
 8823  030206  012706  027414                  MOV     #K5,%6          ;R6-050505--.EVEN ADDRESS
 8824  030212  112625                          MOVB    (6)+,(5)+       ;LOW OF R6 TO HIGH OF R5
 8825  030214  022767  042456  177162          CMP     #042456,K1
 8826  030222  001404                          BEQ     TS302
 8827                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8828                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 8829                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 8830                                                                  ;           WHICH FOLLOWS W/ 644           <====
 8831  030224  012742  000673                  MOV     #673,-(R2)      ;MOVE TO MAILBOX #  *******  673  *******
 8832  030230  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8833  030232  000000                          HALT                    ;FAILED LOW OF 6 TO HIGH OF 5.OR WRONG $TSTNM
 8834                                                                  ;  OR SEQUENCE ERROR
 8835                                  ;*****************************************************************************
 8836                                  ;TEST 302       TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS
 8837                                  ;*****************************************************************************
 8838  030234  005212                  TS302:  INC     (R2)            ;UPDATE TEST NUMBER
 8839  030236  022712  000302                  CMP     #302,(R2)       ;SEQUENCE ERROR?
 8840  030242  001074                          BNE     TS303-10        ;BR TO ERROR HALT ON SEQ ERROR
 8841  030244  126767  177150  177147          CMPB    K7,K7+1         ;SAME .WORD LOW TO HIGH
 8842  030252  001404                          BEQ     BR15
```

```
 8843                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- --
 8844                                                          ;           CONDITIONAL BRANCH INST. AND   <--==
 8845                                                          ;           REPLACE THE MOVE INSTRUCTION   <- --
 8846                                                          ;           WHICH FOLLOWS W/ 773           <-= =
 8847  030254  012742  000674                MOV     #674,-(R2)      ;MOVE TO MAILBOX #  *******  674  *******
 8848  030260  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8849  030262  000000                        HALT                    ;SHOULD COMPARE LOW TO HIGH
 8850
 8851  030264  126767  177131  177126 BR15:  CMPB    K7+1,K7         ;COMPARE ODD TO .EVEN SAME .WORD
 8852  030272  001404                        BEQ     BR16
 8853                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=- -
 8854                                                          ;           CONDITIONAL BRANCH INST. AND   < =
 8855                                                          ;           REPLACE THE MOVE INSTRUCTION   <  ==
 8856                                                          ;           WHICH FOLLOWS W/ 763           < - -
 8857  030274  012742  000675                MOV     #675,-(R2)      ;MOVE TO MAILBOX #  *******  675  *******
 8858  030300  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8859  030302  000000                        HALT                    ;ODD TO .EVEN .BYTE FAILURE
 8860
 8861  030304  126767  177113  177106 BR16:  CMPB    K10+1,K7        ;SEQUENTIAL .BYTES
 8862  030312  001404                        BEQ     BR17
 8863                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8864                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 8865                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 8866                                                          ;           WHICH FOLLOWS W/ 753           <====
 8867  030314  012742  000676                MOV     #676,-(R2)      ;MOVE TO MAILBOX #  *******  676  *******
 8868  030320  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8869  030322  000000                        HALT                    ;ODD TO .EVEN FAILED
 8870
 8871  030324  126767  177072  177064 BR17:  CMPB    K10,K6
 8872  030332  001404                        BEQ     BR20
 8873                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8874                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 8875                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 8876                                                          ;           WHICH FOLLOWS W/ 743           <====
 8877  030334  012742  000677                MOV     #677,-(R2)      ;MOVE TO MAILBOX #  *******  677  *******
 8878  030340  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8879  030342  000000                        HALT                    ;.EVEN TO EVEN FAILED
 8880  030344  126767  177051  177051 BR20:  CMPB    K7+1,K10+1
 8881  030352  001404                        BEQ     BR21
 8882                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8883                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 8884                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 8885                                                          ;           WHICH FOLLOWS W/ 733           <====
 8886  030354  012742  000700                MOV     #700,-(R2)      ;MOVE TO MAILBOX #  *******  700  *******
 8887  030360  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8888  030362  000000                        HALT                    ;ODD TO ODD FAILED
 8889
 8890  030364  126767  177032  177031 BR21:  CMPB    K10,K10+1
 8891  030372  001004                        BNE     BR22
 8892                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 8893                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 8894                                                          ;           RFPLACE THE MOVE INSTRUCTION   <====
 8895                                                          ;           WHICH FOLLOWS W/ 723           <====
 8896  030374  012742  000701                MOV     #701,-(R2)      ;MOVE TO MAILBOX #  *******  701  *******
 8897  030400  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8898  030402  000000                        HALT                    ;LOW TO HIGH IN SAME .WORD FAILED
```

```
8899
8900  030404  126767  177013  177011  BR22:   CMPB    K10+1,K10+1
8901  030412  001404                          BEQ     BR23
8902                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8903                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8904                                                    ;           REPLACE THE MOVE INSTRUCTION   <=-==
8905                                                    ;           WHICH FOLLOWS W/ 713           <====
8906  030414  012742  000702                          MOV     #702,-(R2)      ;MOVE TO MAILBOX #  *******  702  *******
8907  030420  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8908  030422  000000                          HALT                    ;HIGH TO LOW IN SAME .WORD FAILED
8909
8910  030424  126767  176772  176767  BR23:   CMPB    K10,K7+1
8911  030432  001004                          BNE     TS303
8912                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8913                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8914                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8915                                                    ;           WHICH FOLLOWS W/ 703           <====
8916  030434  012742  000703                          MOV     #703,-(R2)      ;MOVE TO MAILBOX #  *******  703  *******
8917  030440  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8918  030442  000000                          HALT                    ;.EVEN TO ODD FAILED,OR WRONG $TSTNM
8919                                                                    ;  OR SEQUENCE ERROR
8920
8921
8922                                    ;*****************************************************************************************
8923                                    ;TEST 303       TEST THE CC BITS
8924                                    ;*****************************************************************************************
8925  030444  005212                    TS303:  INC     (R2)            ;UPDATE TEST NUMBER
8926  030446  022712  000303                    CMP     #303,(R2)       ;SEQUENCE ERROR?
8927  030452  001053                            BNE     TS304-10        ;BR TO ERROR HALT ON SEQ ERROR
8928  030454  000277                            SCC                     ;SET STATUS
8929  030456  005067  147314                    CLR     STATUS          ;CLEAR STATUS
8930  030462  103004                            BCC     BR33
8931                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-==
8932                                                    ;           CONDITIONAL BRANCH INST. AND   <- =-
8933                                                    ;           REPLACE THE MOVE INSTRUCTION   <  --
8934                                                    ;           WHICH FOLLOWS W/ 773           <=-=-
8935  030464  012742  000704                          MOV     #704,-(R2)      ;MOVE TO MAILBOX #  *******  704  *******
8936  030470  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8937  030472  000000                          HALT                    ;C NOT CLEAR
8938  030474                          BR33:
8939  030474  102004                          BVC     BR34
8940                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- --
8941                                                    ;           CONDITIONAL BRANCH INST. AND   <- =-
8942                                                    ;           REPLACE THE MOVE INSTRUCTION   <=-=
8943                                                    ;           WHICH FOLLOWS W/ 766           <---
8944  030476  012742  000705                          MOV     #705,-(R2)      ;MOVE TO MAILBOX #  *******  705  *******
8945  030502  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8946  030504  000000                          HALT                    ;V NOT CLEAR
8947  030506                          BR34:
8948  030506  001004                          BNE     BR35
8949                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8950                                                    ;           CONDITIONAL BRANCH INST. AND   <=-==
8951                                                    ;           REPLACE THE MOVE INSTRUCTION   <=---=
8952                                                    ;           WHICH FOLLOWS W/ 761           <=- -
8953  030510  012742  000706                          MOV     #706,-(R2)      ;MOVE TO MAILBOX #  *******  706  *******
8954  030514  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
8955  030516  000000                          HALT                    ;Z NOT CLEAR
8956  030520                          BR35:
8957  030520  100004                          BPL     BR36
8958                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8959                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8960                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8961                                                                  ;           WHICH FOLLOWS W/ 754           <====
8962  030522  012742  000707                MOV     #707,-(R2)      ;MOVE TO MAILBOX #  *******  707  *******
8963  030526  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8964  030530  000000                        HALT                    ;N NOT CLEAR
8965  030532  000257                  BR36: CCC                     ;CLEAR CONDITION CODES
8966  030534  052767  000017  147234        BIS     #17,STATUS      ;SET STATUS TO ONES
8967
8968  030542  103404                        BCS     BR37
8969                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8970                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8971                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8972                                                                  ;           WHICH FOLLOWS W/ 743           <====
8973  030544  012742  000710                MOV     #710,-(R2)      ;MOVE TO MAILBOX #  *******  710  *******
8974  030550  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8975  030552  000000                        HALT                    ;C NOT SET
8976  030554                          BR37:
8977  030554  102404                        BVS     BR40
8978                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8979                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8980                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8981                                                                  ;           WHICH FOLLOWS W/ 736           <====
8982  030556  012742  000711                MOV     #711,-(R2)      ;MOVE TO MAILBOX #  *******  711  *******
8983  030562  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8984  030564  000000                        HALT                    ;V NOT SET
8985  030566                          BR40:
8986  030566  001404                        BEQ     BR41
8987                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8988                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8989                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8990                                                                  ;           WHICH FOLLOWS W/ 731           <====
8991  030570  012742  000712                MOV     #712,-(R2)      ;MOVE TO MAILBOX #  *******  712  *******
8992  030574  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8993  030576  000000                        HALT                    ;Z NOT SET
8994  030600                          BR41:
8995  030600  100404                        BMI     TS304
8996                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8997                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8998                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8999                                                                  ;           WHICH FOLLOWS W/ 724           <====
9000  030602  012742  000713                MOV     #713,-(R2)      ;MOVE TO MAILBOX #  *******  713  *******
9001  030606  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9002  030610  000000                        HALT                    ;N NOT SET,OR WRONG $TSTNM
9003                                                                  ;  OR SEQUENCE ERROR
9004  ;*****************************************************************************************
9005  ;TEST 304       TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION
9006  ;*****************************************************************************************
9007  030612  005212                  TS304: INC     (R2)            ;UPDATE TEST NUMBER
9008  030614  022712  000304                CMP     #304,(R2)       ;SEQUENCE ERROR?
9009  030620  001006                        BNE     RETA            ;BR TO ERROR HALT ON SEQ ERROR
9010  030622  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
```

```
9011  030626  012767  030646  147154          MOV     #RETAH,RTRAP    ;RETURN LOCATION
9012  030634  000077                          TRAPA                   ;RESERVED INSTRUCTION, SHOULD TRAP
9013  030636                          RETA:
9014  030636  012742  000714                  MOV     #714,-(R2)      ;MOVE TO MAILBOX #  *******  714  *******
9015  030642  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9016  030644  000000                          HALT                    ;RESERVE INSTRUCTION DIDN'T TRAP,OR WRONG $TSTNM
9017  030646                          RETAH:
9018                                  ;*****************************************************************************************
9019                                  ;TEST 305       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9020                                  ;*****************************************************************************************
9021  030646  005212                  TS305:  INC     (R2)            ;UPDATE TEST NUMBER
9022  030650  022712  000305                  CMP     #305,(R2)       ;SEQUENCE ERROR?
9023  030654  001011                          BNE     TS306-10        ;BR TO ERROR HALT ON SEQ ERROR
9024  030656  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
9025  030662  012767  030672  147120          MOV     #RETB,RTRAP     ;RETURN POINTER
9026  030670  000077                          TRAPA                   ;RESERVED INSTRUCTION
9027  030672  020627  000774          RETB:   CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
9028  030676  001404                          BEQ     TS306
9029                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-
9030                                                                  ;           CONDITIONAL BRANCH INST. AND   <==--.
9031                                                                  ;           REPLACE THE MOVE INSTRUCTION   <---=
9032                                                                  ;           WHICH FOLLOWS W/ 766           <===
9033  030700  012742  000715                  MOV     #715,-(R2)      ;MOVE TO MAILBOX #  *******  715  *******
9034  030704  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9035  030706  000000                          HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
9036                                                                  ;  OR SEQUENCE ERROR
9037                                  ;*****************************************************************************************
9038                                  ;TEST 306       TEST THAT PROPER P.C. IS SAVED
9039                                  ;*****************************************************************************************
9040  030710  005212                  TS306:  INC     (R2)            ;UPDATE TEST NUMBER
9041  030712  022712  000306                  CMP     #306,(R2)       ;SEQUENCE ERROR?
9042  030716  001012                          BNE     TS307-10        ;BR TO ERROR HALT ON SEQ ERROR
9043  030720  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
9044  030724  012767  030734  147056          MOV     #RETC,RTRAP     ;RETURN FROM TRAP POINTER
9045  030732  000077                  INSTC:  TRAPA                   ;TRAP ON THIS INSTRUCTION
9046  030734  022767  030734  150032  RETC:   CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
9047  030742  001404                          BEQ     TS307
9048                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < ---
9049                                                                  ;           CONDITIONAL BRANCH INST. AND   <-===
9050                                                                  ;           REPLACE THE MOVE INSTRUCTION   <=-=-
9051                                                                  ;           WHICH FOLLOWS W/ 765           <- ==
9052  030744  012742  000716                  MOV     #716,-(R2)      ;MOVE TO MAILBOX #  *******  716  *******
9053  030750  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9054  030752  000000                          HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
9055                                                                  ;  OR SEQUENCE ERROR
9056                                  ;*****************************************************************************************
9057                                  ;TEST 307       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
9058                                  ;*****************************************************************************************
9059  030754  005212                  TS307:  INC     (R2)            ;UPDATE TEST NUMBER
9060  030756  022712  000307                  CMP     #307,(R2)       ;SEQUENCE ERROR?
9061  030762  001037                          BNE     TS310-10        ;BR TO ERROR HALT ON SEQ ERROR
9062  030764  012706  001000                  MOV     #BUFF,SP        ;SET UP
9063  030770  012767  031006  147012          MOV     #RETD,RTRAP     ;SET UP
9064  030776  005067  146774                  CLR     CC              ;CLEAR CC AND PRIORITY
9065  031002  000257                          CCC
9066  031004  000077                          TRAPA                   ;TRAP
```

```
9067  031006  026727  147764  000000  RETD:   CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
9068  031014  001404                  BEQ     1$
9069                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <===-=
9070                                            ;           CONDITIONAL BRANCH INST. AND   <  --
9071                                            ;           REPLACE THE MOVE INSTRUCTION   <---=
9072                                            ;           WHICH FOLLOWS W/ 762           <--
9073  031016  012742  000717                  MOV     #717,-(R2)      ;MOVE TO MAILBOX #  *******  717  *******
9074  031022  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9075  031024  000000                  HALT                    ;INCORRECT STATUS
9076  031026  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
9077  031032  012767  031052  146750          MOV     #RETE,RTRAP     ;SET UP
9078  031040  012767  000357  146730          MOV     #357,CC         ;SET PRIORITY
9079  031046  000277                  SCC                     ;SET CC
9080  031050  000077                  TRAPA                   ;TRAP
9081  031052  026727  147720  000357  RETE:   CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
9082  031060  001404                  BEQ     TS310
9083                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=
9084                                            ;           CONDITIONAL BRANCH INST. AND   <- =
9085                                            ;           REPLACE THE MOVE INSTRUCTION   <- =-
9086                                            ;           WHICH FOLLOWS W/ 740           <=-=-
9087  031062  012742  000720                  MOV     #720,-(R2)      ;MOVE TO MAILBOX #  *******  720  *******
9088  031066  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9089  031070  000000                  HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9090                                            ;  OR SEQUENCE ERROR
9091                    ;*****************************************************************************
9092                    ;TEST 310       TEST THAT 'NEW' STATUS IS CORRECT
9093                    ;*****************************************************************************
9094  031072  005212          TS310:  INC     (R2)            ;UPDATE TEST NUMBER
9095  031074  022712  000310          CMP     #310,(R2)       ;SEQUENCE ERROR?
9096  031100  001110                  BNE     STPP            ;BR TO ERROR HALT ON SEQ ERROR
9097  031102  012706  001000          MOV     #BUFF,SP
9098  031106  012767  031122  146674  MOV     #RETF,RTRAP
9099  031114  005067  146672          CLR     RTRAP+2         ;CLEAR FUTURE PRIORITY AND CC
9100  031120  000077                  TRAPA
9101  031122                  RETF:                           ;TEST FOR "C" CLEARED
9102  031122  100004                  BPL     1$
9103                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
9104                                            ;           CONDITIONAL BRANCH INST. AND   <====
9105                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9106                                            ;           WHICH FOLLOWS W/ 766           <====
9107  031124  012742  000721                  MOV     #721,-(R2)      ;MOVE TO MAILBOX #  *******  721  *******
9108  031130  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9109  031132  000000                  HALT                    ;N NOT CLEARED
9110  031134                  1$:
9111  031134  001004                  BNE     2$
9112                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9113                                            ;           CONDITIONAL BRANCH INST. AND   <====
9114                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9115                                            ;           WHICH FOLLOWS W/ 761           <====
9116  031136  012742  000722                  MOV     #722,-(R2)      ;MOVE TO MAILBOX #  *******  722  *******
9117  031142  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9118  031144  000000                  HALT                    ;Z NOT CLEARED
9119  031146                  2$:
9120  031146  102004                  BVC     3$
9121                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9122                                            ;           CONDITIONAL BRANCH INST. AND   <====
```

```
9123                                                                    ;            REPLACE THE MOVE INSTRUCTION   <
9124                                                                    ;            WHICH FOLLOWS W/ 754           <
9125  031150  012742  000723                MOV     #723,-(R2)       ;MOVE TO MAILBOX #  *******  723  *******
9126  031154  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9127  031156  000000                        HALT                     ;V NOT CLEARED
9128  031160                          3$:
9129  031160  103004                        BCC     4$
9130                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
9131                                                                    ;            CONDITIONAL BRANCH INST. AND   <---
9132                                                                    ;            REPLACE THE MOVE INSTRUCTION   <-
9133                                                                    ;            WHICH FOLLOWS W/ 747           <
9134  031162  012742  000724                MOV     #724,-(R2)       ;MOVE TO MAILBOX #  *******  724  *******
9135  031166  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9136  031170  000000                        HALT                     ;C NOT CLEARED
9137  031172  032767  000340  146576  4$:   BIT     #340,CC          ;TEST PRIORITY
9138  031200  001404                        BEQ     5$
9139                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
9140                                                                    ;            CONDITIONAL BRANCH INST. AND   <-
9141                                                                    ;            REPLACE THE MOVE INSTRUCTION   <====
9142                                                                    ;            WHICH FOLLOWS W/ 737           <====
9143  031202  012742  000725                MOV     #725,-(R2)       ;MOVE TO MAILBOX #  *******  725  *******
9144  031206  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9145  031210  000000                        HALT                     ;PRIORITY NOT ZERO
9146  031212  012706  001000          5$:   MOV     #BUFF,SP
9147  031216  012767  031234  146564        MOV     #RETG,RTRAP
9148  031224  012767  000357  146560        MOV     #357,RTRAP+2     ;SET NEW ''CC'' AND PRIORITY
9149  031232  000077                        TRAPA                    ;TRAP HERE
9150  031234                          RETG:
9151  031234  100404                        BMI     1$
9152                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9153                                                                    ;            CONDITIONAL BRANCH INST. AND   <---
9154                                                                    ;            REPLACE THE MOVE INSTRUCTION   <-
9155                                                                    ;            WHICH FOLLOWS W/ 721           <
9156  031236  012742  000726                MOV     #726,-(R2)       ;MOVE TO MAILBOX #  *******  726  *******
9157  031242  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9158  031244  000000                        HALT                     ;N NOT SET
9159  031246                          1$:
9160  031246  001404                        BEQ     2$
9161                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9162                                                                    ;            CONDITIONAL BRANCH INST. AND   <==
9163                                                                    ;            REPLACE THE MOVE INSTRUCTION   <=-
9164                                                                    ;            WHICH FOLLOWS W/ 714           <
9165  031250  012742  000727                MOV     #727,-(R2)       ;MOVE TO MAILBOX #  *******  727  *******
9166  031254  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9167  031256  000000                        HALT                     ;Z NOT SET
9168  031260                          2$:
9169  031260  102404                        BVS     3$
9170                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=---
9171                                                                    ;            CONDITIONAL BRANCH INST. AND   <-
9172                                                                    ;            REPLACE THE MOVE INSTRUCTION   <--
9173                                                                    ;            WHICH FOLLOWS W/ 707           <
9174  031262  012742  000730                MOV     #730,-(R2)       ;MOVE TO MAILBOX #  *******  730  *******
9175  031266  005242                        INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
9176  031270  000000                        HALT                     ;V NOT SET
9177  031272                          3$:
9178  031272  103404                        BCS     4$
```

```
9179                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=-
9180                                                    ;           CONDITIONAL BRANCH INST. AND   <--==
9181                                                    ;           REPLACE THE MOVE INSTRUCTION   <-==-
9182                                                    ;           WHICH FOLLOWS W/ 702           <=- -
9183  031274  012742  000731              MOV     #731,-(R2)      ;MOVE TO MAILBOX #  *******  731  *******
9184  031300  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9185  031302  000000                      HALT                    ;C NOT SET
9186  031304  016706  146466      4$:     MOV     CC,SP
9187  031310  042706  000017              BIC     #17,SP
9188  031314  022706  000340              CMP     #340,SP
9189  031320  001404                      BEQ     STPPA
9190                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- -=
9191                                                    ;           CONDITIONAL BRANCH INST. AND   <  -
9192                                                    ;           REPLACE THE MOVE INSTRUCTION   <
9193                                                    ;           WHICH FOLLOWS W/ 667           <
9194  031322                      STPP:
9195  031322  012742  000732              MOV     #732,-(R2)      ;MOVE TO MAILBOX #  *******  732  *******
9196  031326  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9197  031330  000000                      HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9198  031332  012767  000012  146450  STPPA:  MOV     #12,10
9199  031340  005067  146446              CLR     12
9200                      ;*****************************************************************************
9201                      ;TEST 311       TEST THAT A TRAP OCCURS FOR A ''TRAP'' INSTRUCTION
9202                      ;*****************************************************************************
9203  031344  005212      TS311:  INC     (R2)            ;UPDATE TEST NUMBER
9204  031346  022712  000311              CMP     #311,(R2)       ;SEQUENCE ERROR?
9205  031352  001013                      BNE     TS312-10        ;BR TO ERROR HALT ON SEQ ERROR
9206  031354  012767  000012  146426      MOV     #12,10
9207  031362  005067  146424              CLR     12
9208  031366  012706  001000              MOV     #BUFF,SP        ;STACK POINTER SETUP
9209  031372  012767  031412  146434      MOV     #RETA1,RTRAP1   ;RETURN LOCATION
9210  031400  104400                      TRAP                    ;RESERVED INSTRUCTION, SHOULD TRAP
9211  031402  012742  000733              MOV     #733,-(R2)      ;MOVE TO MAILBOX #  *******  733  *******
9212  031406  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9213  031410  000000                      HALT                    ;TRAP DIDN'T TRAP,OR WRONG $TSTNM
9214  031412                      RETA1:
9215                      ;*****************************************************************************
9216                      ;TEST 312       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9217                      ;*****************************************************************************
9218  031412  005212      TS312:  INC     (R2)            ;UPDATE TEST NUMBER
9219  031414  022712  000312              CMP     #312,(R2)       ;SEQUENCE ERROR?
9220  031420  001011                      BNE     TS313-10        ;BR TO ERROR HALT ON SEQ ERROR
9221  031422  012706  001000              MOV     #BUFF,SP        ;STACK POINTER SETUP
9222  031426  012767  031436  146400      MOV     #RETB1,RTRAP1   ;RETURN POINTER
9223  031434  104400                      TRAP                    ;RESERVED INSTRUCTION
9224  031436  020627  000774      RETB1:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
9225  031442  001404                      BEQ     TS313
9226                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9227                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9228                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9229                                                    ;           WHICH FOLLOWS W/ 766           <====
9230  031444  012742  000734              MOV     #734,-(R2)      ;MOVE TO MAILBOX #  *******  734  *******
9231  031450  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9232  031452  000000                      HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
9233                                                    ;  OR SEQUENCE ERROR
9234                      ;*****************************************************************************
```

```
                                        ;TEST 313       TEST THAT PROPER P.C. IS SAVED
                                        ;*************************************************************************************
031454  005212                  TS313:  INC     (R2)            ;UPDATE TEST NUMBER
031456  022712  000313                  CMP     #313,(R2)       ;SEQUENCE ERROR?
031462  001012                          BNE     TS314-10        ;BR TO ERROR HALT ON SEQ ERROR
031464  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
031470  012767  031500  146336          MOV     #RETC1,RTRAP1   ;RETURN FROM TRAP POINTER
031476  104400                          TRAP                    ;TRAP ON THIS INSTRUCTION
031500  022767  031500  147266  RETC1:  CMP     #.,BUFF-4       ;CHECK INCREMENTED P.C.
031506  001404                          BEQ     TS314
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W' 765           <====
031510  012742  000735                  MOV     #735,-(R2)      ;MOVE TO MAILBOX #  *******  735  *******
031514  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031516  000000                          HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                        ;*************************************************************************************
                                        ;TEST 314       TEST THAT 'OLD'' CC AND PRIORITY ARE PLACED ON STACK
                                        ;*************************************************************************************
031520  005212                  TS314:  INC     (R2)            ;UPDATE TEST NUMBER
031522  022712  000314                  CMP     #314,(R2)       ;SEQUENCE ERROR?
031526  001036                          BNE     TS315-10        ;BR TO ERROR HALT ON SEQ ERROR
031530  012706  001000                  MOV     #BUFF,SP        ;SET UP
031534  012767  031552  146272          MOV     #RETD1,RTRAP1   ;SET UP
031542  005067  146230                  CLR     CC              ;CLEAR CC AND PRIORITY
031546  000257                          CCC
031550  104400                          TRAP                    ;TRAP
031552  026727  147220  000000  RETD1:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
031560  001404                          BEQ     1$
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
031562  012742  000736                  MOV     #736,-(R2)      ;MOVE TO MAILBOX #  *******  736  *******
031566  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031570  000000                          HALT                    ;INCORRECT STATUS
031572  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
031576  012767  031614  146230          MOV     #RETE1,RTRAP1   ;SET UP
031604  012767  000357  146164          MOV     #357,CC         ;SET PRIORITY
031612  104400                          TRAP                    ;SET CC
031614  026727  147156  000357  RETE1:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
031622  001404                          BEQ     TS315
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 741           <====
031624  012742  000737                  MOV     #737,-(R2)      ;MOVE TO MAILBOX #  *******  737  *******
031630  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031632  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                        ;*************************************************************************************
                                        ;TEST 315       TEST THAT 'NEW'' STATUS IS CORRECT
                                        ;*************************************************************************************
031634  005212                  TS315:  INC     (R2)            ;UPDATE TEST NUMBER
```

```
 9291  031636  022712  000315                    CMP     #315,(R2)       ;SEQUENCE ERROR?
 9292  031642  001110                            BNE     TS316-10        ;BR TO ERROR HALT ON SEQ ERROR
 9293  031644  012706  001000                    MOV     #BUFF,SP
 9294  031650  012767  031664  146156            MOV     #RETF1,RTRAP1
 9295  031656  005067  146154                    CLR     RTRAP1+2        ;CLEAR FUTURE PRIORITY AND CC
 9296  031662  104400                            TRAP
 9297  031664                          RETF1:                            ;TEST FOR "C" CLEARED
 9298  031664  100004                            BPL     1$
 9299                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9300                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9301                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9302                                                                    ;           WHICH FOLLOWS W/ 766           <====
 9303  031666  012742  000740                    MOV     #740,-(R2)      ;MOVE TO MAILBOX #  *******  740  *******
 9304  031672  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9305  031674  000000                            HALT                    ;C NOT CLEARED
 9306  031676                          1$:
 9307  031676  001004                            BNE     2$
 9308                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9309                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9310                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9311                                                                    ;           WHICH FOLLOWS W/ 761           <====
 9312  031700  012742  000741                    MOV     #741,-(R2)      ;MOVE TO MAILBOX #  *******  741  *******
 9313  031704  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9314  031706  000000                            HALT                    ;Z NOT CLEARED
 9315  031710                          2$:
 9316  031710  102004                            BVC     3$
 9317                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9318                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9319                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9320                                                                    ;           WHICH FOLLOWS W/ 754           <====
 9321  031712  012742  000742                    MOV     #742,-(R2)      ;MOVE TO MAILBOX #  *******  742  *******
 9322  031716  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9323  031720  000000                            HALT                    ;V NOT CLEARED
 9324  031722                          3$:
 9325  031722  103004                            BCC     4$
 9326                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9327                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9328                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9329                                                                    ;           WHICH FOLLOWS W/ 747           <====
 9330  031724  012742  000743                    MOV     #743,-(R2)      ;MOVE TO MAILBOX #  *******  743  *******
 9331  031730  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9332  031732  000000                            HALT                    ;C NOT CLEARED
 9333  031734  032767  000340  146034    4$:     BIT     #340,CC         ;TEST PRIORITY
 9334  031742  001404                            BEQ     5$
 9335                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9336                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9337                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9338                                                                    ;           WHICH FOLLOWS W/ 737           <====
 9339  031744  012742  000744                    MOV     #744,-(R2)      ;MOVE TO MAILBOX #  *******  744  *******
 9340  031750  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9341  031752  000000                            HALT                    ;PRIORITY NOT ZERO
 9342  031754  012706  001000            5$:     MOV     #BUFF,SP
 9343  031760  012767  031776  146046            MOV     #RETG1,RTRAP1
 9344  031766  012767  000357  146042            MOV     #357,RTRAP1+2   ;SET NEW "CC" AND PRIORITY
 9345  031774  104400                            TRAP                    ;TRAP HERE
 9346  031776                          RETG1:
```

```
9347  031776  100404                        BMI     1$
9348                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9349                                        ;           CONDITIONAL BRANCH INST. AND   <====
9350                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9351                                        ;           WHICH FOLLOWS W/ 721           <====
9352  032000  012742  000745                MOV     #745,-(R2)      ;MOVE TO MAILBOX #  *******  745  *******
9353  032004  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9354  032006  000000                        HALT                    ;N NOT SET
9355  032010                        1$:
9356  032010  001404                        BEQ     2$
9357                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9358                                        ;           CONDITIONAL BRANCH INST. AND   <====
9359                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9360                                        ;           WHICH FOLLOWS W/ 714           <====
9361  032012  012742  000746                MOV     #746,-(R2)      ;MOVE TO MAILBOX #  *******  746  *******
9362  032016  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9363  032020  000000                        HALT                    ;Z NOT SET
9364  032022                        2$:
9365  032022  102404                        BVS     3$
9366                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9367                                        ;           CONDITIONAL BRANCH INST. AND   <====
9368                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9369                                        ;           WHICH FOLLOWS W/ 707           <====
9370  032024  012742  000747                MOV     #747,-(R2)      ;MOVE TO MAILBOX #  *******  747  *******
9371  032030  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9372  032032  000000                        HALT                    ;V NOT SET
9373  032034                        3$:
9374  032034  103404                        BCS     4$
9375                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9376                                        ;           CONDITIONAL BRANCH INST. AND   <====
9377                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9378                                        ;           WHICH FOLLOWS W/ 702           <====
9379  032036  012742  000750                MOV     #750,-(R2)      ;MOVE TO MAILBOX #  *******  750  *******
9380  032042  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9381  032044  000000                        HALT                    ;C NOT SET
9382  032046  016706  145724        4$:     MOV     CC,SP
9383  032052  042706  000017                BIC     #17,SP
9384  032056  022706  000340                CMP     #340,SP
9385  032062  001404                        BEQ     TS316
9386                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9387                                        ;           CONDITIONAL BRANCH INST. AND   <====
9388                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9389                                        ;           WHICH FOLLOWS W/ 667           <====
9390  032064  012742  000751                MOV     #751,-(R2)      ;MOVE TO MAILBOX #  *******  751  *******
9391  032070  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9392  032072  000000                        HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9393                                                                ;  OR SEQUENCE ERROR
9394                          ;*************************************************************************
9395                          ;TEST 316       TEST THAT ALL COMBINATION OF 'TRAP'' WILL CAUSE A TRAP
9396                          ;*************************************************************************
9397  032074  005212        TS316:  INC     (R2)            ;UPDATE TEST NUMBER
9398  032076  022712  000316                CMP     #316,(R2,       ;SEQUENCE ERROR?
9399  032102  001011                        BNE     BR45            ;BR TO ERROR HALT ON SEQ ERROR
9400                                                                ;***** F11 **** ADD +376 TO SHORTEN TEST
9401  032104  012767  104776  000012        MOV     #TRAP+376,RB1   ;INITIALIZE BASE TRAP INSTRUCTION
9402  032112  012767  032136  145714        MOV     #RA1,34         ;RETURN FROM TRAP TO RA1
```

```
 9403  032120  012706  001000            RC1:    MOV     #BUFF,SP        ;SET UP STACK POINTER
 9404  032124  104400                    RB1:    TRAP                    ;TRAP INST WILL BE MODIFIED TO TRAP+377
 9405  032126                            BR45:
 9406  032126  012742  000752                    MOV     #752,-(R2)      ;MOVE TO MAILBOX #  *******  752  *******
 9407  032132  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9408  032134  000000                            HALT                    ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TSTNM
 9409  032136  005267  177762            RA1:    INC     RB1             ;INCREMENT TRAP INSTRUCTION
 9410  032142  022767  104777  177754            CMP     #104777,RB1     ;TRAP+377 TO UPPER LIMIT
 9411  032150  103363                            BHIS    RC1             ;HAVE WE TESTED ALL
 9412  032152  012767  000036  145654            MOV     #36,34
 9413  032160  005067  145652                    CLR     36
 9414                                    ;*****************************************************************************
 9415                                    ;TEST 317       TEST THAT A TRAP OCCURES ON AN ''IOT'' INSTRUCTION
 9416                                    ;*****************************************************************************
 9417  032164  005212                    TS317:  INC     (R2)            ;UPDATE TEST NUMBER
 9418  032166  022712  000317                    CMP     #317,(R2)       ;SEQUENCE ERROR?
 9419  032172  001006                            BNE     TS320-10        ;BR TO ERROR HALT ON SEQ ERROR
 9420  032174  012706  001000                    MOV     #BUFF,SP        ;STACK POINTER SETUP
 9421  032200  012767  032220  145612            MOV     #RETA2,RTRAP2   ;RETURN LOCATION
 9422  032206  000004                            IOT                     ;RESERVE INSTRUCTION, SHOULD TRAP
 9423  032210  012742  000753                    MOV     #753,-(R2)      ;MOVE TO MAILBOX #  *******  753  *******
 9424  032214  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9425  032216  000000                            HALT                    ;IOT DIDN'T TRAP,OR WRONG $TSTNM
 9426  032220                            RETA2:
 9427                                    ;*****************************************************************************
 9428                                    ;TEST 320       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
 9429                                    ;*****************************************************************************
 9430  032220  005212                    TS320:  INC     (R2)            ;UPDATE TEST NUMBER
 9431  032222  022712  000320                    CMP     #320,(R2)       ;SEQUENCE ERROR?
 9432  032226  001011                            BNE     TS321-10        ;BR TO ERROR HALT ON SEQ ERROR
 9433  032230  012706  001000                    MOV     #BUFF,SP        ;STACK POINTER SETUP
 9434  032234  012767  032244  145556            MOV     #RETB2,RTRAP2   ;RETURN POINTER
 9435  032242  000004                            IOT                     ;RESERVED INSTRUCTION
 9436  032244  020627  000774            RETB2:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
 9437  032250  001404                            BEQ     TS321
 9438                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <-
 9439                                                                    ;           CONDITIONAL BRANCH INST. AND    <  =-
 9440                                                                    ;           REPLACE THE MOVE INSTRUCTION    <--
 9441                                                                    ;           WHICH FOLLOWS W/ 766            <
 9442  032252  012742  000754                    MOV     #754,-(R2)      ;MOVE TO MAILBOX #  *******  754  *******
 9443  032256  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9444  032260  000000                            HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
 9445                                                                    ;  OR SEQUENCE ERROR
 9446                                    ;*****************************************************************************
 9447                                    ;TEST 321       TEST THAT PROPER P.C. IS SAVED
 9448                                    ;*****************************************************************************
 9449  032262  005212                    TS321:  INC     (R2)            ;UPDATE TEST NUMBER
 9450  032264  022712  000321                    CMP     #321,(R2)       ;SEQUENCE ERROR?
 9451  032270  001012                            BNE     TS322-10        ;BR TO ERROR HALT ON SEQ ERROR
 9452  032272  012706  001000                    MOV     #BUFF,SP        ;STACK POINTER SETUP
 9453  032276  012767  032306  145514            MOV     #RETC2,RTRAP2   ;RETURN FROM TRAP POINTER
 9454  032304  000004                            IOT                     ;TRAP ON THIS INSTRUCTION
 9455  032306  022767  032306  146460  RETC2:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
 9456  032314  001404                            BEQ     TS322
 9457                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <-
 9458                                                                    ;           CONDITIONAL BRANCH INST. AND    <  =-
```

```
9459                                                    ;           REPLACE THE MOVE INSTRUCTION   <=-=
9460                                                    ;           WHICH FOLLOWS W/ 765           <=-=
9461  032316  012742  000755              MOV     #755,-(R2)      ;MOVE TO MAILBOX #  *******  755  *******
9462  032322  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9463  032324  000000                      HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
9464                                                            ;  OR SEQUENCE ERROR
9465                                  ;*****************************************************************************
9466                                  ;TEST 322       TEST THAT 'OLD'' CC AND PRIORITY ARE PLACED ON STACK
9467                                  ;*****************************************************************************
9468  032326  005212                  TS322:  INC     (R2)            ;UPDATE TEST NUMBER
9469  032330  022712  000322                  CMP     #322,(R2)       ;SEQUENCE ERROR?
9470  032334  001037                          BNE     ^S323-10        ;BR TO ERROR HALT ON SEQ ERROR
9471  032336  012706  001000                  MOV     #BUFF,SP        ;SET UP
9472  032342  012767  032360  145450          MOV     #RETD2,RTRAP2   ;SET UP
9473  032350  005067  145422                  CLR     CC              ;CLEAR CC AND PRIORITY
9474  032354  000257                          CCC
9475  032356  000004                          IOT                     ;TRAP
9476  032360  026727  146412  000000  RETD2:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
9477  032366  001404                          BEQ     1$
9478                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9479                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9480                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9481                                                    ;           WHICH FOLLOWS W/ 762           <====
9482  032370  012742  000756              MOV     #756,-(R2)      ;MOVE TO MAILBOX #  *******  756  *******
9483  032374  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9484  032376  000000                      HALT                    ;INCORRECT STATUS
9485  032400  012706  001000          1$: MOV     #BUFF,SP        ;SET UP
9486  032404  012767  032424  145406          MOV     #RETE2,RTRAP2   ;SET UP
9487  032412  012767  000357  145356          MOV     #357,CC         ;SET PRIORITY
9488  032420  000277                          SCC                     ;SET CC
9489  032422  000004                          IOT                     ;TRAP
9490  032424  026727  146346  000357  RETE2:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
9491  032432  001404                          BEQ     TS323
9492                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9493                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9494                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9495                                                    ;           WHICH FOLLOWS W/ 740           <====
9496  032434  012742  000757              MOV     #757,-(R2)      ;MOVE TO MAILBOX #  *******  757  *******
9497  032440  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9498  032442  000000                      HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9499                                                            ;  OR SEQUENCE ERROR
9500                                  ;*****************************************************************************
9501                                  ;TEST 323       TEST THAT 'NEW'' STATUS IS CORRECT
9502                                  ;*****************************************************************************
9503  032444  005212                  TS323:  INC     (R2)            ;UPDATE TEST NUMBER
9504  032446  022712  000323                  CMP     #323,(R2)       ;SEQUENCE ERROR?
9505  032452  001110                          BNE     BR46            ;BR TO ERROR HALT ON SEQ ERROR
9506  032454  012706  001000                  MOV     #BUFF,SP
9507  032460  012767  032474  145332          MOV     #RETF2,RTRAP2
9508  032466  005067  145330                  CLR     RTRAP2+2        ;CLEAR FUTURE PRIORITY AND CC
9509  032472  000004                          IOT
9510  032474                          RETF2:                          ;TEST FOR ''C'' CLEARED
9511  032474  100004                          BPL     1$
9512                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=--
9513                                                    ;           CONDITIONAL BRANCH INST. AND   <=--
9514                                                    ;           REPLACE THE MOVE INSTRUCTION   <=--
```

```
 9515                                                    ;          WHICH FOLLOWS W/ 766          <-===
 9516  032476  012742  000760                 MOV   #760,-(R2)  ;MOVE TO MAILBOX #  *******  760  *******
 9517  032502  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9518  032504  000000                         HALT              ;N NOT CLEARED
 9519  032506                          1$:
 9520  032506  001004                         BNE   2$
 9521                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9522                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 9523                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 9524                                                    ;          WHICH FOLLOWS W/ 761           <====
 9525  032510  012742  000761                 MOV   #761,-(R2)  ;MOVE TO MAILBOX #  *******  761  *******
 9526  032514  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9527  032516  000000                         HALT              ;Z NOT CLEARED
 9528  032520                          2$:
 9529  032520  102004                         BVC   3$
 9530                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9531                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 9532                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 9533                                                    ;          WHICH FOLLOWS W/ 754           <====
 9534  032522  012742  000762                 MOV   #762,-(R2)  ;MOVE TO MAILBOX #  *******  762  *******
 9535  032526  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9536  032530  000000                         HALT              ;V NOT CLEARED
 9537  032532                          3$:
 9538  032532  103004                         BCC   4$
 9539                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9540                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 9541                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 9542                                                    ;          WHICH FOLLOWS W/ 747           <====
 9543  032534  012742  000763                 MOV   #763,-(R2)  ;MOVE TO MAILBOX #  *******  763  *******
 9544  032540  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9545  032542  000000                         HALT              ;C NOT CLEARED
 9546  032544  032767  000340  145224  4$:    BIT   #340,CC     ;TEST PRIORITY
 9547  032552  001404                         BEQ   5$
 9548                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
 9549                                                    ;          CONDITIONAL BRANCH INST. AND   <=-==
 9550                                                    ;          REPLACE THE MOVE INSTRUCTION   <  =
 9551                                                    ;          WHICH FOLLOWS W/ 737           <= --
 9552  032554  012742  000764                 MOV   #764,-(R2)  ;MOVE TO MAILBOX #  *******  764  *******
 9553  032560  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9554  032562  000000                         HALT              ;PRIORITY NOT ZERO
 9555  032564  012706  001000          5$:    MOV   #BUFF,SP
 9556  032570  012767  032606  145222         MOV   #RETG2,RTRAP2
 9557  032576  012767  000357  145216         MOV   #357,RTRAP2+2 ;SET NEW "CC" AND PRIORITY
 9558  032604  000004                         IOT               ;TRAP HERE
 9559  032606                          RETG2:
 9560  032606  100404                         BMI   1$
 9561                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
 9562                                                    ;          CONDITIONAL BRANCH INST. AND   <   -
 9563                                                    ;          REPLACE THE MOVE INSTRUCTION   <
 9564                                                    ;          WHICH FOLLOWS W/ 721           <-
 9565  032610  012742  000765                 MOV   #765,-(R2)  ;MOVE TO MAILBOX #  *******  765  *******
 9566  032614  005242                         INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
 9567  032616  000000                         HALT              ;N NOT SET
 9568  032620                          1$:
 9569  032620  001404                         BEQ   2$
 9570                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
```

```
9571
9572
9573
9574  032622  012742  000766              MOV     #766,-(R2)      ;MOVE TO MAILBOX #  *******  766  *******
9575  032626  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9576  032630  000000                      HALT                    ;Z NOT SET
9577  032632                      2$:
9578  032632  102404                      BVS     3$
9579                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9580                                                              ;           CONDITIONAL BRANCH INST. AND   <====
9581                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
9582                                                              ;           WHICH FOLLOWS W/ 707           <-=--
9583  032634  012742  000767              MOV     #767,-(R2)      ;MOVE TO MAILBOX #  *******  767  *******
9584  032640  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9585  032642  000000                      HALT                    ;V NOT SET
9586  032644                      3$:
9587  032644  103404                      BCS     4$
9588                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-===
9589                                                              ;           CONDITIONAL BRANCH INST. AND   <-===
9590                                                              ;           REPLACE THE MOVE INSTRUCTION   <==-=
9591                                                              ;           WHICH FOLLOWS W/ 702           <-=-=
9592  032646  012742  000770              MOV     #770,-(R2)      ;MOVE TO MAILBOX #  *******  770  *******
9593  032652  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9594  032654  000000                      HALT                    ;C NOT SET
9595  032656  016706  145114      4$:     MOV     CC,SP
9596  032662  042706  000017              BIC     #17,SP
9597  032666  022706  000340              CMP     #340,SP
9598  032672  001404                      BEQ     BR46A
9599                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
9600                                                              ;           CONDITIONAL BRANCH INST. AND   <-==-
9601                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
9602                                                              ;           WHICH FOLLOWS W/ 667           <====
9603  032674                      BR46:
9604  032674  012742  000771              MOV     #771,-(R2)      ;MOVE TO MAILBOX #  *******  771  *******
9605  032700  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9606  032702  000000                      HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9607  032704  012767  000022  145106  BR46A:  MOV     #22,20          ;.+2
9608  032712  005067  145104              CLR     22              ;HALT
9609            ;*****************************************************************************************
9610            ;TEST 324       TEST THAT A TRAP OCCURS ON AN EMT  INSTRUCTION
9611            ;*****************************************************************************************
9612  032716  005212              TS324:  INC     (R2)            ;UPDATE TEST NUMBER
9613  032720  022712  000324              CMP     #324,(R2)       ;SEQUENCE ERROR?
9614  032724  001006                      BNE     TS325-10        ;BR TO ERROR HALT ON SEQ ERROR
9615  032726  012706  001000              MOV     #BUFF,SP        ;STACK POINTER SETUP
9616  032732  012767  032752  145070      MOV     #RETA3,RTRAP3   ;RETURN LOCATION
9617  032740  104000                      EMT                     ;RESERVE INSTRUCTIOM, SHOULD TRAP
9618  032742  012742  000772              MOV     #772,-(R2)      ;MOVE TO MAILBOX #  *******  772  *******
9619  032746  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9620  032750  000000                      HALT                    ;EMT DIDN'T TRAP,OR WRONG $TSTNM
9621  032752                      RETA3:
9622            ;*****************************************************************************************
9623            ;TEST 325       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9624            ;*****************************************************************************************
9625  032752  005212              TS325:  INC     (R2)            ;UPDATE TEST NUMBER
9626  032754  022712  000325              CMP     #325,(R2)       ;SEQUENCE ERROR?
```

```
9627  032760  001011                        BNE     TS326-10        ;BR TO ERROR HALT ON SEQ ERROR
9628  032762  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9629  032766  012767  032776  145034        MOV     #RETB3,RTRAP3   ;RETURN POINTER
9630  032774  104000                        EMT                     ;RESERVED INSTRUCTION
9631  032776  020627  000774        RETB3:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
9632  033002  001404                        BEQ     TS326
9633                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-_=
9634                                                                ;           CONDITIONAL BRANCH INST. AND   < =_-
9635                                                                ;           REPLACE THE MOVE INSTRUCTION   <_ =-
9636                                                                ;           WHICH FOLLOWS W/ 766           <---=
9637  033004  012742  000773                MOV     #773,-(R2)      ;MOVE TO MAILBOX #  *******  773  *******
9638  033010  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9639  033012  000000                        HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
9640                                                                ;  OR SEQUENCE ERROR
9641                                ;*****************************************************************************
9642                                ;TEST 326       TEST THAT PROPER P.C. IS SAVED
9643                                ;*****************************************************************************
9644  033014  005212                TS326:  INC     (R2)            ;UPDATE TEST NUMBER
9645  033016  022712  000326                CMP     #326,(R2)       ;SEQUENCE ERROR?
9646  033022  001012                        BNE     TS327-10        ;BR TO ERROR HALT ON SEQ ERROR
9647  033024  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9648  033030  012767  033040  144772        MOV     #RETC3,RTRAP3   ;RTURN FROM TRAP POINTER
9649  033036  104000                        EMT                     ;TRAP ON THIS INSTRICTION
9650  033040  022767  033040  145726  RETC3:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
9651  033046  001404                        BEQ     TS327
9652                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-_
9653                                                                ;           CONDITIONAL BRANCH INST. AND   <=-==
9654                                                                ;           REPLACE THE MOVE INSTRUCTION   <= _-
9655                                                                ;           WHICH FOLLOWS W/ 765           <===-
9656  033050  012742  000774                MOV     #774,-(R2)      ;MOVE TO MAILBOX #  *******  774  *******
9657  033054  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9658  033056  000000                        HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
9659                                                                ;  OR SEQUENCE ERROR
9660                                ;*****************************************************************************
9661                                ;TEST 327       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
9662                                ;*****************************************************************************
9663  033060  005212                TS327:  INC     (R2)            ;UPDATE TEST NUMBER
9664  033062  022712  000327                CMP     #327,(R2)       ;SEQUENCE ERROR?
9665  033066  001037                        BNE     TS330-10        ;BR TO ERROR HALT ON SEQ ERROR
9666  033070  012706  001000                MOV     #BUFF,SP        ;SET UP
9667  033074  012767  033112  144726        MOV     #RETD3,RTRAP3   ;SET UP
9668  033102  005067  144670                CLR     CC              ;CLEAR CC AND PRIORITY
9669  033106  000257                        CCC
9670  033110  104000                        EMT                     ;TRAP
9671  033112  026727  145660  000000  RETD3:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
9672  033120  001404                        BEQ     1$
9673                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9674                                                                ;           CONDITIONAL BRANCH INST. AND   <====
9675                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
9676                                                                ;           WHICH FOLLOWS W/ 762           <=-=-
9677  033122  012742  000775                MOV     #775,-(R2)      ;MOVE TO MAILBOX #  *******  775  *******
9678  033126  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9679  033130  000000                        HALT                    ;INCORRECT STATUS
9680  033132  012706  001000        1$:     MOV     #BUFF,SP        ;SET UP
9681  033136  012767  033156  144664        MOV     #RETE3,RTRAP3   ;SET UP
9682  033144  012767  000357  144624        MOV     #357,CC         ;SET PRIORITY
```

```
9683  033152  000277                          SCC                     ;SET CC
9684  033154  104000                          EMT                     ;TRAP
9685  033156  026727  145614  000357  RETE3:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
9686  033164  001404                          BEQ     TS330
9687                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=-
9688                                                                  ;           CONDITIONAL BRANCH INST. AND   <=- =
9689                                                                  ;           REPLACE THE MOVE INSTRUCTION   <-===
9690                                                                  ;           WHICH FOLLOWS W/ 740           < -=-
9691  033166  012742  000776                  MOV     #776,-(R2)      ;MOVE TO MAILBOX #  *******  776  *******
9692  033172  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9693  033174  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9694                                                                  ;  OR SEQUENCE ERROR
9695                                ;*****************************************************************************
9696                                ;TEST 330       TEST THAT 'NEW' STATUS IS CORRECT
9697                                ;*****************************************************************************
9698  033176  005212                  TS330:  INC     (R2)            ;UPDATE TEST NUMBER
9699  033200  022712  000330                  CMP     #330,(R2)       ;SEQUENCE ERROR?
9700  033204  001106                          BNE     TS331-10        ;BR TO ERROR HALT ON SEQ ERROR
9701  033206  012706  001000                  MOV     #BUFF,SP
9702  033212  012767  033226  144610          MOV     #RETF3,RTRAP3
9703  033220  005067  144606                  CLR     RTRAP3+2        ;CLEAR FUTURE PRIORITY AND CC
9704  033224  104000                          EMT
9705  033226                          RETF3:                          ;TEST FOR "C" CLEARED
9706  033226  100004                          BPL     1$
9707                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   < -=-
9708                                                                  ;           CONDITIONAL BRANCH INST. AND   <==-=
9709                                                                  ;           REPLACE THE MOVE INSTRUCTION   <===-
9710                                                                  ;           WHICH FOLLOWS W/ 766           <===-
9711  033230  012742  000777                  MOV     #777,-(R2)      ;MOVE TO MAILBOX #  *******  777  *******
9712  033234  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9713  033236  000000                          HALT                    ;C NOT CLEARED
9714  033240                          1$:
9715  033240  001004                          BNE     2$
9716                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9717                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9718                                                                  ;           REPLACE THE MOVE INSTRUCTION   < -==
9719                                                                  ;           WHICH FOLLOWS W/ 761           <-===
9720  033242  012742  001000                  MOV     #1000,-(R2)     ;MOVE TO MAILBOX #  *******  1000  *******
9721  033246  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9722  033250  000000                          HALT                    ;Z NOT CLEARED
9723  033252                          2$:
9724  033252  102004                          BVC     3$
9725                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9726                                                                  ;           CONDITIONAL BRANCH INST. AND   <===-
9727                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9728                                                                  ;           WHICH FOLLOWS W/ 754           <-=-
9729  033254  012742  001001                  MOV     #1001,-(R2)     ;MOVE TO MAILBOX #  *******  1001  *******
9730  033260  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9731  033262  000000                          HALT                    ;V NOT CLEARED
9732  033264                          3$:
9733  033264  103004                          BCC     4$
9734                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- -=
9735                                                                  ;           CONDITIONAL BRANCH INST. AND   <---
9736                                                                  ;           REPLACE THE MOVE INSTRUCTION   <==-
9737                                                                  ;           WHICH FOLLOWS W/ 747           <- -
9738  033266  012742  001002                  MOV     #1002,-(R2)     ;MOVE TO MAILBOX #  *******  1002  *******
```

```
 9739  033272  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9740  033274  000000                        HALT                    ;C NOT CLEARED
 9741  033276  032767  000340  144472  4$:   BIT     #340,CC         ;TEST PRIORITY
 9742  033304  001404                        BEQ     5$
 9743                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9744                                                                ;           CONDITIONAL BRANCH INST. AND   <====
 9745                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 9746                                                                ;           WHICH FOLLOWS W/ 737           <====
 9747  033306  012742  001003                        MOV     #1003,-(R2)     ;MOVE TO MAILBOX #  *******  1003  *******
 9748  033312  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9749  033314  000000                        HALT                    ;PRIORITY NOT ZERO
 9750  033316  012706  001000          5$:   MOV     #BUFF,SP
 9751  033322  012767  033340  144500        MOV     #RETG3,RTRAP3
 9752  033330  012767  000357  144474        MOV     #357,RTRAP3+2   ;SET NEW ''CC'' AND PRIORITY
 9753  033336  104000                        EMT                     ;TRAP HERE
 9754  033340                          RETG3:
 9755  033340  100404                        BMI     1$
 9756                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9757                                                                ;           CONDITIONAL BRANCH INST. AND   <====
 9758                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 9759                                                                ;           WHICH FOLLOWS W/ 721           <====
 9760  033342  012742  001004                        MOV     #1004,-(R2)     ;MOVE TO MAILBOX #  *******  1004  *******
 9761  033346  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9762  033350  000000                        HALT                    ;N NOT SET
 9763  033352                          1$:
 9764  033352  001404                        BEQ     2$
 9765                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9766                                                                ;           CONDITIONAL BRANCH INST. AND   <====
 9767                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 9768                                                                ;           WHICH FOLLOWS W/ 714           <====
 9769  033354  012742  001005                        MOV     #1005,-(R2)     ;MOVE TO MAILBOX #  *******  1005  *******
 9770  033360  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9771  033362  000000                        HALT                    ;Z NOT SET
 9772  033364                          2$:
 9773  033364  102404                        BVS     3$
 9774                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9775                                                                ;           CONDITIONAL BRANCH INST. AND   <====
 9776                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 9777                                                                ;           WHICH FOLLOWS W/ 707           <====
 9778  033366  012742  001006                        MOV     #1006,-(R2)     ;MOVE TO MAILBOX #  *******  1006  *******
 9779  033372  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9780  033374  000000                        HALT                    ;V NOT SET
 9781  033376                          3$:
 9782  033376  103404                        BCS     4$
 9783                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9784                                                                ;           CONDITIONAL BRANCH INST. AND   <====
 9785                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 9786                                                                ;           WHICH FOLLOWS W/ 702           <====
 9787  033400  012742  001007                        MOV     #1007,-(R2)     ;MOVE TO MAILBOX #  *******  1007  *******
 9788  033404  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9789  033406  000000                        HALT                    ;C NOT SET
 9790  033410  000257                  4$:   CCC
 9791  033412  022767  000340  144356        CMP     #340,CC
 9792  033420  001404                        BEQ     TS331
 9793                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9794                                                                ;           CONDITIONAL BRANCH INST. AND   <====
```

```
9795                                                    ;           REPLACE THE MOVE INSTRUCTION   <----
9796                                                    ;           WHICH FOLLOWS W/ 671           <====
9797  033422  012742  001010                MOV     #1010,-(R2)     ;MOVE TO MAILBOX #  *******  1010  *******
9798  033426  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9799  033430  000000                        HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9800                                                                  ;  OR SEQUENCE ERROR
9801                                ;*****************************************************************************
9802                                ;TEST 331       TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP
9803                                ;*****************************************************************************
9804  033432  005212                TS331:  INC     (R2)            ;UPDATE TEST NUMBER
9805  033434  022712  000331                CMP     #331,(R2)       ;SEQUENCE ERROR?
9806  033440  001011                        BNE     BR47            ;BR TO ERROR HALT ON SEQ ERROR
9807                                                                ;**** F11 **** ADD +376 TO SHORTEN TEST
9808  033442  012767  104376  000012        MOV     #EMT+376,RB             ;INITIALIZE BASE EMT INSTRUCTION
9809  033450  012767  033474  144352        MOV     #RA,30          ;RETURN FROM TRAP TO RA
9810  033456  012706  001000        RC:     MOV     #BUFF,SP        ;SET UP STACK POINTER
9811  033462  104000                RB:     EMT                     ;TRAP INST. WILL BE MODIFIED TO EMT+377
9812  033464                        BR47:
9813  033464  012742  001011                MOV     #1011,-(R2)     ;MOVE TO MAILBOX #  *******  1011  *******
9814  033470  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9815  033472  000000                        HALT                    ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TSTNM
9816  033474  005267  177762        RA:     INC     RB              ;INCREMENT TRAP INSTRUCTION
9817  033500  022767  104377  177754        CMP     #104377,RB      ;EMT+377 TO EMT?
9818  033506  103363                        BHIS    RC              ;HAVE WE TESTED ALL
9819                                                                ;YES
9820  033510  012767  000032  144312        MOV     #32,30          ;/.+
9821  033516  005067  144310                CLR     32              ;HALT
9822                                ;*****************************************************************************
9823                                ;TEST 332       TEST THAT A TRAP OCCURES ON AN 'TRACE-TRT' INSTRUCTION
9824                                ;*****************************************************************************
9825  033522  005212                TS332:  INC     (R2)            ;UPDATE TEST NUMBER
9826  033524  022712  000332                CMP     #332,(R2)       ;SEQUENCE ERROR?
9827  033530  001006                        BNE     TS333-10        ;BR TO ERROR HALT ON SEQ ERROR
9828  033532  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9829  033536  012767  033556  144250        MOV     #RETA4,RTRAP4   ;RETURN LOCATION
9830  033544  000003                        TRT                     ;RESERVED INSTRUCTION, SHOULD TRAP
9831  033546  012742  001012                MOV     #1012,-(R2)     ;MOVE TO MAILBOX #  *******  1012  *******
9832  033552  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9833  033554  000000                        HALT                    ;TRT DIDN'T TRAP,OR WRONG $TSTNM
9834  033556                        RETA4:
9835                                ;*****************************************************************************
9836                                ;TEST 333       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9837                                ;*****************************************************************************
9838  033556  005212                TS333:  INC     (R2)            ;UPDATE TEST NUMBER
9839  033560  022712  000333                CMP     #333,(R2)       ;SEQUENCE ERROR?
9840  033564  001011                        BNE     TS334-10        ;BR TO ERROR HALT ON SEQ ERROR
9841  033566  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9842  033572  012767  033602  144214        MOV     #RETB4,RTRAP4   ;RETURN POINTER
9843  033600  000003                        TRT                     ;RESERVED INSTRUCTION
9844  033602  020627  000774        RETB4:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
9845  033606  001404                        BEQ     TS334
9846                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9847                                                                ;           CONDITIONAL BRANCH INST. AND   <====
9848                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
9849                                                                ;           WHICH FOLLOWS W/ 766           <====
9850  033610  012742  001013                MOV     #1013,-(R2)     ;MOVE TO MAILBOX #  *******  1013  *******
```

```
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                ;*******************************************************************************
                                ;TEST 334       TEST THAT PROPER P.C. IS SAVED
                                ;*******************************************************************************
                                TS334:  INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #334,(R2)       ;SEQUENCE ERROR?
                                        BNE     TS335-10        ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #BUFF,SP        ;STACK POINTER SETUP
                                        MOV     #RETC4,RTRAP4   ;RETURN FROM TRAP POINTER
                                        TRT                     ;TRAP ON THIS INSTRUCTION
                                RETC4:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
                                        BEQ     TS335
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <====
                                        MOV     #1014,-(R2)     ;MOVE TO MAILBOX #  *******  1014  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                ;*******************************************************************************
                                ;TEST 335       TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
                                ;*******************************************************************************
                                TS335:  INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #335,(R2)       ;SEQUENCE ERROR?
                                        BNE     TS336-10        ;BR TO ERROR HALT ON SEQ ERROR
                                        MOV     #BUFF,SP        ;SET UP
                                        MOV     #RETD4,RTRAP4   ;SET UP
                                        CLR     CC              ;CLEAR CC AND PRIORITY
                                        CCC
                                        TRT                     ;TRAP
                                RETD4:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
                                                                ;TEST FOR ALL ZEROS
                                        BEQ     1$
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
                                        MOV     #1015,-(R2)     ;MOVE TO MAILBOX #  *******  1015  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;INCORRECT STATUS
                                1$:     MOV     #BUFF,SP        ;SET UP
                                        MOV     #RETE4,RTRAP4   ;SET UP
                                        MOV     #357,CC         ;SET PRIORITY
                                        SCC                     ;SET-SET CC
                                        TRT                     ;TRAP
                                RETE4:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
                                        BEQ     TS336
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 740           <====
                                        MOV     #1016,-(R2)     ;MOVE TO MAILBOX #  *******  1016  *******
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
9907  034000  000000                        HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9908                                                                  ;  OR SEQUENCE ERROR
9909                                  ;*************************************************************************
9910                                  ;TEST 336       TEST THAT 'NEW' STATUS IS CORRECT
9911                                  ;*************************************************************************
9912  034002  005212                  TS336:  INC     (R2)            ;UPDATE TEST NUMBER
9913  034004  022712  000336                  CMP     #336,(R2)       ;SEQUENCE ERROR?
9914  034010  001410                          BNE     BR51            ;BR TO ERROR HALT ON SEQ ERROR
9915  034012  012706  001000                  MOV     #BUFF,SP
9916  034016  012767  034032  143770          MOV     #RETF4,RTRAP4
9917  034024  005067  143766                  CLR     RTRAP4+2        ;CLEAR FUTURE PRIORITY AND CC
9918  034030  000003                          TRT
9919  034032                          RETF4:                          ;TEST FOR "C" CLEARED
9920  034032  100004                          BPL     1$
9921                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9922                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9923                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9924                                                                  ;           WHICH FOLLOWS W/ 766           <====
9925  034034  012742  001017                  MOV     #1017,-(R2)     ;MOVE TO MAILBOX #  *******  1017  *******
9926  034040  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9927  034042  000000                          HALT                    ;C NOT CLEARED
9928  034044                          1$:
9929  034044  001004                          BNE     2$
9930                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9931                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9932                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9933                                                                  ;           WHICH FOLLOWS W/ 761           <====
9934  034046  012742  001020                  MOV     #1020,-(R2)     ;MOVE TO MAILBOX #  *******  1020  *******
9935  034052  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9936  034054  000000                          HALT                    ;Z NOT CLEARED
9937  034056                          2$:
9938  034056  102004                          BVC     3$
9939                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9940                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9941                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9942                                                                  ;           WHICH FOLLOWS W/ 754           <====
9943  034060  012742  001021                  MOV     #1021,-(R2)     ;MOVE TO MAILBOX #  *******  1021  *******
9944  034064  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9945  034066  000000                          HALT                    ;V NOT CLEARED
9946  034070                          3$:
9947  034070  103004                          BCC     4$
9948                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9949                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9950                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9951                                                                  ;           WHICH FOLLOWS W/ 747           <====
9952  034072  012742  001022                  MOV     #1022,-(R2)     ;MOVE TO MAILBOX #  *******  1022  *******
9953  034076  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9954  034100  000000                          HALT                    ;C NOT CLEARED
9955  034102  032767  000340  143666  4$:     BIT     #340,CC         ;TEST PRIORITY
9956  034110  001404                          BEQ     5$
9957                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9958                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9959                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9960                                                                  ;           WHICH FOLLOWS W/ 737           <====
9961  034112  012742  001023                  MOV     #1023,-(R2)     ;MOVE TO MAILBOX #  *******  1023  *******
9962  034116  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
 9963  034120  000000                          HALT                    ;PRIORITY NOT ZERO
 9964  034122  012706  001000          5$:     MOV     #BUFF,SP
 9965  034126  012767  034144  143660          MOV     #RETG4,RTRAP4
 9966  034134  012767  000357  143654          MOV     #357,RTRAP4+2   ;SET NEW ''CC'' AND PRIORITY
 9967  034142  000003                          TRT                     ;TRAP HERE
 9968  034144                          RETG4:
 9969  034144  100404                          BMI     1$
 9970                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9971                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9972                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9973                                                                  ;           WHICH FOLLOWS W/ 721           <====
 9974  034146  012742  001024                MOV     #1024,-(R2)     ;MOVE TO MAILBOX #  *******  1024  *******
 9975  034152  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9976  034154  000000                          HALT                    ;N NOT SET
 9977  034156                          1$:
 9978  034156  001404                          BEQ     2$
 9979                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9980                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9981                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9982                                                                  ;           WHICH FOLLOWS W/ 714           <====
 9983  034160  012742  001025                MOV     #1025,-(R2)     ;MOVE TO MAILBOX #  *******  1025  *******
 9984  034164  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9985  034166  000000                          HALT                    ;Z NOT SET
 9986  034170                          2$:
 9987  034170  102404                          BVS     3$
 9988                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9989                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9990                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9991                                                                  ;           WHICH FOLLOWS W/ 707           <====
 9992  034172  012742  001026                MOV     #1026,-(R2)     ;MOVE TO MAILBOX #  *******  1026  *******
 9993  034176  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9994  034200  000000                          HALT                    ;V NOT SET
 9995  034202                          3$:
 9996  034202  103404                          BCS     4$
 9997                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9998                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9999                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10000                                                                  ;           WHICH FOLLOWS W/ 702           <====
10001  034204  012742  001027                MOV     #1027,-(R2)     ;MOVE TO MAILBOX #  *******  1027  *******
10002  034210  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10003  034212  000000                          HALT                    ;C NOT SET
10004  034214  016706  143556          4$:     MOV     CC,SP
10005  034220  042706  000017                  BIC     #17,SP
10006  034224  022706  000340                  CMP     #340,SP
10007  034230  001404                          BEQ     BR51A
10008                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10009                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10010                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10011                                                                  ;           WHICH FOLLOWS W/ 667           <====
10012  034232                          BR51:
10013  034232  012742  001030                MOV     #1030,-(R2)     ;MOVE TO MAILBOX #  *******  1030  *******
10014  034236  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10015  034240  000000                          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10016  034242  012767  000016  143544  BR51A:  MOV     #16,14
10017  034250  005067  143542                  CLR     16
10018
```

```
10019                                   ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
10020                                   ;ALL INSTRUCTIONS THAT ARE RESERVED
10021                                   ;SHOULD TRAP TO LOCATION 4, AND THE
10022                                   ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
10023                                   ;SHOULD BE PLACED ON THE STACK
10024
10025                                   ;*****************************************************************************
10026                                   ;TEST 337       TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION
10027                                   ;*****************************************************************************
10028  034254  005212                  TS337:  INC     (R2)            ;UPDATE TEST NUMBER
10029  034256  022712  000337                  CMP     #337,(R2)       ;SEQUENCE ERROR?
10030  034262  001006                          BNE     TS340-10        ;BR TO ERROR HALT ON SEQ ERROR
10031  034264  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10032  034270  012767  034310  143506          MOV     #RETA5,RTRAP5   ;RETURN LOCATION
10033  034276  000000                          JMP     %0              ;ILLEGAL INSTRUCTION, SHOULD TRAP
10034  034300  012742  001031                  MOV     #1031,-(R2)     ;MOVE TO MAILBOX #  *******  1031  *******
10035  034304  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10036  034306  000000                          HALT                    ;ILLEGAL INSTRUCTION DIDN'T TRAP,OR WRONG $TSTNM
10037  034310                          RETA5:
10038                                   ;*****************************************************************************
10039                                   ;TEST 340       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10040                                   ;*****************************************************************************
10041  034310  005212                  TS340:  INC     (R2)            ;UPDATE TEST NUMBER
10042  034312  022712  000340                  CMP     #340,(R2)       ;SEQUENCE ERROR?
10043  034316  001011                          BNE     TS341-10        ;BR TO ERROR HALT ON SEQ ERROR
10044  034320  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10045  034324  012767  034334  143452          MOV     #RETB5,RTRAP5   ;RETURN POINTER
10046  034332  000100                          JMP     %0              ;RESERVED INSTRUCTION
10047  034334  020627  000774          RETB5:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
10048  034340  001404                          BEQ     TS341
10049                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10050                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10051                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10052                                                                  ;           WHICH FOLLOWS W/ 766           <====
10053  034342  012742  001032                  MOV     #1032,-(R2)     ;MOVE TO MAILBOX #  *******  1032  *******
10054  034346  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10055  034350  000000                          HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
10056                                                                  ;  OR SEQUENCE ERROR
10057                                   ;*****************************************************************************
10058                                   ;TEST 341       TEST THAT PROPER P.C. IS SAVED
10059                                   ;*****************************************************************************
10060  034352  005212                  TS341:  INC     (R2)            ;UPDATE TEST NUMBER
10061  034354  022712  000341                  CMP     #341,(R2)       ;SEQUENCE ERROR?
10062  034360  001012                          BNE     TS342-10        ;BR TO ERROR HALT ON SEQ ERROR
10063  034362  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10064  034366  012767  034376  143410          MOV     #RETC5,RTRAP5   ;RETURN FROM TRAP POINTER
10065  034374  000100                          JMP     %0              ;TRAP ON THIS INSTRUCTION
10066  034376  022767  034376  144370  RETC5:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
10067  034404  001404                          BEQ     TS342
10068                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <- -
10069                                                                  ;           CONDITIONAL BRANCH INST. AND   < - -
10070                                                                  ;           REPLACE THE MOVE INSTRUCTION   <- =
10071                                                                  ;           WHICH FOLLOWS W/ 765           <-
10072  034406  012742  001033                  MOV     #1033,-(R2)     ;MOVE TO MAILBOX #  *******  1033  *******
10073  034412  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10074  034414  000000                          HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
```

```
10075                                                   ;  OR SEQUENCE ERROR
10076                                   ;*****************************************************************************
10077                                   ;TEST 342       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
10078                                   ;*****************************************************************************
10079  034416  005212                   TS342:  INC     (R2)            ;UPDATE TEST NUMBER
10080  034420  022712  000342                   CMP     #342,(R2)       ;SEQUENCE ERROR?
10081  034424  001037                           BNE     TS343-10        ;BR TO ERROR HALT ON SEQ ERROR
10082  034426  012706  001000                   MOV     #BUFF,SP        ;SET UP
10083  034432  012767  034450  143344           MOV     #RETD5,RTRAP5   ;SET UP
10084  034440  005067  143332                   CLR     CC              ;CLEAR CC AND PRIORITY
10085  034444  000257                           CCC
10086  034446  000100                           JMP     %0              ;TRAP
10087  034450  026727  144322  000000 RETD5:    CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
10088  034456  001404                           BEQ     1$
10089                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10090                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10091                                                                   ;           REPLACE THE MOVE INSTRUCTION   <-
10092                                                                   ;           WHICH FOLLOWS W/ 762           <   -
10093  034460  012742  001034                   MOV     #1034,-(R2)     ;MOVE TO MAILBOX #  *******  1034  *******
10094  034464  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10095  034466  000000                           HALT                    ;INCORRECT STATUS
10096  034470  012706  001000           1$:     MOV     #BUFF,SP        ;SET UP
10097  034474  012767  034514  143302           MOV     #RETE5,RTRAP5   ;SET UP
10098  034502  012767  000357  143266           MOV     #357,CC         ;SET PRIORITY
10099  034510  000277                           SCC                     ;SET CC
10100  034512  000100                           JMP     %0              ;TRAP
10101  034514  026727  144256  000357 RETE5:    CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
10102  034522  001404                           BEQ     TS343
10103                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=--
10104                                                                   ;           CONDITIONAL BRANCH INST. AND   < --
10105                                                                   ;           REPLACE THE MOVE INSTRUCTION   <-=-=
10106                                                                   ;           WHICH FOLLOWS W/ 740           <==-=
10107  034524  012742  001035                   MOV     #1035,-(R2)     ;MOVE TO MAILBOX #  *******  1035  *******
10108  034530  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10109  034532  000000                           HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
10110                                                                   ;  OR SEQUENCE ERROR
10111                                   ;*****************************************************************************
10112                                   ;TEST 343       TEST THAT 'NEW' STATUS IS CORRECT
10113                                   ;*****************************************************************************
10114  034534  005212                   TS343:  INC     (R2)            ;UPDATE TEST NUMBER
10115  034536  022712  000343                   CMP     #343,(R2)       ;SEQUENCE ERROR?
10116  034542  001106                           BNE     TS344-10        ;BR TO ERROR HALT ON SEQ ERROR
10117  034544  012706  001000                   MOV     #BUFF,SP
10118  034550  012767  034564  143226           MOV     #RETF5,RTRAP5
10119  034556  005067  143224                   CLR     RTRAP5+2        ;CLEAR FUTURE PRIORITY AND CC
10120  034562  000100                           JMP     %0
10121  034564                           RETF5:                          ;TEST FOR "C" CLEARED
10122  034564  100004                           BPL     1$
10123                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
10124                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10125                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10126                                                                   ;           WHICH FOLLOWS W/ 766           <====
10127  034566  012742  001036                   MOV     #1036,-(R2)     ;MOVE TO MAILBOX #  *******  1036  *******
10128  034572  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10129  034574  000000                           HALT                    ;C NOT CLEARED
10130  034576                           1$:
```

```
10131  034576  001004                        BNE     2$
10132                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10133                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10134                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10135                                                  ;           WHICH FOLLOWS W/ 761           <====
10136  034600  012742  001037                  MOV     #1037,-(R2)    ;MOVE TO MAILBOX #  *******  1037  *******
10137  034604  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
10138  034606  000000                        HALT                   ;Z NOT CLEARED
10139  034610                          2$:
10140  034610  102004                        BVC     3$
10141                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10142                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10143                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10144                                                  ;           WHICH FOLLOWS W/ 754           <====
10145  034612  012742  001040                  MOV     #1040,-(R2)    ;MOVE TO MAILBOX #  *******  1040  *******
10146  034616  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
10147  034620  000000                        HALT                   ;V NOT CLEARED
10148  034622                          3$:
10149  034622  103004                        BCC     4$
10150                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10151                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10152                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10153                                                  ;           WHICH FOLLOWS W/ 747           <====
10154  034624  012742  001041                  MOV     #1041,-(R2)    ;MOVE TO MAILBOX #  *******  1041  *******
10155  034630  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
10156  034632  000000                        HALT                   ;C NOT CLEARED
10157  034634  032767  000357  143134  4$:  BIT     #357,CC        ;TEST PRIORITY
10158  034642  001404                        BEQ     5$
10159                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10160                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10161                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10162                                                  ;           WHICH FOLLOWS W/ 737           <====
10163  034644  012742  001042                  MOV     #1042,-(R2)    ;MOVE TO MAILBOX #  *******  1042  *******
10164  034650  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
10165  034652  000000                        HALT                   ;PRIORITY NOT ZERO
10166  034654  012706  001000          5$:   MOV     #BUFF,SP
10167  034660  012767  034676  143116        MOV     #RETG5,RTRAP5
10168  034666  012767  000357  143112        MOV     #357,RTRAP5+2  ;SET NEW "CC" AND PRIORITY
10169  034674  000100                        JMP     %0             ;TRAP HERE
10170  034676                        RETG5:
10171  034676  100404                        BMI     1$
10172                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10173                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10174                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10175                                                  ;           WHICH FOLLOWS W/ 721           <====
10176  034700  012742  001043                  MOV     #1043,-(R2)    ;MOVE TO MAILBOX #  *******  1043  *******
10177  034704  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
10178  034706  000000                        HALT                   ;N NOT SET
10179  034710                          1$:
10180  034710  001404                        BEQ     2$
10181                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10182                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10183                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10184                                                  ;           WHICH FOLLOWS W/ 714           <====
10185  034712  012742  001044                  MOV     #1044,-(R2)    ;MOVE TO MAILBOX #  *******  1044  *******
10186  034716  005242                        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
```

```
10187  034720  000000                          HALT                    ;Z NOT SET
10188  034722                          2$:
10189  034722  102404                          BVS     3$
10190                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-=-
10191                                           ;           CONDITIONAL BRANCH INST. AND   <=-=-
10192                                           ;           REPLACE THE MOVE INSTRUCTION   <-==
10193                                           ;           WHICH FOLLOWS W/ 707           <-==
10194  034724  012742  001045                  MOV     #1045,-(R2)     ;MOVE TO MAILBOX #  *******  1045  *******
10195  034730  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10196  034732  000000                          HALT                    ;V NOT SET
10197  034734                          3$:
10198  034734  103404                          BCS     4$
10199                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-=-=
10200                                           ;           CONDITIONAL BRANCH INST. AND   <=-=-
10201                                           ;           REPLACE THE MOVE INSTRUCTION   <-===
10202                                           ;           WHICH FOLLOWS W/ 702           <=-=-
10203  034736  012742  001046                  MOV     #1046,-(R2)     ;MOVE TO MAILBOX #  *******  1046  *******
10204  034742  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10205  034744  000000                          HALT                    ;C NOT SET
10206  034746  016706  143024          4$:     MOV     CC,SP
10207  034752  022706  000357                  CMP     #357,SP
10208  034756  001404                          BEQ     TS344
10209                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10210                                           ;           CONDITIONAL BRANCH INST. AND   <====
10211                                           ;           REPLACE THE MOVE INSTRUCTION   <-==-
10212                                           ;           WHICH FOLLOWS W/ 671           <====
10213  034760  012742  001047                  MOV     #1047,-(R2)     ;MOVE TO MAILBOX #  *******  1047  *******
10214  034764  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10215  034766  000000                          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10216                                                                  ;  OR SEQUENCE ERROR
10217                                  ;*******************************************************************************
10218                                  ;TEST 344       TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUCION
10219                                  ;*******************************************************************************
10220  034770  005212                  TS344:  INC     (R2)            ;UPDATE TEST NUMBER
10221  034772  022712  000344                  CMP     #344,(R2)       ;SEQUENCE ERROR?
10222  034776  001006                          BNE     TS345-10        ;BR TO ERROR HALT ON SEQ ERROR
10223  035000  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10224  035004  012767  035024  142772          MOV     #RETH5,RTRAP5   ;RETURN LOCATION
10225  035012  004000                          JSR     %0,%0           ;RESERVED INSTRUCTION, SHOULD TRAP
10226  035014  012742  001050                  MOV     #1050,-(R2)     ;MOVE TO MAILBOX #  *******  1050  *******
10227  035020  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10228  035022  000000                          HALT                    ;DIDN'T TRAP,OR WRONG $TSTNM
10229  035024                          RETH5:
10230                                  ;*******************************************************************************
10231                                  ;TEST 345       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10232                                  ;*******************************************************************************
10233  035024  005212                  TS345:  INC     (R2)            ;UPDATE TEST NUMBER
10234  035026  022712  000345                  CMP     #345,(R2)       ;SEQUENCE ERROR?
10235  035032  001011                          BNE     TS346-10        ;BR TO ERROR HALT ON SEQ ERROR
10236  035034  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10237  035040  012767  035050  142736          MOV     #RETJ,RTRAP5    ;RETURN POINTER
10238  035046  004000                          JSR     %0,%0           ;RESERVED INSTRUCTION
10239  035050  020627  000774          RETJ:   CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
10240  035054  001404                          BEQ     TS346
10241                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10242                                           ;           CONDITIONAL BRANCH INST. AND   <=-=
```

```
10243                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10244                                                   ;           WHICH FOLLOWS W/ 766           <====
10245  035056  012742  001051          MOV     #105',-(R2)     ;MOVE TO MAILBOX #  *******  1051  *******
10246  035062  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10247  035064  000000                  HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
10248                                                   ;  OR SEQUENCE ERROR
10249                          ;*****************************************************************************
10250                          ;TEST 346       TEST THAT PROPER P.C. IS SAVED
10251                          ;*****************************************************************************
10252  035066  005212          TS346:  INC     (R2)            ;UPDATE TEST NUMBER
10253  035070  022712  000346          CMP     #346,(R2)       ;SEQUENCE ERROR?
10254  035074  001012                  BNE     TS347-10        ;BR TO ERROR HALT ON SEQ ERROR
10255  035076  012706  001000          MOV     #BUFF,SP        ;STACK POINTER SETUP
10256  035102  012767  035112  142674  MOV     #RETK,PTRAP5    ;RETURN FROM TRAP POINTER
10257  035110  004000          INSTK:  JSR     %0,%0           ;TRAP ON THIS INSTRUCTION
10258  035112  022767  035112  143654  RETK:   CMP     #INSTK+2,BUFF-4 ;CHECK FOR INCREMENED P.C.
10259  035120  001404                  BEQ     TS347
10260                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
10261                                                   ;           CONDITIONAL BRANCH INST. AND   <
10262                                                   ;           REPLACE THE MOVE INSTRUCTION   <-
10263                                                   ;           WHICH FOLLOWS W/ 765           <
10264  035122  012742  001052          MOV     #1052,-(R2)     ;MOVE TO MAILBOX #  *******  1052  *******
10265  035126  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10266  035130  000000                  HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
10267                                                   ;  OR SEQUENCE ERROR
10268
10269                          ;*****************************************************************************
10270                          ;TEST 347       TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
10271                          ;*****************************************************************************
10272  035132  005212          TS347:  INC     (R2)            ;UPDATE TEST NUMBER
10273  035134  022712  000347          CMP     #347,(R2)       ;SEQUENCE ERROR?
10274  035140  001037                  BNE     TS350-10        ;BR TO ERROR HALT ON SEQ ERROR
10275  035142  012706  001000          MOV     #BUFF,SP        ;SET UP
10276  035146  012767  035164  142630  MOV     #RETL,RTRAP5    ;SET UP
10277  035154  005067  142616          CLR     CC              ;CLEAR CC AND PRIORITY
10278  035160  000257                  CCC
10279  035162  004000                  JSR     %0,%0           ;TRAP
10280  035164  026727  143606  000000  RETL:   CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
10281  035172  001404                  BEQ     1$
10282                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10283                                                   ;           CONDITIONAL BRANCH INST. AND   <
10284                                                   ;           REPLACE THE MOVE INSTRUCTION   <
10285                                                   ;           WHICH FOLLOWS W/ 762           <
10286  035174  012742  001053          MOV     #1053,-(R2)     ;MOVE TO MAILBOX #  *******  1053  *******
10287  035200  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10288  035202  000000                  HALT                    ;INCORRCT STATUS
10289  035204  012706  001000  1$:     MOV     #BUFF,SP        ;SET UP
10290  035210  012767  035230  142566  MOV     #RETM,RTRAP5    ;SET UP
10291  035216  012767  000357  142552  MOV     #357,CC         ;SET PRIORITY
10292  035224  000277                  SCC                     ;SET CC
10293  035226  004000                  JSR     %0,%0           ;TRAP
10294  035230  026727  143542  000357  RETM:   CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
10295  035236  001404                  BEQ     TS350
10296                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <--
10297                                                   ;           CONDITIONAL BRANCH INST. AND   <===
10298                                                   ;           REPLACE THE MOVE INSTRUCTION   <===
```

```
10299
10300  035240  012742  001054                  MOV     #1054,-(R2)     ;MOVE TO MAILBOX #  *******  1054  *******
10301  035244  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10302  035246  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
10303                                                                   ;  OR SEQUENCE ERROR
10304                                  ;*****************************************************************************
10305                                  ;TEST 350       TEST THAT 'NEW'' STATUS IS CORRECT
10306                                  ;*****************************************************************************
10307  035250  005212                  TS350.  INC     (R2)            ;UPDATE TEST NUMBER
10308  035252  022712  000350                  CMP     #350,(R2)       ;SEQUENCE ERROR?
10309  035256  001105                          BNE     TS351-10        ;BR TO ERROR HALT ON SEQ ERROR
10310  035260  012706  001000                  MOV     #BUFF,SP
10311  035264  012767  035300  142512          MOV     #RETN,RTRAP5
10312  035272  005067  142510                  CLR     RTRAP5+2        ;CLEAR FUTURE PRIORITY AND CC
10313  035276  004000                          JSR     %0,%0
10314  035300                          RETN:                           ;TEST FOR ''C'' CLEARED
10315  035300  100004                          BPL     1$
10316                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10317                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10318                                                                   ;           REPLACE THE MOVE INSTRUCTION   <
10319                                                                   ;           WHICH FOLLOWS W/ 766           <
10320  035302  012742  001055                  MOV     #1055,-(R2)     ;MOVE TO MAILBOX #  *******  1055  *******
10321  035306  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10322  035310  000000                          HALT                    ;C NOT CLEARED
10323  035312                          1$:
10324  035312  001004                          BNE     2$
10325                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10326                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10327                                                                   ;           REPLACE THE MOVE INSTRUCTION   <-
10328                                                                   ;           WHICH FOLLOWS W/ 761           <
10329  035314  012742  001056                  MOV     #1056,-(R2)     ;MOVE TO MAILBOX #  *******  1056  *******
10330  035320  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10331  035322  000000                          HALT                    ;Z NOT CLEARED
10332  035324                          2$:
10333  035324  102004                          BVC     3$
10334                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10335                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10336                                                                   ;           REPLACE THE MOVE INSTRUCTION   <-
10337                                                                   ;           WHICH FOLLOWS W/ 754           <
10338  035326  012742  001057                  MOV     #1057,-(R2)     ;MOVE TO MAILBOX #  *******  1057  *******
10339  035332  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10340  035334  000000                          HALT                    ;V NOT CLEARED
10341  035336                          3$:
10342  035336  103004                          BCC     4$
10343                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <-
10344                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10345                                                                   ;           REPLACE THE MOVE INSTRUCTION   <
10346                                                                   ;           WHICH FOLLOWS W/ 747           <
10347  035340  012742  001060                  MOV     #1060,-(R2)     ;MOVE TO MAILBOX #  *******  1060  *******
10348  035344  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10349  035346  000000                          HALT                    ;C NOT CLEARED
10350  035350  016700  142422          4$:     MOV     CC,%0           ;TEMP STORAGE
10351  035354  001404                          BEQ     5$
10352                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10353                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10354                                                                   ;           REPLACE THE MOVE INSTRUCTION   <
```

```
10355
10356  035356  012742  001061                  MOV     #1061,-(R2)     ;MOVE TO MAILBOX #  *******  1061  *******
10357  035362  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10358  035364  000000                          HALT                    ;PRIORITY NOT ZERO
10359  035366  012706  001000          5$:     MOV     #BUFF,SP
10360  035372  012767  035410  142404          MOV     #RETO,RTRAP5
10361  035400  012767  000357  142400          MOV     #357,RTRAP5+2   ;SET NEW "CC" AND PRIORITY
10362  035406  004000                          JSR     %0,%0           ;TRAP HERE
10363  035410                          RETO:
10364  035410  100404                          BMI     1$
10365                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
10366                                                                  ;           CONDITIONAL BRANCH INST. AND
10367                                                                  ;           REPLACE THE MOVE INSTRUCTION
10368                                                                  ;           WHICH FOLLOWS W/ 722
10369  035412  012742  001062                  MOV     #1062,-(R2)     ;MOVE TO MAILBOX #  *******  1062  *******
10370  035416  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10371  035420  000000                          HALT                    ;N NOT SET
10372  035422                          1$:
10373  035422  001404                          BEQ     2$
10374                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
10375                                                                  ;           CONDITIONAL BRANCH INST. AND
10376                                                                  ;           REPLACE THE MOVE INSTRUCTION
10377                                                                  ;           WHICH FOLLOWS W/ 715
10378  035424  012742  001063                  MOV     #1063,-(R2)     ;MOVE TO MAILBOX #  *******  1063  *******
10379  035430  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10380  035432  000000                          HALT                    ;Z NOT SET
10381  035434                          2$:
10382  035434  102404                          BVS     3$
10383                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
10384                                                                  ;           CONDITIONAL BRANCH INST. AND
10385                                                                  ;           REPLACE THE MOVE INSTRUCTION
10386                                                                  ;           WHICH FOLLOWS W/ 710
10387  035436  012742  001064                  MOV     #1064,-(R2)     ;MOVE TO MAILBOX #  *******  1064  *******
10388  035442  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10389  035444  000000                          HALT                    ;V NOT SET
10390  035446                          3$:
10391  035446  103404                          BCS     4$
10392                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
10393                                                                  ;           CONDITIONAL BRANCH INST. AND
10394                                                                  ;           REPLACE THE MOVE INSTRUCTION
10395                                                                  ;           WHICH FOLLOWS W/ 703
10396  035450  012742  001065                  MOV     #1065,-(R2)     ;MOVE TO MAILBOX #  *******  1065  *******
10397  035454  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10398  035456  000000                          HALT                    ;C NOT SET
10399  035460  016700  142312          4$:     MOV     CC,%0
10400  035464  022700  000357                  CMP     #357,%0
10401  035470  001404                          BEQ     TS351
10402                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
10403                                                                  ;           CONDITIONAL BRANCH INST. AND
10404                                                                  ;           REPLACE THE MOVE INSTRUCTION
10405                                                                  ;           WHICH FOLLOWS W/ 672
10406  035472  012742  001066                  MOV     #1066,-(R2)     ;MOVE TO MAILBOX #  *******  1066  *******
10407  035476  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10408  035500  000000                          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10409                                                                  ;  OR SEQUENCE ERROR
10410
```

```
10411                                   ;*****************************************************************************
10412                                   ;TEST 351       TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS
10413                                   ;*****************************************************************************
10414  035502  005212                   TS351:  INC     (R2)            ;UPDATE TEST NUMBER
10415  035504  022712  000351                   CMP     #351,(R2)       ;SEQUENCE ERROR?
10416  035510  001006                           BNE     TS352-10        ;BR TO ERROR HALT ON SEQ ERROR
10417  035512  012706  000150                   MOV     #150,%6         ;R6 = 150
10418  035516  012767  035536  142260         MOV     #TDEC1,4        ;STACK OVERFLOW TRAP POINTER
10419  035524  005746                           TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
10420  035526  012742  001067                   MOV     #1067,-(R2)     ;MOVE TO MAILBOX #  *******  1067  *******
10421  035532  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10422  035534  000000                           HALT                    ;SHOULD HAVE TRAPPED,OR WRONG $TSTNM
10423  035536                          TDEC1:
10424
10425                                   ;*****************************************************************************
10426                                   ;TEST 352       TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
10427                                   ;*****************************************************************************
10428  035536  005212                   TS352:  INC     (R2)            ;UPDATE TEST NUMBER
10429  035540  022712  000352                   CMP     #352,(R2)       ;SEQUENCE ERROR?
10430  035544  001011                           BNE     TS353-10        ;BR TO ERROR HALT ON SEQ ERROR
10431  035546  012706  000150                   MOV     #150,%6         ;R6 = 150
10432  035552  012767  035562  142224         MOV     #TDEC2,4        ;TRAP POINTER
10433  035560  005746                           TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
10434  035562  020627  000142          TDEC2:  CMP     %6,#142         ;DID R6 DECREMENT
10435  035566  001404                           BEQ     TS353
10436                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10437                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10438                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10439                                                                   ;           WHICH FOLLOWS W/ 766           <====
10440  035570  012742  001070                           MOV     #1070,-(R2)     ;MOVE TO MAILBOX #  *******  1070  *******
10441  035574  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10442  035576  000000                           HALT                    ;R6 NOT = 142,OR WRONG $TSTNM
10443                                                                   ;  OR SEQUENCE ERROR
10444
10445                                   ;*****************************************************************************
10446                                   ;TEST 353       TEST DIFFERENT TYPES OF OVERFLOW
10447                                   ;*****************************************************************************
10448  035600  005212                   TS353:  INC     (R2)            ;UPDATE TEST NUMBER
10449  035602  022712  000353                   CMP     #353,(R2)       ;SEQUENCE ERROR?
10450  035606  001041                           BNE     TS354-10        ;BR TO ERROR HALT ON SEQ ERROR
10451  035610  012706  000150                   MOV     #150,%6
10452  035614  005067  142326                   CLR     146             ;STATUS WORD OF LOC 10
10453  035620  012767  035630  142156         MOV     #TDEC3,4        ;RETURN TO LOC 4
10454  035626  005246                           INC     -(6)
10455  035630  005767  142312          TDEC3:  TST     146
10456  035634  001004                           BNE     1$
10457                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <
10458                                                                   ;           CONDITIONAL BRANCH INST. AND   <
10459                                                                   ;           REPLACE THE MOVE INSTRUCTION   <
10460                                                                   ;           WHICH FOLLOWS W/ 764           <
10461  035636  012742  001071                           MOV     #1071,-(R2)     ;MOVE TO MAILBOX #  *******  1071  *******
10462  035642  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10463  035644  000000                           HALT                    ;INCREMENT OPERATION NOT INHIBITED
10464  035646  012705  001000          1$:     MOV     #1000,%5
10465  035652  012706  000400                   MOV     #400,%6
10466  035656  012767  035676  142120         MOV     #TDEC4,4
```

```
10467  035664  124645                          CMPB    -(6),-(5)
10468  035666  012742  001072                  MOV     #1072,-(R2)     ;MOVE TO MAILBOX #  *******  1072  *******
10469  035672  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10470  035674  000000                          HALT                    ;STACK = 400 AND DECREMENTED, SHOULD TRAP
10471  035676  012706  000400          TDEC4:  MOV     #400,%6
10472  035702  012767  035722  142074          MOV     #TDEC7,4
10473  035710  134546                          BITB    -(5),-(6)
10474  035712                          TDEC6:
10475  035712  012742  001073                  MOV     #1073,-(R2)     ;MOVE TO MAILBOX #  *******  1073  *******
10476  035716  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10477  035720  000000                          HALT                    ;NO STACK OVERFLOW,OR WRONG $TSTNM
10478  035722                          TDEC7:
10479
10480                                   ;*****************************************************************************************
10481                                   ;TEST 354       TEST THAT AN 77 CAUSES AN OVERFLOW TRAP
10482                                   ;*****************************************************************************************
10483  035722  005212                  TS354:  INC     (R2)            ;UPDATE TEST NUMBER
10484  035724  022712  000354                  CMP     #354,(R2)       ;SEQUENCE ERROR?
10485  035730  001011                          BNE     VDEC2           ;BR TO ERROR HALT ON SEQ ERROR
10486  035732  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10487  035736  012767  035754  142044          MOV     #VDEC2,10       ;SET UP 77 VECTOR
10488  035744  012767  035764  142032          MOV     #VDEC1,4        ;SET UP OVERFLOW VECTOR
10489  035752  000077                          77                      ;THIS TRAP SHOULD CAUSE OVERFLOW
10490  035754                          VDEC2:
10491  035754  012742  001074                  MOV     #1074,-(R2)     ;MOVE TO MAILBOX #  *******  1074  *******
10492  035760  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10493  035762  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10494  035764  012767  000012  142016  VDEC1:  MOV     #10+2,10
10495                                   ;*****************************************************************************************
10496                                   ;TEST 355       TEST THAT AN IOT CAUSES AN OVERFLOW TRAP
10497                                   ;*****************************************************************************************
10498  035772  005212                  TS355:  INC     (R2)            ;UPDATE TEST NUMBER
10499  035774  022712  000355                  CMP     #355,(R2)       ;SEQUENCE ERROR?
10500  036000  001011                          BNE     VDEC4           ;BR TO ERROR HALT ON SEQ ERROR
10501  036002  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10502  036006  012767  036024  142004          MOV     #VDEC4,20       ;SET UP IOT VECTOR
10503  036014  012767  036034  141762          MOV     #VDEC3,4        ;SET UP OVERFLOW VECTOR
10504  036022  000004                          IOT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10505  036024                          VDEC4:
10506  036024  012742  001075                  MOV     #1075,-(R2)     ;MOVE TO MAILBOX #  *******  1075  *******
10507  036030  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10508  036032  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10509  036034  012767  000022  141756  VDEC3:  MOV     #20+2,20
10510                                   ;*****************************************************************************************
10511                                   ;TEST 356       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
10512                                   ;*****************************************************************************************
10513  036042  005212                  TS356:  INC     (R2)            ;UPDATE TEST NUMBER
10514  036044  022712  000356                  CMP     #356,(R2)       ;SEQUENCE ERROR?
10515  036050  001011                          BNE     VDEC6           ;BR TO ERROR HALT ON SEQ ERROR
10516  036052  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10517  036056  012767  036074  141744          MOV     #VDEC6,30       ;SET UP EMT VECTOR
10518  036064  012767  036104  141712          MOV     #VDEC5,4        ;SET UP OVERFLOW VECTOR
10519  036072  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10520  036074                          VDEC6:
10521  036074  012742  001076                  MOV     #1076,-(R2)     ;MOVE TO MAILBOX #  *******  1076  *******
10522  036100  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
10523  036102  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10524  036104  012767  000032  141716  VDEC5:  MOV     #30+2,30
10525                                  ;*****************************************************************************
10526                                  ;TEST 357       TEST THAT AN TRAP CAUSES AN OVERFLOW TRAP
10527                                  ;*****************************************************************************
10528  036112  005212                  TS357:  INC     (R2)            ;UPDATE TEST NUMBER
10529  036114  022712  000357                  CMP     #357,(R2)       ;SEQUENCE ERROR?
10530  036120  001011                          BNE     VDEC8           ;BR TO ERROR HALT ON SEQ ERROR
10531  036122  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10532  036126  012767  036144  141700          MOV     #VDEC8,34       ;SET UP TRAP VECTOR
10533  036134  012767  036154  141642          MOV     #VDEC7,4        ;SET UP OVERFLOW VECTOR
10534  036142  104400                          TRAP                    ;THIS TRAP SHOULD CAUSE OVERFLOW
10535  036144                          VDEC8:
10536  036144  012742  001077                  MOV     #1077,-(R2)     ;MOVE TO MAILBOX #  *******  1077  *******
10537  036150  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10538  036152  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10539  036154  012767  000036  141652  VDEC7:  MOV     #34+2,34
10540                                  ;*****************************************************************************
10541                                  ;TEST 360       TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
10542                                  ;*****************************************************************************
10543  036162  005212                  TS360:  INC     (R2)            ;UPDATE TEST NUMBER
10544  036164  022712  000360                  CMP     #360,(R2)       ;SEQUENCE ERROR?
10545  036170  001011                          BNE     VDEC10          ;BR TO ERROR HALT ON SEQ ERROR
10546  036172  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10547  036176  012767  036214  141610          MOV     #VDEC10,14      ;SET UP TRT VECTOR
10548  036204  012767  036224  141572          MOV     #VDEC9,4        ;SET UP OVERFLOW VECTOR
10549  036212  000003                          TRT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10550  036214                          VDEC10:
10551  036214  012742  001100                  MOV     #1100,-(R2)     ;MOVE TO MAILBOX #  *******  1100  *******
10552  036220  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10553  036222  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10554  036224  012767  000016  141562  VDEC9:  MOV     #14+2,14
10555                                  ;*****************************************************************************
10556                                  ;TEST 361       TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
10557                                  ;*****************************************************************************
10558  036232  005212                  TS361:  INC     (R2)            ;UPDATE TEST NUMBER
10559  036234  022712  000361                  CMP     #361,(R2)       ;SEQUENCE ERROR?
10560  036240  001011                          BNE     VDEC11          ;BR TO ERROR HALT ON SEQ ERROR
10561  036242  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10562  036246  012767  036264  141530          MOV     #VDEC11,4       ;SET UP ILLA VECTOR
10563  036254  012767  036274  141522          MOV     #VDEC12,4       ;SET UP OVERFLOW VECTOR
10564  036262  004700                          ILLA                    ;THIS TRAP SHOULD CAUSE OVERFLOW
10565  036264                          VDEC11:
10566  036264  012742  001101                  MOV     #1101,-(R2)     ;MOVE TO MAILBOX #  *******  1101  *******
10567  036270  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10568  036272  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10569  036274  012767  000006  141502  VDEC12: MOV     #4+2,4
10570  036302  020627  000370                  CMP     %6,#370         ;STACK PUSHED FOUR WORDS?
10571  036306  001404                          BEQ     TS362
10572                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10573                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10574                                                                  ;           REPLACE THE MOVE INSTRUCTION   <===-
10575                                                                  ;           WHICH FOLLOWS W/ 754           <= =-
10576  036310  012742  001102                  MOV     #1102,-(R2)     ;MOVE TO MAILBOX #  *******  1102  *******
10577  036314  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10578  036316  000000                          HALT                    ;TRAP OVERFLOW DID NOT OCCUR
```

```
                                                        ;  OR SEQUENCE ERROR
                                        ;*****************************************************************************
                                        ;TEST 362       TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
                                        ;*****************************************************************************
036320  005212                TS362:  INC     (R2)            ;UPDATE TEST NUMBER
036322  022712  000362                CMP     #362,(R2)       ;SEQUENCE ERROR?
036326  001011                        BNE     VDEC13          ;BR TO ERROR HALT ON SEQ ERROR
036330  012706  000400                MOV     #400,%6         ;SET UP STACK TO OVERFLOW
036334  012767  036352  141442        MOV     #VDEC13,4       ;SET UP ILLB VECTOR
036342  012767  036362  141434        MOV     #VDEC14,4       ;SET UP OVERFLOW VECTOR
036350  000100                        ILLB                    ;THIS TRAP SHOULD CAUSE OVERFLOW
036352                      VDEC13:
036352  012742  001103                MOV     #1103,-(R2)     ;MOVE TO MAILBOX #  *******  1103  *******
036356  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
036360  000000                        HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
036362  012767  000006  141414  VDEC14: MOV     #4+2,4

                                        ;*****************************************************************************
                                        ;TEST 363       TEST FOR FALSE OVERFLOW TRAP
                                        ;*****************************************************************************
036370  005212                TS363:  INC     (R2)            ;UPDATE TEST NUMBER
036372  022712  000363                CMP     #363,(R2)       ;SEQUENCE ERROR?
036376  001023                        BNE     FOVER           ;BR TO ERROR HALT ON SEQ ERROR

036400  012767  036446  141376        MOV     #FOVER,4        ;SET UP OVERFLOW POINTER
036406  012706  001002                MOV     #1002,%6
036412  005746                        TST     -(6)            ;SHOULD NOT OVERFLOW
036414  012706  002002                MOV     #2002,%6
036420  005746                        TST     -(6)            ;SHOULD NOT OVERFLOW
036422  012706  004002                MOV     #4002,%6
036426  005746                        TST     -(6)            ;SHOULD NOT OVERFLOW
036430  012706  010002                MOV     #10002,%6
036434  005746                        TST     -(6)
036436  012706  020000                MOV     #20000,%6       ;SHOULD NOT OVERFLOW
036442  005746                        TST     -(6)
036444  000404                        BR      STP

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <===-
                                                        ;           CONDITIONAL BRANCH INST. AND    <===-
                                                        ;           REPLACE THE MOVE INSTRUCTION    <===-
                                                        ;           WHICH FOLLOWS W/ 754            <-===
036446                      FOVER:
036446  012742  001104                MOV     #1104,-(R2)     ;MOVE TO MAILBOX #  *******  1104  *******
036452  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
036454  000000                        HALT                    ;IT OVERFLOWED,OR WRONG $TSTNM
```

```
10623  036456  012767  000006  141320  STP:    MOV     #6,4
10624  036464  005067  141316          CLR     6
10625                                  ;*****************************************************************************************
10626                                  ;TEST 364       TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
10627                                  ;*****************************************************************************************
10628  036470  005212                  TS364:  INC     (R2)            ;UPDATE TEST NUMBER
10629  036472  022712  000364                  CMP     #364,(R2)       ;SEQUENCE ERROR?
10630  036476  001013                          BNE     TS365-10        ;BR TO ERROR HALT ON SEQ ERROR
10631  036500  012706  001000                  MOV     #BUFF,SP
10632  036504  012767  036536  141302          MOV     #RETAT,RTRAP4   ;SET UP TO TRAP TO 14
10633  036512  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10634  036516  012746  036524                  MOV     #.+6,-(SP)      ;PUSH PC
10635  036522  000002                          RTI                     ;SET T BIT
10636  036524  000240                          NOP                     ;TRAP HERE
10637  036526  012742  001105                  MOV     #1105,-(R2)     ;MOVE TO MAILBOX #  *******  1105  *******
10638  036532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10639  036534  000000                          HALT                    ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10640  036536                          RETAT:
10641                                  ;*****************************************************************************************
10642                                  ;TEST 365       TEST  STACK POINTER DECREMENTS
10643                                  ;*****************************************************************************************
10644  036536  005212                  TS365:  INC     (R2)            ;UPDATE TEST NUMBER
10645  036540  022712  000365                  CMP     #365,(R2)       ;SEQUENCE ERROR?
10646  036544  001022                          BNE     TS366-10        ;BR TO ERROR HALT ON SEQ ERROR
10647  036546  012706  001000                  MOV     #BUFF,SP
10648  036552  012767  036604  141234          MOV     #RETBT,RTRAP4
10649  036560  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10650  036564  012746  036572                  MOV     #.+6,-(SP)      ;PUSH PC
10651  036570  000002                          RTI                     ;SET T BIT
10652  036572  000240                          NOP                     ;TRAP HERE
10653  036574  012742  001106                  MOV     #1106,-(R2)     ;MOVE TO MAILBOX #  *******  1106  *******
10654  036600  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10655  036602  000000                          HALT                    ;TRACE BIT DID NOT TRAP!
10656  036604  020627  000774          RETBT:  CMP     SP,#BUFF-4
10657  036610  001404                          BEQ     TS366
10658                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10659                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10660                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10661                                                                  ;           WHICH FOLLOWS W/ 755           <====
10662  036612  012742  001107                  MOV     #1107,-(R2)     ;MOVE TO MAILBOX #  *******  1107  *******
10663  036616  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10664  036620  000000                          HALT                    ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10665                                                                  ;  OR SEQUENCE ERROR
10666                                  ;*****************************************************************************************
10667                                  ;TEST 366       TEST  FOR PROPER PC ON STACK
10668                                  ;*****************************************************************************************
10669  036622  005212                  TS366:  INC     (R2)            ;UPDATE TEST NUMBER
10670  036624  022712  000366                  CMP     #366,(R2)       ;SEQUENCE ERROR?
10671  036630  001016                          BNE     TS367-10        ;BR TO ERROR HALT ON SEQ ERROR
10672  036632  012706  001000                  MOV     #BUFF,SP
10673  036636  012767  036656  141150          MOV     #RETCT,RTRAP4
10674  036644  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10675  036650  012746  036656                  MOV     #.+6,-(SP)      ;PUSH PC
10676  036654  000002                          RTI                     ;SET T BIT
10677                                                                  ;TRAP HERE
10678  036656  022767  036656  142110  RETCT:  CMP     #.,BUFF-4
```

```
10679  036664  001404                          BEQ     TS367
10680                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10681                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10682                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10683                                                                   ;           WHICH FOLLOWS W/ 761           <====
10684  036666  012742  001110                    MOV     #1110,-(R2)     ;MOVE TO MAILBOX #  *******  1110  *******
10685  036672  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10686  036674  000000                          HALT                    ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
10687                                                                   ;  OR SEQUENCE ERROR
10688
10689
10690                                  ;*************************************************************************************
10691                                  ;TEST 367       TEST  THAT RTT POPS T- BIT
10692                                  ;*************************************************************************************
10693  036676  005212                  TS367:  INC     (R2)            ;UPDATE TEST NUMBER
10694  036700  022712  000367                  CMP     #367,(R2)       ;SEQUENCE ERROR?
10695  036704  001015                          BNE     TS370-10        ;BR TO ERROR HALT ON SEQ ERROR
10696
10697  036706  012706  001000                  MOV     #BUFF,SP
10698  036712  005001                          CLR     R1              ;CLEAR R1
10699  036714  012746  000020                  MOV     #20,-(SP)
10700  036720  012746  036734                  MOV     #RTT1,-(SP)
10701  036724  012767  036750  141062          MOV     #RTT2,14
10702  036732  000006                          RTT
10703  036734  000240                  RTT1:   NOP
10704  036736  001404                          BEQ     TS370
10705                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10706                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10707                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10708                                                                   ;           WHICH FOLLOWS W/ 762           <====
10709  036740  012742  001111                    MOV     #1111,-(R2)     ;MOVE TO MAILBOX #  *******  1111  *******
10710  036744  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10711  036746  000000                          HALT                    ;T-BIT DID NOT TRAP,OR WRONG $TESTN
10712                                                                   ;  OR SEQUENCE ERROR
10713
10714  036750                          RTT2:
10715                                  ;*************************************************************************************
10716                                  ;TEST 370       TEST  THAT RTT ALLOWS ONE INST. BEFORE TRAP
10717                                  ;*************************************************************************************
10718  036750  005212                  TS370:  INC     (R2)            ;UPDATE TEST NUMBER
10719  036752  022712  000370                  CMP     #370,(R2)       ;SEQUENCE ERROR?
10720  036756  001030                          BNE     TS371-10        ;BR TO ERROR HALT ON SEQ ERROR
10721  036760  012705  177777                  MOV     #177777,%5
10722  036764  012706  001000          RTT5:   MOV     #BUFF,SP
10723  036770  012746  000020                  MOV     #20,-(SP)
10724  036774  012746  037012                  MOV     #RTT3,-(SP)
10725  037000  012767  037030  141006          MOV     #RTT4,14
10726  037006  005001                          CLR     R1              ;CLEAR R0
10727  037010  000006                          RTT                     ;SET T-BIT
10728  037012  005201                  RTT3:   INC     R1
10729  037014  005205                          INC     %5
10730  037016  001762                          BEQ     RTT5            ;DO THIS TEST NO MORE THAN 2 TIMES
10731  037020  012742  001112                  MOV     #1112,-(R2)     ;MOVE TO MAILBOX #  *******  1112  *******
10732  037024  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10733  037026  000000                          HALT                    ;DID NOT TRAP
10734  037030  005301                  RTT4:   DEC     P1              ;SEE IF RTT ALLOWS 1 INST.
```

```
10735  037032  001406                          BEQ     RTT6
10736  037034  005205                          INC     %5              ;DO THIS TEST NO MORE THAN TWO TIMES
10737  037036  001752                          BEQ     RTT5
10738                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10739                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10740                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10741                                                                  ;           WHICH FOLLOWS W/ 747           <====
10742  037040  012742  001113                    MOV     #1113,-(R2)     ;MOVE TO MAILBOX #  *******  1113  *******
10743  037044  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10744  037046  000000                          HALT                    ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
10745  037050                          RTT6:
10746                                  ;*******************************************************************************
10747                                  ;TEST 371       TEST  THAT RTI DOES NOT ALLOW 1 INST.
10748                                  ;*******************************************************************************
10749  037050  005212                  TS371:  INC     (R2)            ;UPDATE TEST NUMBER
10750  037052  022712  000371                  CMP     #371,(R2)       ;SEQUENCE ERROR?
10751  037056  001022                          BNE     TS372-10        ;BR TO ERROR HALT ON SEQ ERROR
10752  037060  012706  001000                  MOV     #BUFF,SP
10753  037064  012746  000020                  MOV     #20,-(SP)
10754  037070  012746  037106                  MOV     #RTI1,-(SP)
10755  037074  012767  037120  140712          MOV     #RTI2,14
10756  037102  005001                          CLR     R1
10757  037104  000002                          RTI                     ;SET T-BIT
10758  037106  005201                  RTI1:   INC     R1              ;RTI SHOULD NOT ALLOW THIS
10759  037110  012742  001114                  MOV     #1114,-(R2)     ;MOVE TO MAILBOX #  *******  1114  *******
10760  037114  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10761  037116  000000                          HALT                    ;T- BIT DID NOT CAUSE TRAP
10762  037120  005701                  RTI2:   TST     R1
10763                                                                  ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
10764  037122  001404                          BEQ     TS372
10765                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10766                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10767                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10768                                                                  ;           WHICH FOLLOWS W/ 755           <====
10769  037124  012742  001115                    MOV     #1115,-(R2)     ;MOVE TO MAILBOX #  *******  1115  *******
10770  037130  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10771  037132  000000                          HALT                    ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
10772                                                                  ;  OR SEQUENCE ERROR
10773
10774                                  ;*******************************************************************************
10775                                  ;TEST 372       TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
10776                                  ;*******************************************************************************
10777  037134  005212                  TS372:  INC     (R2)            ;UPDATE TEST NUMBER
10778  037136  022712  000372                  CMP     #372,(R2)       ;SEQUENCE ERROR?
10779  037142  001026                          BNE     BR70            ;BR TO ERROR HALT ON SEQ ERROR
10780
10781  037144  012706  001000                  MOV     #BUFF,%6
10782  037150  012767  037210  140636          MOV     #TRACE,14       ;TRACE TRAP
10783  037156  005027  000016                  CLR     #16
10784  037162  005027  000022                  CLR     #22
10785  037166  012767  037230  140624          MOV     #TONT1,20       ;IOT TRAP
10786  037174  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10787  037200  012746  037206                  MOV     #.+6,-(SP)      ;PUSH PC
10788  037204  000006                          RTT
10789  037206  000004                          IOT                     ;TRAP, NEW CC HAVE TRACE RESET
10790  037210                          TRACE:
```

```
10791  037210  012742  001116                  MOV     #1116,-(R2)     ;MOVE TO MAILBOX #  *******  1116  *******
10792  037214  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10793  037216  000000                          HALT                    ;TRACE TRAP WAS NOT INHIBITED
10794  037220                          BR70:
10795  037220  012742  001117                  MOV     #1117,-(R2)     ;MOVE TO MAILBOX #  *******  1117  *******
10796  037224  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10797  037226  000000                          HALT                    ;WRONG TSTNM,OR WRONG $TSTNM
10798  037230  012767  000016  140556  TONT1:  MOV     #16,14
10799  037236  012767  000022  140554          MOV     #22,20
10800                                  ;*****************************************************************************************
10801                                  ;TEST 373       TEST THAT THE TRACE BIT IS SAVED IN THE STACK
10802                                  ;*****************************************************************************************
10803  037244  005212                  TS373:  INC     (R2)            ;UPDATE TEST NUMBER
10804  037246  022712  000373                  CMP     #373,(R2)       ;SEQUENCE ERROR?
10805  037252  001020                          BNE     STP3            ;BR TO ERROR HALT ON SEQ ERROR
10806  037254  012706  001000                  MOV     #BUFF,%6        ;SET UP STACK POINTER
10807  037260  012767  037304  140526          MOV     #TRC1,14        ;TRACE TRAP RETURN
10808  037266  005067  140524                  CLR     16
10809  037272  012746  000020                  MOV     #20,-(SP)       ;SET THE T BIT
10810  037276  012746  037304                  MOV     #TRC1,-(SP)
10811  037302  000002                          RTI
10812  037304  036727  141466  000020  TRC1:   BIT     BUFF-2,#20      ;CHECK FOR T BIT ON STACK
10813  037312  001004                          BNE     STP3D
10814                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10815                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10816                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10817                                                                  ;           WHICH FOLLOWS W/ 757           <====
10818  037314                          STP3:
10819  037314  012742  001120                  MOV     #1120,-(R2)     ;MOVE TO MAILBOX #  *******  1120  *******
10820  037320  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10821  037322  000000                          HALT                    ;T BIT NOT SAVED ON THE STACK,OR WRONG $TSTNM
10822  037324  012767  000016  140462  STP3D:  MOV     #16,14
10823
10824                                  ;*****************************************************************************
10825                                  ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL
10826                                  ;ADDRESS TRAPS TO LOCATION 4.  THIS WILL RUN ON 30K SYSTEM.  BUT IF
10827                                  ;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED
10828                                  ;AT, SINCE IT MAY HAVE I/O DEVICES.  IF SWR BIT 1=1, THEN THAT AREA IS
10829                                  ;CHECKED.  (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP).  LOC 160000
10830                                  ;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION
10831                                  ;177700 (THE UNIBUS ADDRESS FOR R0 ON OLDER SYSTEMS) IS USED FOR FORCING
10832                                  ;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.
10833                                  ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
10834                                  ;*****************************************************************************************
10835                                  ;TEST 374       TEST NON-EXISTENT ADDRESS TRAPS
10836                                  ;*****************************************************************************************
10837  037332  005212                  TS374:  INC     (R2)            ;UPDATE TEST NUMBER
10838  037334  022712  000374                  CMP     #374,(R2)       ;SEQUENCE ERROR?
10839  037340  001150                          BNE     TS375-10        ;BR TO ERROR HALT ON SEQ ERROR
10840  037342  042737  010000  037412          BIC     #10000,@#HICORE ;SET HIGHT CORE LIMIT TO 160000
10841  037350  032737  000002  000322          BIT     #2,@#$SWREG     ;CHECK IF BIT 1 IS SET
10842  037356  001403                          BEQ     1$              ;BRANCH IF IT IS, LEAVE LIMIT=160000
10843  037360  052737  010000  037412          BIS     #10000,@#HICORE ;SET UPPER CORE LIMIT TO 30K (170000)
10844  037366  005000                  1$:     CLR     R0              ;
10845  037370  005067  140412                  CLR     6               ;
10846  037374  012767  037502  140402          MOV     #ATRAP,4        ;SET UP ADDRESS TRAP ENTRANCE
```

```
10847  037402  012706  001000            MOV     #BUFF,SP        ;SET STACK POINTER
10848  037406  105720            NOR:    TSTB    (0)+            ;IF OUTSIDE OF CORE, TRAP TO 4
10849  037410  020027                    CMP     R0,(PC)+        ;IS POINTER INSIDE 28K (30K) CORE
10850  037412  160000            HICORE: .WORD   160000          ;MAY BE CHANGED TO 170000 IF 30K
10851  037414  103774                    BLO     NOR             ;TEST THE REST OF CORE
10852  037416  012737  037440  000004    MOV     #ROTRAP,@#4     ;SET UP NEW VECTOR POINTER
10853  037424  105737  177700            TSTB    @#177700        ;SHOULD CAUSE A TRAP
10854  037430                    TRPADR:
10855  037430  012742  001121            MOV     #1121,-(R2)     ;MOVE TO MAILBOX #  *******  1121  *******
10856  037434  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10857  037436  000000                    HALT                    ;SHOULD HAVE TRAPED
10858                            ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
10859  037440  106767  140332  ROTRAP:   MFPS    STATUS
10860  037444  005767  140326            TST     STATUS          ;TEST PSW
10861  037450  001404                    BEQ     1$
10862                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <=-==
10863                                                            ;           CONDITIONAL BRANCH INST. AND   <====
10864                                                            ;           REPLACE THE MOVE INSTRUCTION   <==-=
10865                                                            ;           WHICH FOLLOWS W/ 733           <====
10866  037452  012742  001122            MOV     #1122,-(R2)     ;MOVE TO MAILBOX #  *******  1122  *******
10867  037456  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10868  037460  000000                    HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
10869  037462  026727  141306  037430  1$:     CMP     BUFF-4,#TRPADR  ;TEST OLD PC AT STACK
10870  037470  001453                    BEQ     TRAPB
10871                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10872                                                            ;           CONDITIONAL BRANCH INST. AND   <=-==
10873                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
10874                                                            ;           WHICH FOLLOWS W/ 723           <====
10875  037472  012742  001123            MOV     #1123,-(R2)     ;MOVE TO MAILBOX #  *******  1123  *******
10876  037476  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10877  037500  000000                    HALT                    ;OLD PC WAS NOT SAVED
10878                            ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
10879  037502  005300            ATRAP:  DEC     R0              ;
10880  037504  010067  000032            MOV     R0,CORH         ;MOVE THE FIRST NXM LOCATION IN CORH
10881                            ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
10882  037510  013700  037412            MOV     @#HICORE,R0     ;SET UP THE HIGHEST MEM LOCATION
10883  037514  005300                    DEC     R0              ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
10884  037516  000402                    BR      NOSUB           ;DON'T SUBTRACT 1K FIRST TIME
10885  037520  162700  001000  CTRAP:    SUB     #1000,R0        ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
10886                                                            ;TO SPEED UP TESTING
10887  037524  012767  037556  140252  NOSUB:  MOV     #BTRAP,4        ;SET UP THE VECTOR
10888  037532  012706  001000            MOV     #BUFF,SP
10889  037536  005710                    TST     (R0)            ;DOES THIS MEMORY EXIST?
10890                                                            ;IF NXM, TRAP TO BTRAP
10891  037540  020027            DTRAP1: CMP     R0,(PC)+        ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
10892                                                            ;TRAP TO ATRAP
10893  037542  000000            CORH:   .WORD   0
10894  037544  101425                    BLOS    TRAPB
10895                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <==-=
10896                                                            ;           CONDITIONAL BRANCH INST. AND   <=-==
10897                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
10898                                                            ;           WHICH FOLLOWS W/ 675           <====
10899  037546  012742  001124            MOV     #1124,-(R2)     ;MOVE TO MAILBOX #  *******  1124  *******
10900  037552  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10901  037554  000000                    HALT                    ;CONTENTS OF R0 SHOULD BE LESS THAN OR EQUAL  TO CORH
10902                                                            ;IF THIS COMPARISON FAILS IT MEANS
```

```
10903
10904                                                   ;THAT SOME LEGAL ADDRESS TRAPPED, OR
10905                                                   ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
10905  037556  106767  140214          BTRAP:  MFPS    STATUS
10906  037562  005767  140210                  TST     STATUS
10907  037566  001404                          BEQ     1$
10908                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
10909                                                   ;           CONDITIONAL BRANCH INST. AND    <====
10910                                                   ;           REPLACE THE MOVE INSTRUCTION    <====
10911                                                   ;           WHICH FOLLOWS W/ 664            <====
10912  037570  012742  001125                          MOV     #1125,-(R2)     ;MOVE TO MAILBOX #  *******  1125  *******
10913  037574  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10914  037576  000000                                  HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
10915  037600  026727  141170  037540    1$:     CMP     BUFF-4,#DTRAP1  ;CHECK IF TRAP PC IS OK
10916  037606  001744                          BEQ     CTRAP
10917                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
10918                                                   ;           CONDITIONAL BRANCH INST. AND    <====
10919                                                   ;           REPLACE THE MOVE INSTRUCTION    <====
10920                                                   ;           WHICH FOLLOWS W/ 654            <====
10921  037610                                  AUTO1:
10922  037610  012742  001126                          MOV     #1126,-(R2)     ;MOVE TO MAILBOX #  *******  1126  *******
10923  037614  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10924  037616  000000                                  HALT                    ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10925  037620  012767  000006  140156    TRAPB:  MOV     #6,4            ;RESET TRAP CATCHER
10926  037626  005067  140154                  CLR     6               ;RESET TRAP CATCHER
10927
10928
10929                                   ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10930  037632  005067  000020                  CLR     PROFTE
10931  037636  012706  001000                  MOV     #BUFF,SP        ;SET UP THE STACK POINTER
10932  037642  012767  037660  140134          MOV     #DL11W,4        ;SET UP THE TRAP VECTOR
10933  037650  005767  137710                  TST     TPS             ;TEST THE PUNCH STATUS REGISTER
10934  037654  000403                          BR      DL11W1
10935  037656  000000                  PROFTE: 000000
10936  037660  005267  177772          DL11W:  INC     PROFTE          ;INCR IF NO DL11W
10937  037664  012767  000006  140112  DL11W1: MOV     #6,4
10938
10939  037672                                  SKP104:
10940                                   ;******************************************************************************************
10941                                   ;TEST 375       TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
10942                                   ;******************************************************************************************
10943  037672  005212                  TS375:  INC     (R2)            ;UPDATE TEST NUMBER
10944  037674  022712  000375                  CMP     #375,(R2)       ;SEQUENCE ERROR?
10945  037700  001037                          BNE     TDEC8           ;BR TO ERROR HALT ON SEQ ERROR
10946  037702  005767  177750                  TST     PROFTE
10947  037706  001047                          BNE     R7TRX
10948  037710  122767  000001  140402          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
10949  037716  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
10950  037720  005767  140362                  TST     $PASS                   ;IS THIS THE FIRST PASS?
10951  037724  001040                          BNE     R7TRX                   ;IF NOT FIRST PASS, SKIP TEST
10952  037726                                  2$:
10953  037726  000005                          RESET                           ;
10954  037730  012767  000340  140040          MOV     #340,STATUS     ;LOCK OUT INTERRUPT
10955  037736  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10956  037742  012767  040010  140034          MOV     #TDEC77,4       ;SET UP OVERFLOW TRAP
10957  037750  012767  040000  140106          MOV     #TDEC8,64       ;SET UP INTERRUPT VECTOR
10958  037756  012767  000100  137600          MOV     #100,TTCSR      ;SET INTERRUPT ENABLE
```

```
10959  037764  005067  140006                  CLR     STATUS          ;ALLOW INTERRUPT TO OCCUR
10960  037770  012742  001127                  MOV     #1127,-(R2)     ;MOVE TO MAILBOX #  *******  1127  *******
10961  037774  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10962  037776  000000                          HALT                    ;NO INTERRUPT OCCURRED
10963  040000                          TDEC8:
10964  040000  012742  001130                  MOV     #1130,-(R2)     ;MOVE TO MAILBOX #  *******  1130  *******
10965  040004  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10966  040006  000000                          HALT                    ;OVERFLOW TRAP DID NOT OCCUR OR WRONG $TSTNM
10967  040010  005067  137550          TDEC77: CLR     TTCSR           ;CLEAR INTERRUPT ENABLE
10968  040014  012767  000006  137762          MOV     #6,4
10969  040022  005067  137760                  CLR     6
10970  040026                          R7TRX:
10971                                  ;*****************************************************************************************
10972                                  ;TEST 376       TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
10973                                  ;*****************************************************************************************
10974  040026  005212                  TS376:  INC     (R2)            ;UPDATE TEST NUMBER
10975  040030  022712  000376                  CMP     #376,(R2)       ;SEQUENCE ERROR?
10976  040034  001045                          BNE     BR71            ;BR TO ERROR HALT ON SEQ ERROR
10977  040036  005767  177614                  TST     PROFTE
10978  040042  001053                          BNE     NODL
10979  040044  122767  000001  140246          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
10980  040052  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
10981  040054  005767  140226                  TST     $PASS                   ;IS THIS THE FIRST PASS?
10982  040060  001044                          BNE     NODL                    ;IF NOT FIRST PASS, SKIP TEST
10983  040062                          2$:
10984  040062  012706  001000                  MOV     #BUFF,%6
10985  040066  012767  000340  137702          MOV     #340,STATUS     ;SET TO A HIGH PRIORITY LEVEL
10986  040074  012767  040140  137762          MOV     #TR0,64
10987  040102  012767  000100  137454          MOV     #100,TTCSR      ;INTERRUPT FOR TTY PUNCH/PRINTER
10988  040110  012767  040150  137716          MOV     #BR71,34        ;TRAP VECTOR
10989  040116  012767  040160  137740          MOV     #TR2,64         ;TTY VECTOR
10990  040124  012767  000340  137704          MOV     #340,36         ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
10991  040132  005067  137640                  CLR     STATUS          ;SHOULD INTERRUPT AT END OF CLR INST
10992  040136  104400                          TRAP                    ;TTY INTERRUPT SHOULD OVERRIDE TRAP
10993  040140                          TR0:
10994  040140  012742  001131                  MOV     #1131,-(R2)     ;MOVE TO MAILBOX #  *******  1131  *******
10995  040144  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10996  040146  000000                          HALT                    ;TTY SHOULDN'T HAVE INTERRUPTED
10997  040150                          BR71:
10998  040150  012742  001132                  MOV     #1132,-(R2)     ;MOVE TO MAILBOX #  *******  1132  *******
10999  040154  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11000  040156  000000                          HALT                    ;TRAP OCCURRED FIRST,OR WRONG $TSTNM
11001  040160  005067  137652          TR2:    CLR     36
11002  040164  042767  000100  137372          BIC     #100,TTCSR
11003  040172                          NODL:
11004                                  ;*****************************************************************************************
11005                                  ;TEST 377       TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS
11006                                  ;*****************************************************************************************
11007  040172  005212                  TS377:  INC     (R2)            ;UPDATE TEST NUMBER
11008  040174  022712  000377                  CMP     #377,(R2)       ;SEQUENCE ERROR?
11009  040200  001043                          BNE     TR5             ;BR TO ERROR HALT ON SEQ ERROR
11010  040202  005767  177450                  TST     PROFTE
11011  040206  001063                          BNE     NODL1
11012  040210  122767  000001  140102          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11013  040216  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
11014  040220  005767  140062                  TST     $PASS                   ;IS THIS THE FIRST PASS?
```

```
11015  040224  001054                          BNE     NODL1                   ;IF NOT FIRST PASS, SKIP TEST
11016  040226                          2$:
11017  040226  012706  001000                  MOV     #BUFF,%6
11018  040232  012767  000340  137536          MOV     #340,STATUS
11019  040240  012767  000100  137316          MOV     #100,TTCSR
11020  040246  012767  040306  137560          MOV     #TR3,34         ;TRAP
11021  040254  012767  040320  137602          MOV     #TR4,64         ;TTY OUTPUT
11022  040262  012767  000340  137576          MOV     #340,66         ;TTY OUTPUT PRIORITY
11023  040270  012767  040310  137522          MOV     #TR5,20         ;IOT
11024  040276  012767  000340  137516          MOV     #340,22         ;IOT PRIORITY
11025  040304  104400                          TRAP                    ;THE ACT OF TRAPPING LOWER PRIORITY
11026  040306  000004                  TR3:    IOT                     ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
11027  040310                          TR5:
11028  040310  012742  001133                  MOV     #1133,-(R2)     ;MOVE TO MAILBOX #  *******  1133  *******
11029  040314  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11030  040316  000000                          HALT                    ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TSTNM
11031  040320  005067  137476          TR4:    CLR     22              ;CLR IOT PRIORITY
11032  040324  005067  137536                  CLR     66
11033  040330  012767  000036  137476          MOV     #36,34
11034  040336  012767  000066  137520          MOV     #66,64
11035  040344  012767  000022  137446          MOV     #22,20
11036  040352  005067  137206                  CLR     TTCSR
11037  040356                          NODL1:
11038
11039                                   ;*************************************************************************************
11040                                   ;TEST 400       TEST THAT 'RESET' GOES TO OUTSIDE WORLD
11041                                   ;*************************************************************************************
11042  040356  005212                  TS400:  INC     (R2)            ;UPDATE TEST NUMBER
11043  040360  022712  000400                  CMP     #400,(R2)       ;SEQUENCE ERROR?
11044  040364  001021                          BNE     TS401-10        ;BR TO ERROR HALT ON SEQ ERROR
11045  040366  005767  177264                  TST     PROFTE
11046  040372  001022                          BNE     NODL2
11047  040374  016700  137162                  MOV     TKB,R0          ;MAKE SURE RECEIVER DONE IS SET
11048  040400  012767  000100  137152          MOV     #100,TRCSR      ;SET INTERRUPT ENABLE
11049  040406  000005                          RESET                   ;SHOULD CLEAR INTERRUPT ENABLE
11050  040410  032767  000100  137142          BIT     #100,TRCSR      ;TEST FOR CLEAR
11051  040416  001410                          BEQ     TS401
11052                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
11053                                                                   ;           CONDITIONAL BRANCH INST. AND    <====
11054                                                                   ;           REPLACE THE MOVE INSTRUCTION    <====
11055                                                                   ;           WHICH FOLLOWS W/ 762            <====
11056  040420  012742  001134                  MOV     #1134,-(R2)     ;MOVE TO MAILBOX #  *******  1134  *******
11057  040424  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11058  040426  000000                          HALT                    ;RESET FAILED TO CLEAR TRCSR
11059                                                                   ;  OR SEQUENCE ERROR
11060  040430  012742  001135                  MOV     #1135,-(R2)     ;MOVE TO MAILBOX #  *******  1135  *******
11061  040434  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11062  040436  000000                          HALT                    ; WRONG $TSTNM
11063  040440                          NODL2:
11064                                   ;*************************************************************************************
11065                                   ;TEST 401       TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
11066                                   ;*************************************************************************************
11067  040440  005212                  TS401:  INC     (R2)            ;UPDATE TEST NUMBER
11068  040442  022712  000401                  CMP     #401,(R2)       ;SEQUENCE ERROR?
11069  040446  001014                          BNE     RESET3          ;BR TO ERROR HALT ON SEQ ERROR
11070  040450  012706  001000                  MOV     #BUFF,%6        ;SET STACK
```

```
11071  040454  012767  040510  137332          MOV     #RESET2,14      ;SET UP TRACE VECTOR
11072  040462  012746  000020                  MOV     #20,-(R6)       ;SET THE T-BIT ON STACK
11073  040466  012746  040474                  MOV     #1$,-(R6)       ;MOVE NEW PC ON STACK
11074  040472  000006                          RTT
11075  040474  000005                  1$:     RESET                   ;SHOULD HAVE NO EFFECT
11076  040476  000005                          RESET                   ;NO EFFECT
11077  040500                          RESET3:
11078  040500  012742  001136                  MOV     #1136,-(R2)     ;MOVE TO MAILBOX #  *******  1136  *******
11079  040504  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11080  040506  000000                          HALT                    ;TRACE TRAP FAILED,OR WRONG $TSTNM
11081  040510  005067  137262          RESET2: CLR     STATUS          ;CLEAR TRACK
11082  040514  005067  137276                  CLR     16              ;TRACE STATUS
11083  040520  012767  000016  137266          MOV     #16,14
11084  040526                          SKTST2:
11085                                  ;*************************************************************************************
11086                                  ;TEST 402       TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
11087                                  ;*************************************************************************************
11088  040526  005212                  TS402:  INC     (R2)            ;UPDATE TEST NUMBER
11089  040530  022712  000402                  CMP     #402,(R2)       ;SEQUENCE ERROR?
11090  040534  001057                          BNE     TTY11           ;BR TO ERROR HALT ON SEQ ERROR
11091  040536  005767  177114                  TST     PROFTE
11092  040542  001062                          BNE     NODL3
11093  040544  122767  000001  137546          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11094  040552  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
11095  040554  005767  137526                  TST     $PASS                   ;IS THIS THE FIRST PASS?
11096  040560  001053                          BNE     NODL3                   ;IF NOT FIRST PASS, SKIP TEST
11097  040562                          2$:
11098  040562  000005                          RESET
11099  040564  012706  001000                  MOV     #BUFF,%6        ;SET UP STACK
11100  040570  012767  040614  137266          MOV     #TTY3,64        ;INTERRUPT VECTOR
11101  040576  005067  137174                  CLR     STATUS          ;DROP PROCESSOR PRIORITY
11102  040602  012767  000357  137256          MOV     #357,66         ;HIGH PRIORITY ON INTERRUPT
11103  040610  105167  136750                  COMB    TTCSR           ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
11104  040614  026727  137156  000357  TTY3:   CMP     STATUS,#357
11105  040622  001404                          BEQ     1$
11106                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11107                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11108                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11109                                                                   ;           WHICH FOLLOWS W/ 744           <====
11110  040624  012742  001137                  MOV     #1137,-(R2)     ;MOVE TO MAILBOX #  *******  1137  *******
11111  040630  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11112  040632  000000                          HALT                    ;INTERRUPT DID NOT POP CORRECT STATUS
11113  040634  000005                  1$:     RESET                   ;CLR NTERRUPT ENABLE
11114  040636  012706  001000                  MOV     #BUFF,%6        ;STACK SET UP
11115  040642  012767  040666  137214          MOV     #TTY4,64        ;INTERRUPT VECTOR
11116  040650  005067  137212                  CLR     66              ;CLR NEW STATUS
11117  040654  012767  000157  137114          MOV     #157,STATUS     ;PROCESSOR STATUS
11118  040662  105167  136676                  COMB    TTCSR           ;SET INTERRUPT ENABLE
11119  040666  005767  137104          TTY4:   TST     STATUS
11120  040672  001404                          BEQ     TTT37
11121                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11122                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11123                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11124                                                                   ;           WHICH FOLLOWS W/ 720           <====
11125  040674                          TTY11:
11126  040674  012742  001140                  MOV     #1140,-(R2)     ;MOVE TO MAILBOX #  *******  1140  *******
```

```
11127  040700  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11128  040702  000000                    HALT                    ;INCORRECT STATUS,OR WRONG $TSTNM
11129  040704  105167  136654    TTT37:  COMB    TTCSR
11130  040710                    NODL3:
11131
11132                                    ;*************************************************************************************
11133                                    ;TEST 403       TEST THE 'WAIT' INSTRUCTION
11134                                    ;*************************************************************************************
11135  040710  005212            TS403:  INC     (R2)            ;UPDATE TEST NUMBER
11136  040712  022712  000403            CMP     #403,(R2)       ;SEQUENCE ERROR?
11137  040716  001062                    BNE     STP4            ;BR TO ERROR HALT ON SEQ ERROR
11138  040720  122767  000001  137372    CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11139  040726  001003                    BNE     1$              ;IF NOT, DO THIS TEST
11140  040730  005767  137352            TST     $PASS           ;IS THIS THE FIRST PASS?
11141  040734  001057                    BNE     STP4E           ;IF NOT FIRST PASS, SKIP TEST
11142  040736                    1$:
11143  040736  042767  000100  136620    BIC     #100,TPS        ;CLEAR INTERRUPT ENABLE
11144  040744  012706  001000            MOV     #BUFF,SP        ;SET UP THE STACK
11145  040750  012767  041036  137106    MOV     #WATE,64        ;SET UP THE INTERRUPT VECTOR
11146  040756  005067  137104            CLR     66
11147  040762  105767  136576    WATE1:  TSTB    TPS             ;WAIT FOR READY
11148  040766  100375                    BPL     WATE1           ;TO BE UP
11149  040770  012767  000015  136570    MOV     #15,TPB         ;DO A CARRIAGE RETURN
11150  040776  105767  136562    WATE2:  TSTB    TPS             ;WAIT FOR READY TO COME UP
11151  041002  100375                    BPL     WATE2
11152  041004  012767  000015  136554    MOV     #15,TPB         ;DO ANOTHER CARRIAGE RETURN
11153  041012  052767  000100  136544    BIS     #100,TPS        ;SET THE INTERRUPT ENABLE
11154  041020  005067  136752            CLR     STATUS          ;CLEAR THE PSW
11155  041024  000001            WATE3:  WAIT                    ;WAIT FOR THE INTERRUPT
11156  041026  012742  00114i            MOV     #1141,-(R2)     ;MOVE TO MAILBOX #  *******  1141  *******
11157  041032  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11158  041034  000000                    HALT                    ;WAIT INSTRUCTION DID NOT LOOP
11159  041036  005767  136734    WATE:   TST     STATUS  ;IS THE PSW CORRECT?
11160  041042  001404                    BEQ     1$
11161                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11162                                                            ;           CONDITIONAL BRANCH INST. AND   <====
11163                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
11164                                                            ;           WHICH FOLLOWS W/ 725           <====
11165  041044  012742  001142            MOV     #1142,-(R2)     ;MOVE TO MAILBOX #  *******  1142  *******
11166  041050  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11167  041052  000000                    HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
11168  041054  026727  137714  041026  1$:     CMP     BUFF-4,#WATE3+2 ;IS THE OLD PC SAVED
11169  041062  001404                    BEQ     STP4E
11170                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11171                                                            ;           CONDITIONAL BRANCH INST. AND   <====
11172                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
11173                                                            ;           WHICH FOLLOWS W/ 715           <====
11174  041064                    STP4:
11175  041064  012742  001143            MOV     #1143,-(R2)     ;MOVE TO MAILBOX #  *******  1143  *******
11176  041070  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11177  041072  000000                    HALT                    ;OLD PC WAS NOT SAVED OR WRONG $TESTN
11178  041074                    STP4E:
11179                                    ;*************************************************************************************
11180                                    ;TEST 404       TEST THAT USING REGISTER ADDR (177700) CAUSES TIME OUT.
11181                                    ;*************************************************************************************
11182  041074  005212            TS404:  INC     (R2)            ;UPDATE TEST NUMBER
```

```
11183  041076  022712  000404                  CMP     #404,(R2)       ;SEQUENCE ERROR?
11184  041102  001017                          BNE     TS405-10        ;BR TO ERROR HALT ON SEQ ERROR
11185                                          ;
11186                                          ;REGISTER ADDRESS (177700-177717) CAUSE TIME OUT WHEN USED
11187                                          ;AS PROGRAM ADDRESS BY THE CPU.
11188                                          ;
11189  041104  012706  001000                  MOV     #BUFF,SP        ;SET  STACK POINTER
11190  041110  012737  041132  000004          MOV     #RETR1,@#RTRAP5 ;SET TRAP RETURN ADDR
11191  041116  005737  177700          PCN1:   TST     @#177700        ;BAD ADDR REFERENCE, TRAP TO 4
11192  041122  012742  001144                  MOV     #1144,-(R2)     ;MOVE TO MAILBOX #  *******  1144  *******
11193  041126  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11194  041130  000000                          HALT                    ;REFERENCING 177700 DID NOT CAUSE TIME OUT
11195  041132  022767  041122  137634  RETR1:  CMP     #PCN1+4,BUFF-4  ;PROPER PC STORED ON STACK?
11196  041140  001404                          BEQ     TS405
11197                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11198                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11199                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11200                                                                  ;           WHICH FOLLOWS W/ 760           <====
11201  041142  012742  001145                  MOV     #1145,-(R2)     ;MOVE TO MAILBOX #  *******  1145  *******
11202  041146  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11203  041150  000000                          HALT                    ;OLD PC WAS NOT SAVED IN STACK
11204                                                                  ;  OR SEQUENCE ERROR
11205
11206                                  ;*****************************************************************
11207                                  ;ODD ADDRESS USED BY A 'WORD'' INSTRUCTION SHOULD NOT
11208                                  ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
11209                                  ;*****************************************************************************
11210                                  ;TEST 405       TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
11211                                  ;*****************************************************************************
11212  041152  005212                  TS405:  INC     (R2)            ;UPDATE TEST NUMBER
11213  041154  022712  000405                  CMP     #405,(R2)       ;SEQUENCE ERROR?
11214  041160  001013                          BNE     TS406-10        ;BR TO ERROR HALT ON SEQ ERROR
11215                                          ;
11216  041162  012737  041210  000004          MOV     #RETR2,@#RTRAP5 ;SET TRAP RETURN ADDR
11217  041170  005037  000000                  CLR     @#0             ;PUT ALL 0 IN LOC 0
11218  041174  005337  000001                  DEC     @#1             ;DECREMENT ODD ADDRESS, SHOULD NOT TRAP
11219  041200  022737  177777  000000          CMP     #-1,@#0         ;WORD LOC 0 HAS ALL ONES?
11220  041206  001404                          BEQ     TS406
11221                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11222                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11223                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11224                                                                  ;           WHICH FOLLOWS W/ 764           <====
11225  041210                          RETR2:
11226  041210  012742  001146                  MOV     #1146,-(R2)     ;MOVE TO MAILBOX #  *******  1146  *******
11227  041214  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11228  041216  000000                          HALT                    ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP
11229                                                                  ;  OR SEQUENCE ERROR
11230
11231                                  ;*************************************************************
11232                                  ;
11233                                  ;USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
11234                                  ;THE REGISTER IN USE WILL BE INCREMENTED.
11235                                  ;
11236                                  ;*****************************************************************************
11237                                  ;TEST 406       TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
11238                                  ;*****************************************************************************
```

```
11239  041220  005212                          TS406:  INC     (R2)              ;UPDATE TEST NUMBER
11240  041222  022712  000406                          CMP     #406,(R2)         ;SEQUENCE ERROR?
11241  041226  001016                                  BNE     TS407-10          ;BR TO ERROR HALT ON SEQ ERROR
11242  041230  012737  041256  000004                  MOV     #RETR3,@#RTRAP5   ;SET TRAP RETURN ADDR
11243  041236  012700  177700                          MOV     #177700,R0        ;STORES BAD MEMORY REFERENCE
11244  041242  012720  001234                          MOV     #1234,(R0)+       ;BAD ADDR REFERENCE, TRAP TO LOC 4
11245  041246  012742  001147                          MOV     #1147,-(R2)       ;MOVE TO MAILBOX #  *******  1147  *******
11246  041252  005242                                  INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
11247  041254  000000                                  HALT                      ;ADDRESSING 177700 DID NOT CAUSE TRAP
11248  041256  022700  177702                  RETR3:  CMP     #177702,R0        ;WAS R0 INCREMENTED?
11249  041262  001404                                  BEQ     TS407
11250                                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11251                                                                            ;           CONDITIONAL BRANCH INST. AND   <====
11252                                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
11253                                                                            ;           WHICH FOLLOWS W/ 761           <====
11254  041264  012742  001150                          MOV     #1150,-(R2)       ;MOVE TO MAILBOX #  *******  1150  *******
11255  041270  005242                                  INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
11256  041272  000000                                  HALT                      ;R0 WAS NOT INCREMENTED
11257                                                                            ;  OR SEQUENCE ERROR
11258                                  ;*****************************************************************
11259                                  ;
11260                                  ;AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
11261                                  ;TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
11262                                  ;WAS BAD, A BOUBLE BUS ERROR OCCURED.  THE STACK POINTER WOULD
11263                                  ;THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
11264                                  ;LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
11265                                  ;EXECUTION.
11266                                  ;
11267                                  ;*************************************************************************************************
11268                                  ;TEST 407       TEST FOR DOUBLE BUS ERROR.
11269                                  ;*************************************************************************************************
11270  041274  005212                  TS407:  INC     (R2)              ;UPDATE TEST NUMBER
11271  041276  022712  000407                  CMP     #407,(R2)         ;SEQUENCE ERROR?
11272  041302  001054                          BNE     TS410-10          ;BR TO ERROR HALT ON SEQ ERROR
11273  041304  012737  041362  000004          MOV     #DBE1,@#RTRAP5    ;SET TRAP RETURN ADDR
11274  041312  012737  000340  000006          MOV     #340,@#6          ;SET UP PS
11275  041320  012737  041352  000010          MOV     #DBE2,@#RTRAP     ;SET TRAP RETURN ADDR
11276  041326  012737  000340  000012          MOV     #340,@#12         ;SET UP PS
11277  041334  012706  177700                  MOV     #177700,SP        ;SET ILLEGAL SP
11278  041340  000077                  DBE:    TRAPA                     ;ILLEGAL INSTRUCTION
11279  041342  012742  001151                  MOV     #1151,-(R2)       ;MOVE TO MAILBOX #  *******  1151  *******
11280  041346  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
11281  041350  000000                          HALT                      ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
11282  041352                          DBE2:
11283  041352  012742  001152                  MOV     #1152,-(R2)       ;MOVE TO MAILBOX #  *******  1152  *******
11284  041356  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
11285  041360  000000                          HALT                      ;TRAP TO WRONG LOCATION
11286  041362  022737  041342  000000  DBE1:   CMP     #DBE+2,@#0        ;OLD PC GOT SAVED?
11287  041370  001404                          BEQ     DBE3
11288                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11289                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
11290                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
11291                                                                    ;           WHICH FOLLOWS W/ 744           <====
11292  041372  012742  001153                  MOV     #1153,-(R2)       ;MOVE TO MAILBOX #  *******  1153  *******
11293  041376  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
11294  041400  000000                          HALT                      ;OLD PC DID NOT GET SAVEDD
```

```
11295  041402  022737  000340  000002  DBE3:   CMP     #340,@#2        ;CORRECT PS SAVED?
11296  041410  001404                  DBE4    BEQ     DBE4
11297                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11298                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11299                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11300                                                                   ;           WHICH FOLLOWS W/ 734           <====
11301  041412  012742  001154                  MOV     #1154,-(R2)     ;MOVE TO MAILBOX #  *******  1154  *******
11302  041416  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11303  041420  000000                          HALT                    ;CORRECT PS DID NOT GET SAVE
11304  041422  022706  000000          DBE4:   CMP     #0,SP           ;SP POINTS TO LOC 0?
11305  041426  001404                          BEQ     DBE5
11306                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11307                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11308                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11309                                                                   ;           WHICH FOLLOWS W/ 725           <====
11310  041430  012742  001155                  MOV     #1155,-(R2)     ;MOVE TO MAILBOX #  *******  1155  *******
11311  041434  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11312  041436  000000                          HALT                    ;SP IS NOT POINTING TO LOC 0
11313  041440  012706  001000          DBE5:   MOV     #STBOT,SP       ;RESET SP
11314
11315
11316                                   ;*************************************************************************************
11317                                   ;TEST 410       TEST MFPT
11318                                   ;*************************************************************************************
11319  041444  005212                  TS410:  INC     (R2)            ;UPDATE TEST NUMBER
11320  041446  022712  000410                  CMP     #410,(R2)       ;SEQUENCE ERROR?
11321  041452  001023                          BNE     TS411-10        ;BR TO ERROR HALT ON SEQ ERROR
11322
11323                                   ;THIS TESTS THE MFPT INSTRUCTION- MOVE FROM PROCESSOR TYPE
11324                                   ;UPON EXECUTION, R0 WILL RECEIVE THE PROCESSOR MODEL CODE
11325          000007                   ;WHICH IS ''000003'' FOR F11.
                                        MFPT=000007
11326  041454  012706  001000                  MOV     #STBOT,R6       ;INIT. SP
11327  041460  013746  000010                  MOV     @#10,-(SP)      ;SAVE TRAP VECTOR
11328  041464  012737  041514  000010          MOV     #1$,@#10        ;SET UP ILLEGAL INSTRUCTION TRAP
11329  041472  010046                          MOV     R0,-(SP)        ;SAVE R0
11330  041474  000007                          MFPT                    ;GET PROCESSOR MODEL
11331  041476  010037  041524                  MOV     R0,@#CPUTYP     ;STORE IT
11332  041502  012600                          MOV     (SP)+,R0        ;RESTORE R0
11333  041504  022737  000003  041524          CMP     #3,@#CPUTYP     ;CHECK MODEL TYPE
11334  041512  001405                          BEQ     XXT
11335                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11336                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11337                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11338                                                                   ;           WHICH FOLLOWS W/ 757           <====
11339  041514                          1$:
11340  041514  012742  001156                  MOV     #1156,-(R2)     ;MOVE TO MAILBOX #  *******  1156  *******
11341  041520  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11342  041522  000000                          HALT                    ;ILLEGAL INSTR TRAP OR WRONG MODEL TYPE
11343
11344  041524  000000                  CPUTYP: .WORD   0
11345  041526                          XXT:
11346  041526  012637  000010                  MOV     (SP)+,@#10      ;RESTORE TRAP VECTOR
11347                                   ;*************************************************************
11348                                   ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
11349                                   ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
11350                                   ;CONTROL-CHIP.  ON THE KDF11-A, THIS INCLUDES: FIS(CTL3), CIS AND WCS
```

```
11351                                   ;INSTRUCTIONS.  A CTLERR TRAPS TO LOCATION 10.
11352                                   ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
11353                                   ;*****************************************************************************************
11354                                   ;TEST 411       TEST CTLERR SERVICE ROUTINE
11355                                   ;*****************************************************************************************
11356  041532  005212                  TS411:  INC     (R2)            ;UPDATE TEST NUMBER
11357  041534  022712  000411                  CMP     #411,(R2)       ;SEQUENCE ERROR?
11358  041540  001022                          BNE     TS412-10        ;BR TO ERROR HALT ON SEQ ERROR
11359
11360  041542  012706  001000                  MOV     #STBOT,R6       ;INIT STACK POINTER
11361  041546  012737  041566  000010          MOV     #1$,@#10                ;SET UP RETURN ADDR FROM TRAP
11362  041554  012737  000340  000012          MOV     #340,@#12       ;SET TRAP PRIORITY=7
11363  041562  075006                          FADD    R6              ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
11364  041564  000000                          HALT                    ;DID NOT TRAP..CHECK CSEL LINE
11365  041566  012737  041600  000010  1$:     MOV     #2$,@#10                ;SET UP RETURN ADDR FROM TRAP
11366  041574  076000                          76000                   ;EXECUTE CIS INSTR..SHOULD TRAP
11367  041576  000000                          HALT                    ;DID NOT TRAP
11368  041600  012737  041612  000010  2$:     MOV     #3$,@#10                ;SET UP RETURN ADDR FROM TRAP
11369  041606  076700                          76700                   ;EXECUTE WCS INSTR..SHOULD CAUSE CTLERR AND TRAP
11370  041610  000000                          HALT                    ;DID NOT TRAP
11371  041612  012706  001000          3$:     MOV     #STBOT,R6       ;RE-INIT STACK POINTER
11372
11373                                   ;*****************************************************************************************
11374                                   ;TEST 412       TEST THAT ALL RESERVED INSTRUCTIONS TRAP
11375                                   ;*****************************************************************************************
11376  041616  005212                  TS412:  INC     (R2)            ;UPDATE TEST NUMBER
11377  041620  022712  000412                  CMP     #412,(R2)       ;SEQUENCE ERROR?
11378  041624  001112                          BNE     RET4            ;BR TO ERROR HALT ON SEQ ERROR
11379  041626  042767  000100  135730          BIC     #100,TPS
11380  041634  012737  041662  000244          MOV     #TRAP244,@#244  ; SET UP TO SEE IF
11381  041642  013767  000010  000024          MOV     @#10,TENSAVE    ;  THIS PROCESSOR HAS THE
11382  041650  012737  041672  000010          MOV     #TRAP10,@#10    ;  FLOATING POINT OPTION
11383  041656  170007                          .WORD   170007          ; AN ILLEGAL FPP INSTRUCTION
11384  041660  000406                          BR      AROUND          ; THE FOLLOWING
11385  041662                          TRAP244:                        ; IF FPP IN--
11386  041662  013767  042114  000234          MOV     @#FPP,FINISH    ;  RESET END OF TABLE POINTER
11387  041670  000002                          RTI                     ; AND RETURN
11388  041672                          TRAP10:                         ; LEAVE THE TABLE ALONE
11389  041672  000002                          RTI                     ; AND RETURN
11390  041674  000000                  TENSAVE:.WORD   0               ; A PLACE TO STORE CONTENTS OF 10
11391
11392  041676                          AROUND:                         ;  CONTINUATION POINT
11393  041676  012737  000246  000244          MOV     #246,@#244      ;  RESTORE THE TRAP VECTOR
11394  041704  016737  177764  000010          MOV     TENSAVE,@#10    ; RESTORE THE ILLEGAL INST. VECTOR
11395  041712  012703  042074                  MOV     #TABLE,TAB      ;TABLE POINTER
11396  041716  012305                  GIN1:   MOV     (TAB)+,FIRST    ;FIRST OR CURRENT INSTRUCTION
11397  041720  012301                          MOV     (TAB)+,LAST     ;LAST INSTRUCTION OR GROUP
11398  041722  020567  000176                  CMP     FIRST,FINISH    ;TESTED ALL
11399  041726  001415                          BEQ     GIN3            ;YES BRANCH
11400  041730  010567  000172                  MOV     FIRST,INST      ;SET UP INST
11401  041734  005267  000166          GIN2:   INC     INST
11402  041740  012767  042006  136042          MOV     #RET,10         ;SET UP RETURN FROM TRAP
11403  041746  012706  001000                  MOV     #BUFF,SP        ;SET UP STACK POINTER
11404  041752  005067  136020                  CLR     CC              ;CLEAR PRIORITY
11405  041756  000167  000144                  JMP     INST            ;EXECUTE RESERVED INSTRUCTION
11406  041762  012737  062234  000034  GIN3:   MOV     #$TRAP,@#34     ;TRAP VECTOR FOR TRAP CALL
```

```
11407  041770  012737  000340  000036          MOV     #340,@#36       ;LEVEL 7
11408  041776  012700  000370                  MOV     #370,R0         ;RESET RESERVED AREA 370-402
11409  042002  000167  000234                  JMP     THRPRT          ;JUMP TO EIS TEST
11410
11411                                  ;TRAPPING SHOULD SEND YOU HERE
11412  042006  020627  000774          RET:    CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
11413  042012  001404                          BEQ     RET1
11414  042014  012742  001157                  MOV     #1157,-(R2)     ;MOVE TO MAILBOX #  *******  1157  *******
11415  042020  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11416  042022  000000                          HALT                    ;WRONG DECREMENT
11417  042024  026727  136744  042130  RET1:   CMP     BUFF-4,#INST+2  ;LOC OF INST UNINCREMENTED
11418  042032  001404                          BEQ     RET2
11419  042034  012742  001160                  MOV     #1160,-(R2)     ;MOVE TO MAILBOX #  *******  1160  *******
11420  042040  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11421  042042  000000                          HALT                    ;INST INC ON TRAP
11422  042044  005767  136726          RET2:   TST     BUFF-2
11423  042050  001404                          BEQ     RET3
11424                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11425                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11426                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11427                                                                  ;           WHICH FOLLOWS W/ 665           <====
11428  042052                          RET4:
11429  042052  012742  001161                  MOV     #1161,-(R2)     ;MOVE TO MAILBOX #  *******  1161  *******
11430  042056  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11431  042060  000000                          HALT                    ;CONDITION CODES SET ON TRAP OR WRONG $TSTNM
11432  042062  026701  000040          RET3:   CMP     INST,LAST
11433  042066  001713                          BEQ     GIN1            ;SET UP NEW GROUP
11434  042070  000167  177640                  JMP     GIN2            ;FINISH OLD GROUP
11435                                                                  ;END OF INSTRUCTION GROUP
11436  042074  000007                  TABLE:  7                       ;END OF OPERATE
11437  042076  000077                          77
11438  042100  000207                          207                     ;RTS,RT1,JMP
11439  042102  000227                          227
11440  042104  006777                          6777
11441  042106  007777                          7777
11442  042110  075037                          075037
11443  042112  076777                          76777
11444  042114  167777                  FPP:    167777                  ;  START OF THE FPP INSTRUCTIONS
11445  042116  177700                          177700
11446  042120  177716                          177716
11447  042122  177777                          177777
11448  042124  042124                  FINISH: .                       ;END FLAG
11449  042126  000000                  INST:   HALT                    ;WILL CONTINUE RESERVED INST
11450  042130  000000                          HALT                    ;SHOULD TRAP TO LOC 10
11451  042132  000000                          HALT                    ;LOC 10 SHOULD SEND YOU TO
11452  042134  000000                          HALT                    ;RET
11453  042136  000000                          HALT
11454                                  .SBTTL ** STARTING OF EIS TEST **
11455          000000                  .REPT   0
11456
11457
11458                                  .PAGE
11459
11460                                          PART THREE:             EIS INSTRUCTION TESTS
11461
11462
```

```
                                        ABSTRACT

                                        THIS PROGRAM TESTS THE F11 EXTENDED INSTRUCTION SET
                                        <ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT-
                                        LEAST ONCE WITH EACH INSTRUCTION.



                                        SWITCH SETTINGS

                                        IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM
                                        AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE
                                        SWITCH REGISTER.  THE USER SHOULD SET THIS LOCATION BEFORE
                                        STARTING THE PROGRAM.


                                        BIT #   OCTAL VALUE             FUNCTION

                                        15      100000..... HALT ON ERROR
                                        13      020000..... INHIBIT ERROR PRINTOUT

                                        AN 8 BIT BYTE $ENVM [I.E. LOCATION 321] HAS BEEN USED TO DEFINE
                                        THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING
                                        BIT 5 OF BYTE $ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN
                                        LOCATION 320

                                .ENDR


                                .ABS
                                .NLIST  MD,MC,CND
                                .LIST   ME
```

```
11497
11498
11499             000000                  DUMMY=  0
11500             000001                  ERRNM=  1
11501             000051                  F=      51
11502             000176                  N=      176
11503             001000                  SW09=   1000
11504             002000                  SW10=   2000
11505             004000                  SW11=   4000
11506             010000                  SW12=   10000
11507
11508  042140                             COUNT:
11509             042142                          .=COUNT+2
11510  042142                             PSWORD:
11511             042144                          .=PSWORD+2
11512  042144                             TEMP1:
11513             042146                          .=TEMP1+2
11514  042146                             TEMP2:
11515             042150                          .=TEMP2+2
11516  042150                             TEMP3:
11517             042152                          .=TEMP3+2
11518  042152                             TEMP4:
11519             042154                          .=TEMP4+2
11520  042154  000000                     TEMP5:  .WORD
11521  042156  000000                     TEMP6:  .WORD
11522  042160     000                     TYPCNT: .BYTE
11523  042161     000                     $TPCNT: .BYTE
11524  042162  000007                     S0:     7
11525  042164  177771                     S1:     -7
11526  042166  042164                     S2:     S1
11527  042170  177772                     S3:     -6
11528  042172  177777                     S4:     -1
11529  042174  040000                     S5:     40000
11530  042176  042174                     S6:     S5
11531  042200  040000                     S7:     40000
11532  042202  177776                     S8:     -2
11533  042204  000002                     S9:     2
11534  042206  042204                     S10:    S9
11535  042210  000002                     S11:    2
11536  042212  177570                     SWR:    177570
11537  042214  177570                     DISPLAY:        177570
11538  042216  000064                     TTYOUT: 64
11539  042220  177566                     $TPB:   177566
11540  042222  177564                     $TPS:   177564
11541  042224  005015  020040  000040     $CRLF:  .ASCIZ  <15><12>/   /
11542  042232  006412  047520  042527     POWER:  .ASCIZ  <12><15>/POWER/
11543  042240  000122
11544
11545
11546
11547
11548
11549
11550
11551
11552
```

```
                                                                    I  2
 11553
```

```
11554  042242                          THRPRT:
11555
11556  042242  012705  000304                  MOV     #$TESTN,R5      ;MAKE R5 POINT TO THE LOCATION $TESTN
11557  042246  005037  042140                  CLR     @#COUNT         ;CLEAR THE COUNTER
11558  042252  012715  000001                  MOV     #1,(R5)         ;INITIALIZE TEST NUMBER
11559  042256  012706  001000                  MOV     #STBOT,SP       ;** STACK AT STBOT **
11560  042262  013746  000004                  MOV     @#4,-(SP)       ;SAVE ERROR VECTOR
11561  042266  013746  000006                  MOV     @#6,-(SP)       ;
11562  042272  012767  042306  135504          MOV     #1$,4           ;SET UP TIMEOUT VECTOR
11563  042300  005777  177706                  TST     @SWR            ;TRY TO REFERENCE HARDWRE SWR
11564  042304  000407                          BR      3$              ;BRANCH IF NO TIMEOUT TRAP OCCURS
11565  042306  012767  000176  177676  1$:     MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
11566  042314  012767  000174  177672          MOV     #DISPREG,DISPLAY        ;POINT TO SOFTWARE DISPLAY REG
11567  042322  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
11568  042324  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR
11569  042330  012637  000004                  MOV     (SP)+,@#4       ;
11570  042334  106427  000000                  MTPS    #0              ;PLACE #0 IN PSW
11571  042340  132737  000001  000320          BITB    #1,@#$ENV       ;ARE WE UNDER APT ?
11572  042346  001403                          BEQ     2$              ;IF NOT THEN GO TO 2$
11573  042350  012767  000322  177634          MOV     #$SWREG,SWR     ;USE APT SWITCH REGISTER
11574  042356  012737  000001  042144  2$:     MOV     #1,@#TEMP1      ;TEMP1=1
11575  042364  005037  042146                  CLR     @#TEMP2         ;TEMP2=0
11576  042370  012737  000001  042150          MOV     #1,@#TEMP3      ;TEMP3=1
11577  042376  005037  042152                  CLR     @#TEMP4         ;TEMP4=0
11578
11579
```

```
11580
11581                                   ;*********************************
11582                                   ;     ASH INSTRUCTION TESTS
11583                                   ;*********************************
11584
11585
11586
11587
11588                                           ;*************
11589                                           ; TESTS 1-36
11590                                           ;*************
11591
11592
11593  042402  010767  135564          ASTART: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11594  042406  013700  042144                  MOV     @#TEMP1,%0      ;LOAD R0 WITH THE CONTENTS OF TEMP1
11595  042412  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11596  042420  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11597  042422  013701  042146                  MOV     @#TEMP2,R1      ;OTHERWISE EXECUTE THE INSTRUCTION
11598                                                                  ;IN MODE 0 USING R1
11599  042426  072001                          ASH     R1,R0
11600  042430  000402                          BR      4$
11601  042432  072067  177510          2$:     ASH     TEMP2,%0        ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
11602  042436  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11603  042442  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11604  042450  001403                          BEQ     .+10
11605  042452  004767  016050                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11606                                                                  ;THE PS IS NOT EQUAL TO 0
11607  042456  000001                          1               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11608                                                          ;BY  (013746  000172  000207)
11609
11610  042460  005237  042140                  INC     @#COUNT         ;INCREMENT THE COUNTER
11611  042464  023700  042150                  CMP     @#TEMP3,%0      ;IS THE RESULT IN R0 EQUAL TO TEMP3?
11612  042470  001403                          BEQ     .+10
11613  042472                          6$:
11614  042472  004767  016030                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11615                                                                  ;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
11616  042476  000002                          2               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11617                                                          ;BY  (013746  000172  000207)
11618
11619  042500  021537  042140                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE
11620                                                                  ;COUNTER?
11621  042504  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11622  042506  005215                          INC     (R5)
11623  042510  010767  135456                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11624  042514  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11625                                                                  ;BY 14. AND RIGHT BY 14.?
11626  042520  002011                          BGE     8$
11627  042522  005237  042146                  INC     @#TEMP2
11628  042526  006367  177416                  ASL     TEMP3           ;SHIFT TEMP3 LEFT.
11629  042532  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11630  042536  001004                          BNE     REGR1
11631  042540  000167  001010                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11632  042544  004767  001032          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11633  042550  010767  135416          REGR1:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11634  042554  013701  042144                  MOV     @#TEMP1,%1      ;LOAD R1 WITH THE CONTENTS OF TEMP1
11635  042560  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
```

```
11636  042566  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11637  042570  013702  042146                  MOV     @#TEMP2,R2      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11638  042574  072102                          ASH     R2,R1           ;USING R1
11639  042576  000402                          BR      4$
11640  042600  072167  177342          2$:     ASH     TEMP2,%1        ;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
11641  042604  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11642  042610  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11643  042616  001403                          BEQ     .+10
11644  042620  004767  015702                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11645                                                          ;THE PS IS NOT FQUAL TO 0
11646  042624  000003                          3               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11647                                                          ;BY  (013746  000172  000207)
11648
11649  042626  005237  042140                  INC     @#COUNT         ;INCREMENT THE COUNTER
11650  042632  023701  042150                  CMP     @#TEMP3,%1      ;IS THE RESULT IN R1 EQUAL TO TEMP3?
11651  042636  001403                          BEQ     .+10
11652  042640                          6$:
11653  042640  004767  015662                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11654                                                          ;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
11655  042644  000004                          4               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11656                                                          ;BY  (013746  000172  000207)
11657
11658  042646  021537  042140                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11659  042652  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11660  042654  005215                          INC     (R5)
11661  042656  010767  135310                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11662  042662  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11663                                                          ;BY 14. AND RIGHT BY 14.?
11664  042666  002011                          BGE     8$
11665  042670  005237  042146                  INC     @#TEMP2
11666  042674  006367  177250                  ASL     TEMP3           ;SHIFT TEMP3 LEFT
11667  042700  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11668  042704  001004                          BNE     REGR2
11669  042706  000167  000642                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11670  042712  004767  000664          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11671  042716  010767  135250          REGR2:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11672  042722  013702  042144                  MOV     @#TEMP1,%2      ;LOAD R2 WITH THE CONTENTS OF TEMP1
11673  042726  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11674  042734  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11675  042736  013703  042146                  MOV     @#TEMP2,R3      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11676  042742  072203                          ASH     R3,R2           ;USING R2
11677  042744  000402                          BR      4$
11678  042746  072267  177174          2$:     ASH     TEMP2,%2        ;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
11679  042752  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11680  042756  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS THE PS - TEMP4 ?
11681  042764  001403                          BEQ     .+10
11682  042766  004767  015534                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11683                                                          ;THE PS IS NOT EQUAL TO 0
11684  042772  000005                          5               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11685                                                          ;BY  (013746  000172  000207)
11686
11687  042774  005237  042140                  INC     @#COUNT
11688  043000  023702  042150                  CMP     @#TEMP3,%2      ;IS THE RESULT IN R2 EQUAL TO TEMP3?
11689  043004  001403                          BEQ     .+10
11690  043006                          6$:
11691  043006  004767  015514                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```
11692
11693  043012  000006                          6               ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
                                                               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11694                                                          ;BY  (013746  000172  000207)
11695
11696  043014  021537  042140                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11697  043020  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11698  043022  005215                          INC     (R5)
11699  043024  010767  135142                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11700  043030  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11701                                                                  ;LEFT BY 14, AND RIGHT BY 14.?
11702  043034  002011                          BGE     8$
11703  043036  005237  042146                  INC     @#TEMP2
11704  043042  006367  177102                  ASL     TEMP3           ;SHIFTED TEMP3 LEFT
11705  043046  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11706  043052  001004                          BNE     REGR3
11707  043054  000167  000474                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11708  043060  004767  000516          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11709  043064  010767  135102          REGR3:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11710  043070  013703  042144                  MOV     @#TEMP1,%3      ;LOAD R3 WITH THE CONTENTS OF TEMP1
11711  043074  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11712  043102  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11713  043104  013704  042146                  MOV     @#TEMP2,R4      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11714  043110  072304                          ASH     R4,R3           ;USING R3
11715  043112  000402                          BR      4$
11716  043114  072367  177026          2$:     ASH     TEMP2,%3        ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
11717  043120  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11718  043124  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11719  043132  001403                          BEQ     .+10
11720  043134  004767  015366                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11721                                                                  ;THE PS IS NOT EQUAL TO 0.
11722  043140  000007                          7               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11723                                                          ;BY  (013746  000172  000207)
11724
11725  043142  005237  042140                  INC     @#COUNT
11726  043146  023703  042150                  CMP     @#TEMP3,%3      ;IS THE RESULT IN R3 EQUAL TO TEMP3?
11727  043152  001403                          BEQ     .+10
11728  043154                          6$:
11729  043154  004767  015346                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11730                                                                  ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
11731  043160  000010                          10              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11732                                                          ;BY  (013746  000172  000207)
11733
11734  043162  021537  042140                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11735  043166  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11736  043170  005215                          INC     (R5)
11737  043172  010767  134774                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11738  043176  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11739                                                                  ;LEFT BY 14, AND RIGHT BY 14.?
11740  043202  002010                          BGE     8$
11741  043204  005237  042146                  INC     @#TEMP2
11742  043210  006367  176734                  ASL     TEMP3           ;SHIFT TEMP3 LEFT?
11743  043214  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11744  043220  001003                          BNE     REGR4
11745  043222  000554                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11746  043224  004767  000352          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11747  043230  010767  134736          REGR4:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
```

```
11748  043234  013704  042144                  MOV     @#TEMP1,%4      ;LOAD R4 WITH THE CONTENTS OF TEMP1
11749  043240  010501                          MOV     R5,R1           ;SAVE R5
11750  043242  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11751  043250  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11752  043252  013705  042146                  MOV     @#TEMP2,R5      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11753  043256  072405                          ASH     R5,R4           ;USING R4
11754  043260  000402                          BR      4$
11755  043262  072467  176660          2$:     ASH     TEMP2,%4        ;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
11756  043266  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11757  043272  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11758  043300  001403                          BEQ     .+10
11759  043302  004767  015220                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11760                                                                  ;THE PS IS NOT EQUAL TO 0
11761  043306  000011                          11              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11762                                                          ;BY  (013746  000172  000207)
11763
11764  043310  005237  042140                  INC     @#COUNT
11765  043314  023704  042150                  CMP     @#TEMP3,%4      ;IS THE RESULT IN R4 EQUAL TO TEMP3?
11766  043320  001403                          BEQ     .+10
11767  043322                          6$:
11768  043322  004767  015200                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11769                                                                  ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
11770  043326  000012                          12              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11771                                                          ;BY  (013746  000172  000207)
11772
11773  043330  010105                          MOV     R1,R5           ;RESTORE R5
11774  043332  021537  042140                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11775  043336  001371                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11776  043340  005215                          INC     (R5)
11777  043342  010767  134624                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11778  043346  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN
11779                                                                  ;SHIFTED LEFT BY 14. AND RIGHT BY 14.?
11780  043352  002010                          BGE     8$
11781  043354  005237  042146                  INC     @#TEMP2
11782  043360  006367  176564                  ASL     TEMP3           ;SHIFT TEMP3 LEFT
11783  043364  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
11784  043370  001003                          BNE     REGR5
11785  043372  000470                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11786  043374  004767  000202          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE RES1 OF THE PROGRAM
11787  043400  010767  134566          REGR5:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11788  043404  010501                          MOV     R5,R1           ;SAVE R5
11789  043406  013705  042144                  MOV     @#TEMP1,%5      ;LOAD R5 WITH THE CONTENTS OF TEMP1
11790  043412  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11791  043420  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11792  043422  013700  042146                  MOV     @#TEMP2,R0      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11793  043426  072500                          ASH     R0,R5           ;USING R5
11794  043430  000402                          BR      4$
11795  043432  072567  176510          2$:     ASH     TEMP2,%5        ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
11796  043436  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
11797  043442  123737  042152  042142          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11798  043450  001403                          BEQ     .+10
11799  043452  004767  015050                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11800                                                                  ;THE PS IS NOT EQUAL TO 0.
11801  043456  000013                          13              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11802                                                          ;BY  (013746  000172  000207)
11803
```

```
11804  043460  005237  042140                  INC     @#COUNT
11805  043464  023705  042150                  CMP     @#TEMP3,%5      ;IS THE RESULT IN R5 EQUAL TO TEMP3?
11806  043470  001403                          BEQ     .+10
11807  043472                          6$:
11808  043472  004767  015030                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11809                                                                  ;EITHER INCORRECT R5 OR INCORRECT SEQUFNCE
11810  043476  000014                          14              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11811                                                          ;BY  (013746  000172  000207)
11812
11813  043500  021137  042140                  CMP     (R1),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11814  043504  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11815  043506  010105                          MOV     R1,R5           ;RESTORE R5
11816  043510  005215                          INC     (R5)
11817  043512  010767  134454                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11818  043516  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11819                                                                  ;LEFT BY 14. AND RIGHT BY 14.?
11820  043522  002010                          BGE     8$              ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11821  043524  005237  042146                  INC     @#TEMP2
11822  043530  006367  176414                  ASL     TEMP3           ;SHIFT TEMP3 LEFT
11823  043534  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11824  043540  001405                          BEQ     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11825  043542  000402                          BR      10$
11826  043544  004767  000032          8$:     JSR     PC,TST37
11827  043550  000167  176626          10$:    JMP     ASTART          ;GO BACK TO START
11828  043554  012737  040000  042144  NEGAT:  MOV     #40000,@#TEMP1  ;TEMP1=40000
11829  043562  012737  177762  042146          MOV     #177762,@#TEMP2 ;TEMP2=177762
11830  043570  012737  000001  042150          MOV     #1,@#TEMP3      ;TEMP3=1
11831  043576  000167  176600                  JMP     ASTART
11832  043602  021527  000037          TST37:  CMP     (R5),#37        ;IS IT TEST 37?
11833  043606  001013                          BNE     TST40           ;IF NOT THEN TRY TEST 40
11834  043610  005037  042144                  CLR     @#TEMP1         ;0
11835  043614  012737  000020  042146          MOV     #16.,@#TEMP2    ;SHIFTED BY 16
11836  043622  005037  042150                  CLR     @#TEMP3         ;IS=0
11837  043626  012737  000004  042152          MOV     #4,@#TEMP4      ;AND PS=4
11838  043634  000207                          RTS     PC
11839  043636  021527  000040          TST40:  CMP     (R5),#40        ;IS IT TEST 40?
11840  043642  001003                          BNE     TST41           ;IF NOT THEN TRY TEST 41
11841  043644  005037  042146                  CLR     @#TEMP2         ;0 SHIFTED BY 0=0 AND PS=4
11842  043650  000207                          RTS     PC
11843  043652  021527  000041          TST41:  CMP     (R5),#41        ;IS IT TEST 41?
11844  043656  001004                          BNE     TST42           ;IF NOT THEN TRY TEST 42
11845  043660  012737  177760  042146          MOV     #-16.,@#TEMP2   ;0 SHIFTED BY -16.=0 AND PS=4
11846  043666  000207                          RTS     PC
11847  043670  021527  000042          TST42:  CMP     (R5),#42        ;IS IT TEST 42?
11848  043674  001013                          BNE     TST43           ;IF NOT THEN TRY TEST 43
11849  043676  012737  100000  042144          MOV     #100000,@#TEMP1 ;100000
11850  043704  005237  042146                  INC     @#TEMP2         ;SHIFTED BY -15
11851  043710  005337  042150                  DEC     @#TEMP3         ;IS=-1
11852  043714  012737  000010  042152          MOV     #10,@#TEMP4             ;AND PS=10
11853  043722  000207                          RTS     PC
11854  043724  021527  000043          TST43:  CMP     (R5),#43        ;IS IT TEST 43?
11855  043730  001012                          BNE     TST44           ;IF NOT THEN IF NOT THEN TRY TEST 44
11856  043732  012737  125252  042144          MOV     #125252,@#TEMP1 ;125252
11857  043740  012737  177777  042146          MOV     #-1,@#TEMP2     ;SHIFTED BY -1
11858  043746  012737  152525  042150          MOV     #152525,@#TEMP3 ;IS=152525 AND PS=10
11859  043754  000207                          RTS     PC
```

```
11860  043756  021527  000044          TST44:  CMP     (R5),#44        ;IS IT TEST 44?
11861  043762  001012                          BNE     TST45           ;IF NOT THEN TRY TEST 45
11862  043764  012737  000001  042146          MOV     #1,@#TEMP2      ;125252 SHIFTED BY 1
11863  043772  012737  052524  042150          MOV     #52524,@#TEMP3  ;IS=52524
11864  044000  012737  000003  042152          MOV     #3,@#TEMP4      ;AND PS=3
11865  044006  000207                          RTS     PC
11866  044010  021527  000045          TST45:  CMP     (R5),#45        ;IS IT TEST 45?
11867  044014  001012                          BNE     TST46           ;IF NOT THEN TRY TEST 46
11868  044016  012737  177776  042146          MOV     #-2,@#TEMP2     ;125252 SHIFTED BY -2
11869  044024  012737  165252  042150          MOV     #165252,@#TEMP3 ;IS=165252
11870  044032  012737  000011  042152          MOV     #11,@#TEMP4     ;AND PS=11
11871  044040  000207                          RTS     PC
11872  044042  021527  000046          TST46:  CMP     (R5),#46        ;IS IT TEST 46?
11873  044046  001014                          BNE     TST47           ;IF NOT THEN TRY TEST 47
11874  044050  012737  177777  042144          MOV     #-1,@#TEMP1     ;-1
11875  044056  012737  000020  042146          MOV     #16.,@#TEMP2    ;SHIFTED BY 15.
11876  044064  005037  042150                  CLR     @#TEMP3         ;IS=0
11877  044070  012737  000007  042152          MOV     #7,@#TEMP4      ;AND PS=7
11878  044076  000207                          RTS     PC
11879  044100  021527  000047          TST47:  CMP     (R5),#47        ;IS IT TEST 47?
11880  044104  001011                          BNE     TST50           ;IF NOT THEN TRY TEST 50
11881  044106  005337  042146                  DEC     @#TEMP2         ;-1 SHIFTED BY 15
11882  044112  012737  100000  042150          MOV     #100000,@#TEMP3 ;IS=100000
11883  044120  012737  000011  042152          MOV     #11,@#TEMP4     ;AND PS=11
11884  044126  000207                          RTS     PC
11885  044130  021527  000050          TST50:  CMP     (R5),#50        ;IS IT TEST 50
11886  044134  001007                          BNE     ENT51           ;IF NOT THEN TRY TEST 51
11887  044136  012737  137777  042144          MOV     #137777,@#TEMP1 ;137777 SHIFTED BY 15. IS=100000
11888  044144  012737  000013  042152          MOV     #13,@#TEMP4     ;AND PS=13
11889  044152  000207                          RTS     PC
11890  044154  021527  000051          ENT51:  CMP     (R5),#51        ;IS IT ENTERING TEST 51?
11891  044160  001403                          BEQ     .+10
11892  044162  004767  014340                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11893                                                                  ;TEST NUMBER GOOFED
11894  044166  000015                          15              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11895                                                          ;BY  (013746  000172  000207)
11896
11897
11898  044170  005726                          TST     (SP)+           ;RESTORE STACK POINTER
11899  044172  012704  177771                  MOV     #-7,%4
11900  044176  012702  042164                  MOV     #S1,%2
11901  044202  012703  042166                  MOV     #S2,%3
```

```
11902                                   ;*****************************************************************************
11903                                   ;TEST:51   11/34   ASH     125252 SHIFTED BY #5 = 52500  PS = 3
11904                                   ;*****************************************************************************
11905
11906  044206  010767  133760          TST51:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11907  044212  012701  125252                  MOV     #125252,%1      ;LOAD R1 WITH 125252
11908  044216  072127  000005                  ASH     #5,%1           ;SHIFT R1 BY #5
11909  044222  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
11910  044226  122737  000003  042142          CMPB    #3,@#PSWORD     ;IS THE PS 3?
11911  044234  001403                          BEQ     .+10
11912  044236  004767  014264                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11913                                                                  ;THE PS IS NOT EQUAL TO 3
11914  044242  000016                          16              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11915                                                          ;BY  (013746  000172  000207)
11916
11917  044244  022701  052500                  CMP     #52500,%1       ;IS THE RESULT 52500?
11918  044250  001403                          BEQ     .+10
11919  044252                          1$:
11920  044252  004767  014250                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11921                                                                  ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
11922  044256  000017                          17              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11923                                                          ;BY  (013746  000172  000207)
11924
11925  044260  021527  000051                  CMP     (R5),#51                ;IS $TESTN = #51
11926  044264  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11927  044266  005215                          INC     (R5)
11928
11929
11930
11931                                   ;*****************************************************************************
11932                                   ;TEST:52   11/34   ASH     125252 SHIFTED BY @S2 = 177525  PS = 10
11933                                   ;*****************************************************************************
11934
11935  044270  010767  133676          TST52:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11936  044274  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
11937  044300  072077  175662                  ASH     @S2,%0          ;SHIFT R0 BY @S2
11938  044304  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
11939  044310  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
11940  044316  001403                          BEQ     .+10
11941  044320  004767  014202                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11942                                                                  ;THE PS IS NOT EQUAL TO 10
11943  044324  000020                          20              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11944                                                          ;BY  (013746  000172  000207)
11945
11946  044326  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
11947  044332  001403                          BEQ     .+10
11948  044334                          1$:
11949  044334  004767  014166                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11950                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
11951  044340  000021                          21              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11952                                                          ;BY  (013746  000172  000207)
11953
11954  044342  021527  000052                  CMP     (R5),#52                ;IS $TESTN = #52
11955  044346  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11956  044350  005215                          INC     (R5)
11957
```

```
11958
11959
```

```
11960                                   ;*****************************************************************************
11961                                   ;TEST:53   11/34   ASH     125252 SHIFTED BY @#S1 = 177525  PS = 10
11962                                   ;*****************************************************************************
11963
11964  044352  010767  133614          TST53:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11965  044356  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
11966  044362  072037  042164                  ASH     @#S1,%0         ;SHIFT R0 BY @#S1
11967  044366  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
11968  044372  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
11969  044400  001403                          BEQ     .+10
11970  044402  004767  014120                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11971                                                                  ;THE PS IS NOT EQUAL TO 10
11972  044406  000022                                  22              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11973                                                                  ;BY  (013746  000172  000207)
11974
11975  044410  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
11976  044414  001403                          BEQ     .+10
11977  044416                          1$:
11978  044416  004767  014104                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11979                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
11980  044422  000023                                  23              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11981                                                                  ;BY  (013746  000172  000207)
11982
11983  044424  021527  000053                  CMP     (R5),#53                ;IS $TESTN = #53
11984  044430  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11985  044432  005215                          INC     (R5)
11986
11987
11988
11989                                   ;*****************************************************************************
11990                                   ;TEST:54   11/34   ASH     125252 SHIFTED BY (2) = 177525  PS = 10
11991                                   ;*****************************************************************************
11992
11993  044434  010767  133532          TST54:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11994  044440  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
11995  044444  072012                          ASH     (2),%0          ;SHIFT R0 BY (2)
11996  044446  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
11997  044452  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
11998  044460  001403                          BEQ     .+10
11999  044462  004767  014040                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12000                                                                  ;THE PS IS NOT EQUAL TO 10
12001  044466  000024                                  24              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12002                                                                  ;BY  (013746  000172  000207)
12003
12004  044470  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12005  044474  001403                          BEQ     .+10
12006  044476                          1$:
12007  044476  004767  014024                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12008                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12009  044502  000025                                  25              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12010                                                                  ;BY  (013746  000172  000207)
12011
12012  044504  021527  000054                  CMP     (R5),#54                ;IS $TESTN = #54
12013  044510  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12014  044512  005215                          INC     (R5)
12015
```

```
12016
12017
```

```
12018                                   ;*****************************************************************************
12019                                   ;TEST:55   11/34   ASH     125252 SHIFTED BY (2)+ = 177525  PS = 10
12020                                   ;*****************************************************************************
12021
12022  044514  010767  133452          TST55:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12023  044520  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12024  044524  072022                          ASH     (2)+,%0         ;SHIFT R0 BY (2)+
12025  044526  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12026  044532  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12027  044540  001403                          BEQ     .+10
12028  044542  004767  013760                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12029                                                                  ;THE PS IS NOT EQUAL TO 10
12030  044546  000026                                  26              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12031                                                                  ;BY  (013746  000172  000207)
12032
12033  044550  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12034  044554  001403                          BEQ     .+10
12035  044556                          1$:
12036  044556  004767  013744                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12037                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12038  044562  000027                                  27              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12039                                                                  ;BY  (013746  000172  000207)
12040
12041  044564  021527  000055                  CMP     (R5),#55                ;IS $TESTN = #55
12042  044570  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12043  044572  005215                          INC     (R5)
12044
12045
12046
12047                                   ;*****************************************************************************
12048                                   ;TEST:56   11/34   ASH     125252 SHIFTED BY -(2) = 177525  PS = 10
12049                                   ;*****************************************************************************
12050
12051  044574  010767  133372          TST56:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12052  044600  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12053  044604  072042                          ASH     -(2),%0         ;SHIFT R0 BY -(2)
12054  044606  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12055  044612  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12056  044620  001403                          BEQ     .+10
12057  044622  004767  013700                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12058                                                                  ;THE PS IS NOT EQUAL TO 10
12059  044626  000030                                  30              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12060                                                                  ;BY  (013746  000172  000207)
12061
12062  044630  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12063  044634  001403                          BEQ     .+10
12064  044636                          1$:
12065  044636  004767  013664                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12066                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12067  044642  000031                                  31              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12068                                                                  ;BY  (013746  000172  000207)
12069
12070  044644  021527  000056                  CMP     (R5),#56                ;IS $TESTN = #56
12071  044650  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12072  044652  005215                          INC     (R5)
12073
```

```
12074
12075
```

```
12076                                   ;*****************************************************************************
12077                                   ;TEST:57   11/34   ASH     125252 SHIFTED BY 2(3) = 177252  PS = 11
12078                                   ;*****************************************************************************
12079
12080  044654  010767  133312          TST57:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12081  044660  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12082  044664  072063  000002                  ASH     2(3),%0         ;SHIFT R0 BY 2(3)
12083  044670  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12084  044674  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12085  044702  001403                          BEQ     .+10
12086  044704  004767  013616                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12087                                                                   ;THE PS IS NOT EQUAL TO 11
12088  044710  000032                          32              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12089                                                          ;BY  (013746  000172  000207)
12090
12091  044712  022700  177252                  CMP     #177252,%0      ;IS THE RESULT 177252?
12092  044716  001403                          BEQ     .+10
12093  044720                          1$:
12094  044720  004767  013602                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12095                                                                   ;R0 IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
12096  044724  000033                          33              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12097                                                          ;BY  (013746  000172  000207)
12098
12099  044726  021527  000057                  CMP     (R5),#57                ;IS $TESTN = #57
12100  044732  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12101  044734  005215                          INC     (R5)
12102
12103
12104
12105                                   ;*****************************************************************************
12106                                   ;TEST:60   11/34   ASH     125252 SHIFTED BY @(3) = 177525  PS = 10
12107                                   ;*****************************************************************************
12108
12109  044736  010767  133230          TST60:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12110  044742  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12111  044746  072073  000000                  ASH     @(3),%0         ;SHIFT R0 BY @(3)
12112  044752  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12113  044756  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12114  044764  001403                          BEQ     .+10
12115  044766  004767  013534                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12116                                                                   ;THE PS IS NOT EQUAL TO 10
12117  044772  000034                          34              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12118                                                          ;BY  (013746  000172  000207)
12119
12120  044774  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12121  045000  001403                          BEQ     .+10
12122  045002                          1$:
12123  045002  004767  013520                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12124                                                                   ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12125  045006  000035                          35              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12126                                                          ;BY  (013746  000172  000207)
12127
12128  045010  021527  000060                  CMP     (R5),#60                ;IS $TESTN = #60
12129  045014  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12130  045016  005215                          INC     (R5)
12131
```

```
 12132
 12133
```

```
12134                                   ;*****************************************************************************
12135                                   ;TEST:61   11/34   ASH     125252 SHIFTED BY @(3)+ = 177525  PS - 10
12136                                   ;*****************************************************************************
12137
12138  045020  010767  133146          TST61:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12139  045024  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12140  045030  072033                          ASH     @(3)+,%0        ;SHIFT R0 BY @(3)+
12141  045032  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12142  045036  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12143  045044  001403                          BEQ     .+10
12144  045046  004767  013454                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12145                                                                  ;THE PS IS NOT EQUAL TO 10
12146  045052  000036                          36              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12147                                                          ;BY  (013746  000172  000207)
12148
12149  045054  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12150  045060  001403                          BEQ     .+10
12151  045062                          1$:
12152  045062  004767  013440                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12153                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12154  045066  000037                          37              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12155                                                          ;BY  (013746  000172  000207)
12156
12157  045070  021527  000061                  CMP     (R5),#61                ;IS $TESTN = #61
12158  045074  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12159  045076  005215                          INC     (R5)
12160
12161
12162
12163                                   ;*****************************************************************************
12164                                   ;TEST:62   11/34   ASH     125252 SHIFTED BY @-(3) = 177525  PS   10
12165                                   ;*****************************************************************************
12166
12167  045100  010767  133066          TST62:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12168  045104  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12169  045110  072053                          ASH     @-(3),%0        ;SHIFT R0 BY @-(3)
12170  045112  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12171  045116  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12172  045124  001403                          BEQ     .+10
12173  045126  004767  013374                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12174                                                                  ;THE PS IS NOT EQUAL TO 10
12175  045132  000040                          40              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12176                                                          ;BY  (013746  000172  000207)
12177
12178  045134  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12179  045140  001403                          BEQ     .+10
12180  045142                          1$:
12181  045142  004767  013360                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12182                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12183  045146  000041                          41              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12184                                                          ;BY  (013746  000172  000207)
12185
12186  045150  021527  000062                  CMP     (R5),#62                ;IS $TESTN = #62
12187  045154  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12188  045156  005215                          INC     (R5)
12189
```

```
12190
12191
```

```
12192                                   ;**********************************
12193                                   ;     ASHC INSTRUCTION TESTS
12194                                   ;**********************************
12195
12196
12197
12198                                           ;************
12199                                           ;TESTS 63-157
12200                                           ;************
12201
12202
12203
12204  045160  010767  133006                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12205  045164  012737  000062  042140          MOV     #62,@#COUNT
12206  045172  005037  042144                  CLR     @#TEMP1         ;TEMP1=0
12207  045176  012737  000001  042146          MOV     #1,@#TEMP2      ;TEMP2=1
12208  045204  005037  042150                  CLR     @#TEMP3         ;TEMP3=0
12209  045210  005037  042152                  CLR     @#TEMP4         ;TEMP4=0
12210  045214  012737  000001  042154          MOV     #1,@#TEMP5      ;TEMP5=1
12211  045222  005037  042156                  CLR     @#TEMP6         ;0 1 SHIFTED BY 0=0 1, PS-0
12212
12213  045226  010502                  REG01:  MOV     R5,R2           ;SAVE R5
12214  045230  013700  042144                  MOV     @#TEMP1,%0      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
12215  045234  013701  042146                  MOV     @#TEMP2,%0.1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
12216  045240  000241                          CLC
12217  045242  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12218  045250  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
12219  045252  013705  042150                  MOV     @#TEMP3,R5      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12220  045256  073005                          ASHC    R5,R0           ;USING R0
12221  045260  000402                          BR      4$
12222  045262  073067  174662          2$:     ASHC    TEMP3,%0        ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
12223  045266  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
12224  045272  123737  042156  042142          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12225  045300  001403                          BEQ     .+10
12226  045302  004767  013220                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12227                                                                  ;WRONG PS
12228  045306  000042                  42              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12229                                                   ;BY  (013746  000172  000207)
12230
12231  045310  005237  042140                  INC     @#COUNT
12232  045314  023700  042152                  CMP     @#TEMP4,%0      ;IS THE RESULT IN R0 SAME AS TEMP4?
12233  045320  001403                          BEQ     .+10
12234  045322  004767  013200                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12235                                                                  ;WRONG RESULT IN R0
12236  045326  000043                  43              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12237                                                   ;BY  (013746  000172  000207)
12238
12239  045330  023701  042154                  CMP     @#TEMP5,%1      ;IS THE RESULT IN R1 SAME AS TEMP5?
12240  045334  001403                          BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12241                                                                  ;AND PS=TEMP6
12242  045336  004767  013164                  ISR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12243                                                                  ;WRONG RESULT IN R1
12244  045342  000044                  44              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12245                                                   ;BY  (013746  000172  000207)
12246
12247  045344  010205                  MOV     R2,R5           ;RESTORE R5
```

```
12248  045346  021537  042140                  CMP     (R5),@#COUNT    ;IS TEST NUMBER=COUNTER?
12249  045352  001403                          BEQ     .+10
12250  045354  004767  013146                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12251                                                                   ;NO
12252  045360  000045                          45              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12253                                                          ;BY  (013746  000172  000207)
12254
12255  045362  005215                          INC     (R5)
12256  045364  021527  000160                  CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12257  045370  002014                          BGE     6$              ;YES
12258  045372  005237  042150                  INC     @#TEMP3
12259  045376  000241                          CLC
12260  045400  006137  042154                  ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12261  045404  006137  042152                  ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP4 IN TEMP5
12262  045410  021527  000121                  CMP     (R5),#121       ;IS IT TEST 121?
12263  045414  001004                          BNE     REGR23
12264  045416  004467  000414                  JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12265  045422  004767  000444          6$:     JSR     %7,TST160
12266  045426  010767  132540          REGR23: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12267  045432  013702  042144                  MOV     @#TEMP1,%2      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
12268  045436  013703  042146                  MOV     @#TEMP2,%2!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
12269  045442  000241                          CLC
12270  045444  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12271  045452  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
12272  045454  013704  042150                  MOV     @#TEMP3,R4      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12273  045460  073204                          ASHC    R4,R2           ;USING R2
12274  045462  000402                          BR      4$
12275  045464  073267  174460          2$:     ASHC    TEMP3,%2        ;ASHC REGISTER 2 BY THE CONTETNS OF TEMP3
12276  045470  106737  042142          4$:     MFPS    @#PSWORD        ;SAVE PS
12277  045474  123737  042156  042142          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12278  045502  001403                          BEQ     .+10
12279  045504  004767  013016                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12280                                                                   ;WRONG PS
12281  045510  000046                          46              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12282                                                          ;BY  (013746  000172  000207)
12283
12284  045512  005237  042140                  INC     @#COUNT
12285  045516  023702  042152                  CMP     @#TEMP4,%2      ;IS THE RESULT IN R2 SAME AS TEMP4?
12286  045522  001403                          BEQ     .+10
12287  045524  004767  012776                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12288                                                                   ;WRONG RESULT IN R2
12289  045530  000047                          47              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12290                                                          ;BY  (013746  000172  000207)
12291
12292  045532  023703  042154                  CMP     @#TEMP5,%3      ;IS THE RESULT IN R3 SAME AS TEMP5?
12293  045536  001403                          BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12294                                                                   ;AND PS=TEMP6
12295  045540  004767  012762                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12296                                                                   ;WRONG RESULT IN R1
12297  045544  000050                          50              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12298                                                          ;BY  (013746  000172  000207)
12299
12300  045546  021537  042140                  CMP     (R5),@#COUNT    ;IS TEST NUMBER=COUNTER?
12301  045552  001403                          BEQ     .+10
12302  045554  004767  012746                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12303                                                                   ;NO
```

```
12304  045560  000051                          51              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12305                                                          ;BY  (013746  000172  000207)
12306
12307  045562  005215                  INC     (R5)
12308  045564  021527  000160          CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12309  045570  002014                  BGE     6$              ;YES
12310  045572  005237  042150          INC     @#TEMP3
12311  045576  000241                  CLC
12312  045600  006137  042154          ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12313  045604  006137  042152          ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
12314  045610  021527  000121          CMP     (R5),#121       ;IS IT TEST 121?
12315  045614  001004                  BNE     REG45
12316  045616  004467  000214          JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12317  045622  004767  000244  6$:     JSR     %7,TST160
12318  045626  010767  132340  REG45:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12319  045632  010501                  MOV     R5,R1           ;SAVE R5
12320  045634  013704  042144          MOV     @#TEMP1,%4      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
12321  045640  013705  042146          MOV     @#TEMP2,%4!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
12322  045644  000241                  CLC
12323  045646  032737  000001  000306  BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12324  045654  001004                  BNE     2$              ;IF NOT THEN GO TO 2$
12325  045656  013700  042150          MOV     @#TEMP3,R0      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12326  045662  073400                  ASHC    R0,R4           ;USING R4
12327  045664  000402                  BR      4$
12328  045666  073467  174256  2$:     ASHC    TEMP3,%4        ;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
12329  045672  106737  042142  4$:     MFPS    @#PSWORD        ;SAVE PS
12330  045676  123737  042156  042142  CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12331  045704  001403                  BEQ     .+10
12332  045706  004767  012614          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12333                                                          ;WRONG PS
12334  045712  000052                  52              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12335                                                  ;BY  (013746  000172  000207)
12336
12337  045714  005237  042140          INC     @#COUNT
12338  045720  023704  042152          CMP     @#TEMP4,%4      ;IS THE RESULT IN R4 SAME AS TEMP4?
12339  045724  001403                  BEQ     .+10
12340  045726  004767  012574          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12341                                                          ;WRONG RESULT IN R4
12342  045732  000053                  53              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12343                                                  ;BY  (013746  000172  000207)
12344
12345  045734  023705  042154          CMP     @#TEMP5,%5      ;IS THE RESULT IN R5 SAME AS TEMP5?
12346  045740  001403                  BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12347                                                          ;AND PS=TEMP6
12348  045742  004767  012560          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12349                                                          ;WRONG RESULT IN R5
12350  045746  000054                  54              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12351                                                  ;BY  (013746  000172  000207)
12352
12353  045750  021137  042140          CMP     (R1),@#COUNT    ;IS TEST NUMBER=COUNTER?
12354  045754  001403                  BEQ     .+10
12355  045756  004767  012544          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12356                                                          ;NO
12357  045762  000055                  55              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12358                                                  ;BY  (013746  000172  000207)
12359
```

```
12360  045764  010105                        MOV     R1,R5           ;RESTORE R5
12361  045766  005215                        INC     (R5)
12362  045770  021527  000160                CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12363  045774  002014                        BGE     6$              ;YES
12364  045776  005237  042150                INC     @#TEMP3
12365  046002  000241                        CLC
12366  046004  006137  042154                ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12367  046010  006137  042152                ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
12368  046014  021527  000121                CMP     (R5),#121       ;IS IT TEST 121?
12369  046020  001004                        BNE     8$
12370  046022  004467  000010                JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12371  046026  004767  000040        6$:     JSR     %7,TST160
12372  046032  000167  177170        8$:     JMP     REG01
12373  046036  022424                RITSH:  CMP     (R4)+,(R4)+     ;MAKE R4 POINT TO THE NEXT REG TAG
12374  046040  012737  040000  042144        MOV     #40000,@#TEMP1  ;TEMP1=4000
12375  046046  005037  042146                CLR     @#TEMP2         ;TEMP2=0
12376  046052  012737  177742  042150        MOV     #-30.,@#TEMP3   ;TEMP3=-30
12377  046060  005037  042152                CLR     @#TEMP4         ;TEMP4=0
12378  046064  005237  042154                INC     @#TEMP5         ;TEMP5=1
12379  046070  000204                        RTS     R4
12380  046072  021527  000160        TST160: CMP     (R5),#160       ;IS IT TEST 160
12381  046076  001010                        BNE     TST161          ;IF NOT THEN TRY TEST 161
12382  046100  005037  042144                CLR     @#TEMP1         ;0 0 SHIFTED BY 0
12383  046104  005037  042152                CLR     @#TEMP4         ;IS EQUAL TO 0 0
12384  046110  012737  000004  042156        MOV     #4,@#TEMP6      ;AND PS=4
12385  046116  000207                        RTS     %7
12386  046120  021527  000161        TST161: CMP     (R5),#161       ;IS IT TEST 161
12387  046124  001004                        BNE     TST162
12388  046126  012737  177746  042150        MOV     #-32,@#TEMP3    ;0 0 SHIFTED BY -32=0 0, PS=4
12389  046134  000207                        RTS     %7
12390  046136  021527  000162        TST162: CMP     (R5),#162       ;IS IT TEST 162
12391  046142  001004                        BNE     TST163          ;IF NOT THEN TRY TEST 163
12392  046144  012737  000032  042150        MOV     #32,@#TEMP3     ;0 0 SHIFTED BY 32=0 0, PS=4
12393  046152  000207                        RTS     %7
12394  046154  021527  000163        TST163: CMP     (R5),#163       ;IS IT TEST 163?
12395  046160  001016                        BNE     TST164          ;IF NOT THEN TRY TEST 164
12396  046162  012737  052525  042144        MOV     #52525,@#TEMP1  ;52525 0
12397  046170  012737  177760  042150        MOV     #-16.,@#TEMP3   ;SHIFTED BY -16.
12398  046176  005037  042152                CLR     @#TEMP4
12399  046202  012737  052525  042154        MOV     #52525,@#TEMP5  ;IS EQUAL TO 0 52525
12400  046210  005037  042156                CLR     @#TEMP6         ;AND PS = 0
12401  046214  000207                        RTS     %7
12402  046216  021527  000164        TST164: CMP     (R5),#164       ;IS IT TEST 164?
12403  046222  001014                        BNE     TST165          ;IF NOT THEN TRY TEST 165
12404  046224  012737  125252  042144        MOV     #125252,@#TEMP1 ;125252 0 SHIFTED BY -16.
12405  046232  005337  042152                DEC     @#TEMP4
12406  046236  012737  125252  042154        MOV     #125252,@#TEMP5 ;IS EQUAL TO -1 125252
12407  046244  012737  000010  042156        MOV     #10,@#TEMP6     ;AND PS=10
12408  046252  000207                        RTS     %7
12409  046254  021527  000165        TST165: CMP     (R5),#165       ;IS IT TEST 165?
12410  046260  001007                        BNE     TST166          ;IF NOT THEN TRY TEST 166
12411  046262  012737  177777  042144        MOV     #-1,@#TEMP1     ;-1 0 SHIFTED BY -16
12412  046270  012737  177777  042154        MOV     #-1,@#TEMP5     ;IS EQUAL TO -1 -1, AND PS=10
12413  046276  000207                        RTS     %7
12414  046300  021527  000166        TST166: CMP     (R5),#166       ;IS IT TEST 166?
12415  046304  001011                        BNE     TST167          ;IF NOT THEN TRY TEST 167
```

```
12416  046306  012737  100000  042144          MOV     #100000,@#TEMP1 ;100000 0
12417  046314  012737  177740  042150          MOV     #-32.,@#TEMP3   ;SHIFTED BY -32 IS EQUAL TO -1 -1
12418  046322  005237  042156                  INC     @#TEMP6         ;AND PS=11
12419  046326  000207                          RTS     %7
12420  046330  021527  000167          TST167: CMP     (R5),#167       ;IS IT TEST 167?
12421  046334  001014                          BNE     TST170          ;IF NOT THEN TRY TEST 170
12422  046336  005037  042144                  CLR     @#TEMP1
12423  046342  005337  042146                  DEC     @#TEMP2         ;0 -1
12424  046346  012737  000020  042150          MOV     #16.,@#TEMP3    ;SHIFTED BY 16.
12425  046354  005037  042154                  CLR     @#TEMP5         ;IS EQUAL TO -1 0
12426  046360  005237  042156                  INC     @#TEMP6         ;AND PS=12
12427  046364  000207                          RTS     %7
12428  046366  021527  000170          TST170: CMP     (R5),#170       ;IS IT TEST 170?
12429  046372  001007                          BNE     TST171          ;IF NOT THEN TRY TEST 171
12430  046374  012737  125252  042146          MOV     #125252,@#TEMP2 ;0 125252 SHIFTED BY 16
12431  046402  012737  125252  042152          MOV     #125252,@#TEMP4 ;IS EQUAL TO 125252 0, AND PS=12
12432  046410  000207                          RTS     %7
12433  046412  021527  000171          TST171: CMP     (R5),#171       ;IS IT TEST 171?
12434  046416  001010                          BNE     TST172          ;IF NOT THEN TRY TEST 172
12435  046420  005337  042150                  DEC     @#TEMP3         ;0 125252 SHIFTED BY 15
12436  046424  012737  052525  042152          MOV     #52525,@#TEMP4  ;IS EQUAL TO 52525 0
12437  046432  005037  042156                  CLR     @#TEMP6         ;AND PS=0
12438  046436  000207                          RTS     %7
12439  046440  021527  000172          TST172: CMP     (R5),#172       ;IS IT TEST 172?
12440  046444  001006                          BNE     TST173          ;IF NOT THEN TRY TEST 173
12441  046446  012737  052525  042146          MOV     #52525,@#TEMP2  ;0 52525
12442  046454  005237  042150                  INC     @#TEMP3         ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
12443  046460  000207                          RTS     %7
12444  046462  021527  000173          TST173: CMP     (R5),#173       ;IS IT TEST 173?
12445  046466  001014                          BNE     TST174          ;IF NOT THEN TRY TEST 174
12446  046470  012737  177777  042146          MOV     #-1,@#TEMP2     ;0 -1
12447  046476  005337  042150                  DEC     @#TEMP3         ;SHIFTED BY 15.
12448  046502  012737  077777  042152          MOV     #77777,@#TEMP4
12449  046510  012737  100000  042154          MOV     #100000,@#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
12450  046516  000207                          RTS     %7
12451  046520  021527  000174          TST174: CMP     (R5),#174       ;IS IT TEST 174?
12452  046524  001013                          BNE     TST175          ;IF NOT THEN TRY TEST 175
12453  046526  012737  100000  042144          MOV     #100000,@#TEMP1
12454  046534  005337  042146                  DEC     @#TEMP2         ;100000 -2 SHIFTED BY 15.
12455  046540  005037  042154                  CLR     @#TEMP5         ;IS EQUAL TO 77777 0
12456  046544  012737  000002  042156          MOV     #2,@#TEMP6      ;AND PS=2
12457  046552  000207                          RTS     %7
12458  046554  021527  000175          TST175: CMP     (R5),#175       ;IS IT TEST 175?
12459  046560  001015                          BNE     ENT176          ;IF NOT THEN TRY TEST 176
12460  046562  012737  177777  042144          MOV     #-1,@#TEMP1
12461  046570  005037  042146                  CLR     @#TEMP2         ;-1 0
12462  046574  005237  042150                  INC     @#TEMP3         ;SHIFTED BY 16.
12463  046600  005037  042152                  CLR     @#TEMP4         ;IS EQUAL TO 0 0
12464  046604  012737  000007  042156          MOV     #7,@#TEMP6      ;AND PS=7
12465  046612  000207                          RTS     %7
12466  046614  021527  000176          ENT176: CMP     (R5),#176       ;IS THE PROGRM ENTERING TEST 176?
12467  046620  001403                          BEQ     .+10
12468  046622  004767  011700                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12469                                                                  ;TEST NUMBER GOOFED
12470  046626  000056                          56              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12471                                                          ;BY  (013746  000172  000207)
```

```
12472
12473
12474  046630  005726                    TST     (SP)+           ;RESTORE STACK POINTER
12475
```

```
12476                                   ;*****************************************************************************
12477                                   ;TEST:176       1 SHIFTED BY 8. = 400  PS = 0
12478                                   ;*****************************************************************************
12479
12480  046632  010767  131334          TST176: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12481  046636  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12482  046642  012701  000001                  MOV     #1,%1!1         ;LOAD R1!1 WITH 1
12483  046646  000241                          CLC
12484  046650  073127  000010                  ASHC    #8.,%1          ;SHIFT R1,R1!1 BY 8.
12485  046654  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12486  046660  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12487  046666  001403                          BEQ     .+10
12488  046670  004767  011632                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12489                                                                  ;THE PS IS NOT EQUAL TO 0
12490  046674  000057                          57              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12491                                                          ;BY  (013746  000172  000207)
12492
12493  046676  022701  000400                  CMP     #400,%1         ;IS THE RESULT 400?
12494  046702  001403                          BEQ     .+10
12495  046704  004767  011616                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12496                                                                  ;R1 IS NOT EQUAL TO 400
12497  046710  000060                          60              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12498                                                          ;BY  (013746  000172  000207)
12499
12500  046712  021527  000176                  CMP     (R5),#176               ;IS $TESTN = #176?
12501  046716  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12502  046720  004767  011602                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12503                                                                  ;TEST IS IN WRONG SEQUENCE
12504  046724  000061                          61              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12505                                                          ;BY  (013746  000172  000207)
12506
12507  046726  005215                          INC     (R5)
12508
12509
12510                                   ;*****************************************************************************
12511                                   ;TEST:177       -1 SHIFTED BY 15. = 100000  PS = 11
12512                                   ;*****************************************************************************
12513
12514  046730  010767  131236          TST177: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12515  046734  012703  000000                  MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
12516  046740  012703  177777                  MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
12517  046744  000241                          CLC
12518  046746  073327  000017                  ASHC    #15.,%3         ;SHIFT R3,R3!1 BY 15.
12519  046752  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12520  046756  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12521  046764  001403                          BEQ     .+10
12522  046766  004767  011534                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12523                                                                  ;THE PS IS NOT EQUAL TO 11
12524  046772  000062                          62              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12525                                                          ;BY  (013746  000172  000207)
12526
12527  046774  022703  100000                  CMP     #100000,%3      ;IS THE RESULT 100000?
12528  047000  001403                          BEQ     .+10
12529  047002  004767  011520                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12530                                                                  ;R3 IS NOT EQUAL TO 100000
12531  047006  000063                          63              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
```

```
                                                        ;BY  (013746  000172  000207)
                                                        ;BY  (013746  000172  000207)
012534  047010  021527  000177          CMP     (R5),#177               ;IS $TESTN = #177?
012535  047014  001403                  BEQ     .+10            ;IF NOT THEN GO TO HLT
012536  047016  004767  011504          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                        ;TEST IS IN WRONG SEQUENCE
012538  047022  000064                  64              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)
012541  047024  005215                  INC     (R5)
```

```
12544                                   ;*****************************************************************************
12545                                   ;TEST:200       52525 SHIFTED BY 0 = 52525  PS = 0
12546                                   ;*****************************************************************************
12547
12548  047026  010767  131140          TST200: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12549  047032  010501                          MOV     R5,R1           ;SAVE R5
12550  047034  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
12551  047040  012705  052525                  MOV     #52525,%5!1     ;LOAD R5!1 WITH 52525
12552  047044  000241                          CLC
12553  047046  073527  000000                  ASHC    #0,%5           ;SHIFT R5,R5!1 BY 0
12554  047052  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12555  047056  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12556  047064  001403                          BEQ     .+10
12557  047066  004767  011434                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12558                                                                  ;THE PS IS NOT EQUAL TO 0
12559  047072  000065                          65              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12560                                                          ;BY  (013746  000172  000207)
12561
12562  047074  022705  052525                  CMP     #52525,%5       ;IS THE RESULT 52525?
12563  047100  001403                          BEQ     .+10
12564  047102  004767  011420                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12565                                                                  ;R5 IS NOT EQUAL TO 52525
12566  047106  000066                          66              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12567                                                          ;BY  (013746  000172  000207)
12568
12569  047110  010105                          MOV     R1,R5           ;RESTORE R5
12570  047112  021527  000200                  CMP     (R5),#200               ;IS $TESTN = #200?
12571  047116  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12572  047120  004767  011402                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12573                                                                  ;TEST IS IN WRONG SEQUENCE
12574  047124  000067                          67              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12575                                                          ;BY  (013746  000172  000207)
12576
12577  047126  005215                  INC     (R5)
12578
12579
12580                                   ;*****************************************************************************
12581                                   ;TEST:201       20010 SHIFTED BY -13. = 101  PS = 0
12582                                   ;*****************************************************************************
12583
12584  047130  010767  131036          TST201: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12585  047134  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12586  047140  012701  020010                  MOV     #20010,%1!1     ;LOAD R1!1 WITH 20010
12587  047144  000241                          CLC
12588  047146  073127  177763                  ASHC    #-13.,%1        ;SHIFT R1,R1!1 BY -13.
12589  047152  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12590  047156  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12591  047164  001403                          BEQ     .+10
12592  047166  004767  011334                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12593                                                                  ;THE PS IS NOT EQUAL TO 0
12594  047172  000070                          70              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12595                                                          ;BY  (013746  000172  000207)
12596
12597  047174  022701  000101                  CMP     #101,%1         ;IS THE RESULT 101?
12598  047200  001403                          BEQ     .+10
12599  047202  004767  011320                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```
12600
12601  047206  000071                          71              ;R1 IS NOT EQUAL TO 101
12602                                                          ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12603                                                          ;BY  (013746  000172  000207)
12604  047210  021527  000201                  CMP     (R5),#201               ;IS $TESTN = #201?
12605  047214  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12606  047216  004767  011304                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12607                                                          ;TEST IS IN WRONG SEQUENCE
12608  047222  000072                          72              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12609                                                          ;BY  (013746  000172  000207)
12610
12611  047224  005215                          INC     (R5)
12612
12613
```

```
12614                                   ;*****************************************************************************
12615                                   ;TEST:202       -1 SHIFTED BY 16. = 0  PS = 11
12616                                   ;*****************************************************************************
12617
12618  047226  010767  130740          TST202: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12619  047232  012703  000000                  MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
12620  047236  012703  177777                  MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
12621  047242  000241                          CLC
12622  047244  073327  000020                  ASHC    #16.,%3         ;SHIFT R3,R3!1 BY 16.
12623  047250  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12624  047254  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12625  047262  001403                          BEQ     .+10
12626  047264  004767  011236                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12627                                                                  ;THE PS IS NOT EQUAL TO 11
12628  047270  000073                          73              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12629                                                          ;BY  (013746  000172  000207)
12630
12631  047272  022703  000000                  CMP     #0,%3           ;IS THE RESULT 0?
12632  047276  001403                          BEQ     .+10
12633  047300  004767  011222                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12634                                                                  ;R3 IS NOT EQUAL TO 0
12635  047304  000074                          74              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12636                                                          ;BY  (013746  000172  000207)
12637
12638  047306  021527  000202                  CMP     (R5),#202               ;IS $TESTN = #202?
12639  047312  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12640  047314  004767  011206                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12641                                                                  ;TEST IS IN WRONG SEQUENCE
12642  047320  000075                          75              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12643                                                          ;BY  (013746  000172  000207)
12644
12645  047322  005215                          INC     (R5)
12646
12647
12648                                   ;*****************************************************************************
12649                                   ;TEST:203       1 SHIFTED BY -1 = 100000  PS = 1
12650                                   ;*****************************************************************************
12651
12652  047324  010767  130642          TST203: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12653  047330  010501                          MOV     R5,R1           ;SAVE R5
12654  047332  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
12655  047336  012705  000001                  MOV     #1,%5!1         ;LOAD R5!1 WITH 1
12656  047342  000241                          CLC
12657  047344  073527  177777                  ASHC    #-1,%5          ;SHIFT R5,R5!1 BY -1
12658  047350  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12659  047354  122737  000001  042142          CMPB    #1,@#PSWORD     ;IS THE PS 1?
12660  047362  001403                          BEQ     .+10
12661  047364  004767  011136                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12662                                                                  ;THE PS IS NOT EQUAL TO 1
12663  047370  000076                          76              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12664                                                          ;BY  (013746  000172  000207)
12665
12666  047372  022705  100000                  CMP     #100000,%5      ;IS THE RESULT 100000?
12667  047376  001403                          BEQ     .+10
12668  047400  004767  011122                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12669                                                                  ;R5 IS NOT EQUAL TO 100000
```

```
12670  047404  000077                          77              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12671                                                           ;BY  (013746  000172  000207)
12672
12673  047406  010105                          MOV     R1,R5           ;RESTORE R5
12674  047410  021527  000203                  CMP     (R5),#203               ;IS $TESTN = #203?
12675  047414  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12676  047416  004767  011104                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12677                                                           ;TEST IS IN WRONG SEQUENCE
12678  047422  000100                          100             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12679                                                           ;BY  (013746  000172  000207)
12680
12681  047424  005215                          INC     (R5)
12682
12683
```

```
12684                                   ;*****************************************************************************
12685                                   ;TEST:204       125252 SHIFTED BY -16. = 125252  PS = 11
12686                                   ;*****************************************************************************
12687
12688  047426  010767  130540          TST204: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12689  047432  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12690  047436  012701  125252                  MOV     #125252,%1!1    ;LOAD R1!1 WITH 125252
12691  047442  000241                          CLC
12692  047444  073127  177760                  ASHC    #-16.,%1        ;SHIFT R1,R1!1 BY -16.
12693  047450  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12694  047454  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12695  047462  001403                          BEQ     .+10
12696  047464  004767  011036                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12697                                                                  ;THE PS IS NOT EQUAL TO 11
12698  047470  000101                          101             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12699                                                          ;BY  (013746  000172  000207)
12700
12701  047472  022701  125252                  CMP     #125252,%1      ;IS THE RESULT 125252?
12702  047476  001403                          BEQ     .+10
12703  047500  004767  011022                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12704                                                                  ;R1 IS NOT EQUAL TO 125252
12705  047504  000102                          102             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12706                                                          ;BY  (013746  000172  000207)
12707
12708  047506  021527  000204                  CMP     (R5),#204               ;IS $TESTN = #204?
12709  047512  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12710  047514  004767  011006                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12711                                                                  ;TEST IS IN WRONG SEQUENCE
12712  047520  000103                          103             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12713                                                          ;BY  (013746  000172  000207)
12714
12715  047522  005215                          INC     (R5)
12716
12717
12718                                   ;*****************************************************************************
12719                                   ;TEST:205       125252 125252 SHIFTED BY 21. = 52500 000000  PS = 3
12720                                   ;*****************************************************************************
12721
12722  047524  010767  130442          TST205: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12723  047530  012702  125252                  MOV     #125252,%2      ;LOAD R2 WITH 125252
12724  047534  012703  125252                  MOV     #125252,%2!1    ;LOAD R2!1 WITH 125252
12725  047540  000241                          CLC
12726  047542  073227  000025                  ASHC    #21.,%2         ;SHIFT R2,R2!1 BY 21.
12727  047546  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12728  047552  122737  000003  042142          CMPB    #3,@#PSWORD     ;IS THE PS 3?
12729  047560  001403                          BEQ     .+10
12730  047562  004767  010740                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12731                                                                  ;THE PS IS NOT EQUAL TO 3
12732  047566  000104                          104             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12733                                                          ;BY  (013746  000172  000207)
12734
12735  047570  022702  052500                  CMP     #52500,%2       ;IS THE RESULT 52500?
12736  047574  001403                          BEQ     .+10
12737  047576  004767  010724                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12738                                                                  ;R2 IS NOT EQUAL TO 52500
12739  047602  000105                          105             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
```

```
12740                                                   ;BY  (013746  000172  000207)
12741
12742  047604  022703  000000          CMP     #000000,%2!1    ;IS THE RESULT 000000?
12743  047610  001403                  BEQ     .+10
12744  047612  004767  010710          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12745                                                   ;R2!1 IS NOT EQUAL TO 000000
12746  047616  000106                  106             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12747                                                   ;BY  (013746  000172  000207)
12748
12749  047620  021527  000205          CMP     (R5),#205               ;IS $TESTN = #205?
12750  047624  001403                  BEQ     .+10            ;IF NOT THEN GO TO HLT
12751  047626  004767  010674          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12752                                                   ;TEST IS IN WRONG SEQUENCE
12753  047632  000107                  107             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12754                                                   ;BY  (013746  000172  000207)
12755
12756  047634  005215                  INC     (R5)
12757
12758
12759
12760  047636  012702  177771          MOV     #-7,%2
12761  047642  012703  042164          MOV     #S1,%3
12762  047646  012704  042166          MOV     #S2,%4
12763
```

```
12764                                   ;************************ ************************************************
12765                                   ;TEST:206       125252 125252 SHIFTED BY S1 = 177525 52525  PS = 10
12766                                   ;*****************************************************************************
12767
12768  047652  010767  130314          TST206: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12769  047656  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12770  047662  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12771  047666  000241                          CLC
12772  047670  073067  172270                  ASHC    S1,%0           ;SHIFT R0,R0!1 BY S1
12773  047674  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12774  047700  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12775  047706  001403                          BEQ     .+10
12776  047710  004767  010612                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12777                                                                  ;THE PS IS NOT EQUAL TO 10
12778  047714  000110                          110             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12779                                                          ;BY  (013746  000172  000207)
12780
12781  047716  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12782  047722  001403                          BEQ     .+10
12783  047724  004767  010576                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12784                                                                  ;R0 IS NOT EQUAL TO 177525
12785  047730  000111                          111             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12786                                                          ;BY  (013746  000172  000207)
12787
12788  047732  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12789  047736  001403                          BEQ     .+10
12790  047740                          1$:
12791  047740  004767  010562                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12792                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12793  047744  000112                          112             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12794                                                          ;BY  (013746  000172  000207)
12795
12796  047746  021527  000206                  CMP     (R5),#206               ;IS THE $TESTN = #206?
12797  047752  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12798  047754  005215                          INC  .  (R5)
12799
12800
12801                                   ;*****************************************************************************
12802                                   ;TEST:207       125252 125252 SHIFTED BY @S2 = 177525 52525  PS = 10
12803                                   ;*****************************************************************************
12804
12805  047756  010767  130210          TST207: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12806  047762  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12807  047766  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12808  047772  000241                          CLC
12809  047774  073077  172166                  ASHC    @S2,%0          ;SHIFT R0,R0!1 BY @S2
12810  050000  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12811  050004  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12812  050012  001403                          BEQ     .+10
12813  050014  004767  010506                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12814                                                                  ;THE PS IS NOT EQUAL TO 10
12815  050020  000113                          113             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12816                                                          ;BY  (013746  000172  000207)
12817
12818  050022  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12819  050026  001403                          BEQ     .+10
```

```
12820  050030  004767  010472                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12821                                                           ;R0 IS NOT EQUAL TO 177525
12822  050034  000114                          114             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12823                                                           ;BY  (013746  000172  000207)
12824
12825  050036  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12826  050042  001403                          BEQ     .+10
12827  050044                          1$:
12828  050044  004767  010456                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12829                                                           ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12830  050050  000115                          115             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12831                                                           ;BY  (013746  000172  000207)
12832
12833  050052  021527  000207                  CMP     (R5),#207               ;IS THE $TESTN = #207?
12834  050056  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12835  050060  005215                          INC     (R5)
12836
12837
```

```
12838                                   ;*****************************************************************************
12839                                   ;TEST:210       125252 125252 SHIFTED BY @#S1 = 177525 52525  PS = 10
12840                                   ;*****************************************************************************
12841
12842  050062  010767  130104          TST210: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12843  050066  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12844  050072  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12845  050076  000241                          CLC
12846  050100  073037  042164                  ASHC    @#S1,%0         ;SHIFT R0,R0!1 BY @#S1
12847  050104  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12848  050110  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12849  050116  001403                          BEQ     .+10
12850  050120  004767  010402                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12851                                                                  ;THE PS IS NOT EQUAL TO 10
12852  050124  000116                          116             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12853                                                          ;BY  (013746  000172  000207)
12854
12855  050126  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12856  050132  001403                          BEQ     .+10
12857  050134  004767  010366                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12858                                                                  ;R0 IS NOT EQUAL TO 177525
12859  050140  000117                          117             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12860                                                          ;BY  (013746  000172  000207)
12861
12862  050142  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12863  050146  001403                          BEQ     .+10
12864  050150                          1$:
12865  050150  004767  010352                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12866                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12867  050154  000120                          120             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12868                                                          ;BY  (013746  000172  000207)
12869
12870  050156  021527  000210                  CMP     (R5),#210               ;IS THE $TESTN = #210?
12871  050162  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12872  050164  005215                          INC     (R5)
12873
12874
12875                                   ;*****************************************************************************
12876                                   ;TEST:211       125252 125252 SHIFTED BY (3) = 177525 52525  PS = 10
12877                                   ;*****************************************************************************
12878
12879  050166  010767  130000          TST211: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12880  050172  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12881  050176  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12882  050202  000241                          CLC
12883  050204  073013                          ASHC    (3),%0          ;SHIFT R0,R0!1 BY (3)
12884  050206  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12885  050212  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12886  050220  001403                          BEQ     .+10
12887  050222  004767  010300                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12888                                                                  ;THE PS IS NOT EQUAL TO 10
12889  050226  000121                          121             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12890                                                          ;BY  (013746  000172  000207)
12891
12892  050230  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12893  050234  001403                          BEQ     .+10
```

```
12894  050236  004767  010264                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12895                                                           ;R0 IS NOT EQUAL TO 177525
12896  050242  000122                          122             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12897                                                           ;BY  (013746  000172  000207)
12898
12899  050244  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12900  050250  001403                          BEQ     .+10
12901  050252                          1$:
12902  050252  004767  010250                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12903                                                           ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12904  050256  000123                          123             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 '001403)
12905                                                           ;BY  (013746  000172  000207)
12906
12907  050260  021527  000211                  CMP     (R5),#211               ;IS THE $TESTN = #211?
12908  050264  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12909  050266  005215                          INC     (R5)
12910
12911
```

```
12912                                   ;*****************************************************************************
12913                                   ;TEST:212       125252 125252 SHIFTED BY (3)+ = 177525 52525  PS = 10
12914                                   ;*****************************************************************************
12915
12916  050270  010767  127676          TST212: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12917  050274  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12918  050300  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12919  050304  000241                          CLC
12920  050306  073023                          ASHC    (3)+,%0         ;SHIFT R0,R0!1 BY (3)+
12921  050310  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12922  050314  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12923  050322  001403                          BEQ     .+10
12924  050324  004767  010176                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12925                                                                  ;THE PS IS NOT EQUAL TO 10
12926  050330  000124                                  124             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12927                                                                  ;BY  (013746  000172  000207)
12928
12929  050332  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12930  050336  001403                          BEQ     .+10
12931  050340  004767  010162                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12932                                                                  ;R0 IS NOT EQUAL TO 177525
12933  050344  000125                                  125             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12934                                                                  ;BY  (013746  000172  000207)
12935
12936  050346  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12937  050352  001403                          BEQ     .+10
12938  050354                          1$:
12939  050354  004767  010146                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12940                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12941  050360  000126                                  126             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12942                                                                  ;BY  (013746  000172  000207)
12943
12944  050362  021527  000212                  CMP     (R5),#212               ;IS THE $TESTN = #212?
12945  050366  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12946  050370  005215                          INC     (R5)
12947
12948
12949                                   ;*****************************************************************************
12950                                   ;TEST:213       125252 125252 SHIFTED BY -(3) = 177525 52525  PS = 10
12951                                   ;*****************************************************************************
12952
12953  050372  010767  127574          TST213: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12954  050376  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12955  050402  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12956  050406  000241                          CLC
12957  050410  073043                          ASHC    -(3),%0         ;SHIFT R0,R0!1 BY -(3)
12958  050412  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12959  050416  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12960  050424  001403                          BEQ     .+10
12961  050426  004767  010074                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12962                                                                  ;THE PS IS NOT EQUAL TO 10
12963  050432  000127                                  127             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12964                                                                  ;BY  (013746  000172  000207)
12965
12966  050434  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12967  050440  001403                          BEQ     .+10
```

```
12968  050442  004767  010060                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12969                                                           ;R0 IS NOT EQUAL TO 177525
12970  050446  000130                          130             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12971                                                           ;BY  (013746  000172  000207)
12972
12973  050450  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12974  050454  001403                          BEQ     .+10
12975  050456                          1$:
12976  050456  004767  010044                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12977                                                           ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12978  050462  000131                          131             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12979                                                           ;BY  (013746  000172  000207)
12980
12981  050464  021527  000213                  CMP     (R5),#213               ;IS THE $TESTN = #213?
12982  050470  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12983  050472  005215                          INC     (R5)
12984
12985
```

```
12986                                   ;*************************************************************************
12987                                   ;TEST:214       125252 125252 SHIFTED BY 2(4) = 177252 125252  PS = 11
12988                                   ;*************************************************************************
12989
12990  050474  010767  127472          TST214: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12991  050500  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12992  050504  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12993  050510  000241                          CLC
12994  050512  073064  000002                  ASHC    2(4),%0         ;SHIFT R0,R0!1 BY 2(4)
12995  050516  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
12996  050522  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12997  050530  001403                          BEQ     .+10
12998  050532  004767  007770                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12999                                                                  ;THE PS IS NOT EQUAL TO 11
13000  050536  000132                          132             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13001                                                           ;BY  (013746  000172  000207)
13002
13003  050540  022700  177252                  CMP     #177252,%0      ;IS THE RESULT 177252?
13004  050544  001403                          BEQ     .+10
13005  050546  004767  007754                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13006                                                                  ;R0 IS NOT EQUAL TO 177252
13007  050552  000133                          133             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13008                                                           ;BY  (013746  000172  000207)
13009
13010  050554  022701  125252                  CMP     #125252,%0!1    ;IS THE RESULT 125252?
13011  050560  001403                          BEQ     .+10
13012  050562                          1$:
13013  050562  004767  007740                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13014                                                                  ;R0!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
13015  050566  000134                          134             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13016                                                           ;BY  (013746  000172  000207)
13017
13018  050570  021527  000214                  CMP     (R5),#214               ;IS THE $TESTN = #214?
13019  050574  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13020  050576  005215                          INC     (R5)
13021
13022
13023                                   ;*************************************************************************
13024                                   ;TEST:215       125252 125252 SHIFTED BY @(4) = 177525 52525  PS = 10
13025                                   ;*************************************************************************
13026
13027  050600  010767  127366          TST215: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13028  050604  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13029  050610  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13030  050614  000241                          CLC
13031  050616  073074  000000                  ASHC    @(4),%0         ;SHIFT R0,R0!1 BY @(4)
13032  050622  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13033  050626  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13034  050634  001403                          BEQ     .+10
13035  050636  004767  007664                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13036                                                                  ;THE PS IS NOT EQUAL TO 10
13037  050642  000135                          135             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13038                                                           ;BY  (013746  000172  000207)
13039
13040  050644  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13041  050650  001403                          BEQ     .+10
```

```
13042  050652  004767  007650                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13043                                                           ;R0 IS NOT EQUAL TO 177525
13044  050656  000136                          136             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13045                                                           ;BY  (013746  000172  000207)
13046
13047  050660  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
13048  050664  001403                          BEQ     .+10
13049  050666                          1$:
13050  050666  004767  007634                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13051                                                           ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13052  050672  000137                          137             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13053                                                           ;BY  (013746  000172  000207)
13054
13055  050674  021527  000215                  CMP     (R5),#215               ;IS THE $TESTN = #215?
13056  050700  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13057  050702  005215                          INC     (R5)
13058
13059
```

```
13060                                   ;*************************************************************************
13061                                   ;TEST:216       125252 125252 SHIFTED BY @(4)+ = 177525 52525  PS = 10
13062                                   ;*************************************************************************
13063
13064  050704  010767  127262          TST216: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13065  050710  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13066  050714  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13067  050720  000241                          CLC
13068  050722  073034                          ASHC    @(4)+,%0                ;SHIFT R0,R0!1 BY @(4)+
13069  050724  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13070  050730  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13071  050736  001403                          BEQ     .+10
13072  050740  004767  007562                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13073                                                                  ;THE PS IS NOT EQUAL TO 10
13074  050744  000140                          140             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13075                                                          ;BY  (013746  000172  000207)
13076
13077  050746  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13078  050752  001403                          BEQ     .+10
13079  050754  004767  007546                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13080                                                                  ;R0 IS NOT EQUAL TO 177525
13081  050760  000141                          141             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13082                                                          ;BY  (013746  000172  000207)
13083
13084  050762  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
13085  050766  001403                          BEQ     .+10
13086  050770                          1$:
13087  050770  004767  007532                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13088                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13089  050774  000142                          142             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13090                                                          ;BY  (013746  000172  000207)
13091
13092  050776  021527  000216                  CMP     (R5),#216               ;IS THE $TESTN = #216?
13093  051002  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13094  051004  005215                          INC     (R5)
13095
13096
13097                                   ;*************************************************************************
13098                                   ;TEST:217       125252 125252 SHIFTED BY @-(4) = 177525 52525  PS = 10
13099                                   ;*************************************************************************
13100
13101  051006  010767  127160          TST217: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13102  051012  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13103  051016  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13104  051022  000241                          CLC
13105  051024  073054                          ASHC    @-(4),%0                ;SHIFT R0,R0!1 BY @-(4)
13106  051026  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13107  051032  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13108  051040  001403                          BEQ     .+10
13109  051042  004767  007460                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13110                                                                  ;THE PS IS NOT EQUAL TO 10
13111  051046  000143                          143             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13112                                                          ;BY  (013746  000172  000207)
13113
13114  051050  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13115  051054  001403                          BEQ     .+10
```

```
13116  051056  004767  007444                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13117                                                           ;R0 IS NOT EQUAL TO 177525
13118  051062  000144                          144             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13119                                                           ;BY  (013746  000172  000207)
13120
13121  051064  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
13122  051070  001403                          BEQ     .+10
13123  051072                          1$:
13124  051072  004767  007430                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13125                                                           ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13126  051076  000145                          145             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13127                                                           ;BY  (013746  000172  000207)
13128
13129  051100  021527  000217                  CMP     (R5),#217               ;IS THE $TESTN = #217?
13130  051104  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13131  051106  005215                          INC     (R5)
13132
13133
13134
13135
13136
13137
13138
13139
13140
```

```
13141                                   ;*********************************
13142                                   ;     MUL INSTRUCTION TESTS
13143                                   ;*********************************
13144
13145
13146
13147
13148                           ;*************************************************************************
13149                           ;TEST:220       MUL       1 * #0 = 0 0         PS = 4
13150                           ;*************************************************************************
13151
13152  051110  010767  127056          TST220: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13153  051114  012700  000001                  MOV     #1,%0           ;LOAD MULTIPLICAND WITH 1
13154  051120  070027  000000                  MUL     #0,%0           ;MULTIPLY 1 * #0
13155  051124  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13156  051130  122737  000004  042142          CMPB    #4,@#PSWORD     ;IS PS = 4
13157  051136  001403                          BEQ     .+10
13158  051140  004767  007362                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13159                                                                  ;PS IS WRONG
13160  051144  000146                          146             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13161                                                          ;BY  (013746  000172  000207)
13162
13163  051146  022700  000000                  CMP     #0,%0           ;IS HIGH ORDER = 0
13164  051152  001403                          BEQ     .+10
13165  051154  004767  007346                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13166                                                                  ;HIGH ORDER IS WRONG
13167  051160  000147                          147             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13168                                                          ;BY  (013746  000172  000207)
13169
13170  051162  022701  000000                  CMP     #0,%0!1         ;IS LOW ORDER = 0
13171  051166  001403                          BEQ     .+10
13172  051170                          1$:
13173  051170  004767  007332                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13174                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13175  051174  000150                          150             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13176                                                          ;BY  (013746  000172  000207)
13177
13178  051176  021527  000220                  CMP     (R5),#220
13179  051202  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13180  051204  005215                          INC     (R5)
13181
13182
```

```
13183                                   ;*************************************************************************
13184                                   ;TEST:221       MUL     -1 * #1 = -1 -1         PS = 10
13185                                   ;*************************************************************************
13186
13187  051206  010767  126760          TST221: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13188  051212  012700  177777                  MOV     #-1,%0          ;LOAD MULTIPLICAND WITH -1
13189  051216  070027  000001                  MUL     #1,%0           ;MULTIPLY -1 * #1
13190  051222  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13191  051226  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
13192  051234  001403                          BEQ     .+10
13193  051236  004767  007264                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13194                                                                  ;PS IS WRONG
13195  051242  000151                          151             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13196                                                          ;BY  (013746  000172  000207)
13197
13198  051244  022700  177777                  CMP     #-1,%0          ;IS HIGH ORDER = -1
13199  051250  001403                          BEQ     .+10
13200  051252  004767  007250                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13201                                                                  ;HIGH ORDER IS WRONG
13202  051256  000152                          152             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13203                                                          ;BY  (013746  000172  000207)
13204
13205  051260  022701  177777                  CMP     #-1,%0!1        ;IS LOW ORDER = -1
13206  051264  001403                          BEQ     .+10
13207  051266                          1$:
13208  051266  004767  007234                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13209                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13210  051272  000153                          153             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13211                                                          ;BY  (013746  000172  000207)
13212
13213  051274  021527  000221                  CMP     (R5),#221
13214  051300  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13215  051302  005215                          INC     (R5)
13216
13217
```

```
13218                                   ;*************************************************************************
13219                                   ;TEST:222       MUL     2 * #2 = 0 4        PS = 0
13220                                   ;*************************************************************************
13221
13222  051304  010767  126662          TST222: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13223  051310  012702  000002                  MOV     #2,%2           ;LOAD MULTIPLICAND WITH 2
13224  051314  070227  000002                  MUL     #2,%2           ;MULTIPLY 2 * #2
13225  051320  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13226  051324  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
13227  051332  001403                          BEQ     .+10
13228  051334  004767  007166                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13229                                                                  ;PS IS WRONG
13230  051340  000154                          154             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13231                                                          ;BY  (013746  000172  000207)
13232
13233  051342  022702  000000                  CMP     #0,%2           ;IS HIGH ORDER = 0
13234  051346  001403                          BEQ     .+10
13235  051350  004767  007152                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13236                                                                  ;HIGH ORDER IS WRONG
13237  051354  000155                          155             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13238                                                          ;BY  (013746  000172  000207)
13239
13240  051356  022703  000004                  CMP     #4,%2!1         ;IS LOW ORDER = 4
13241  051362  001403                          BEQ     .+10
13242  051364                          1$:
13243  051364  004767  007136                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13244                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13245  051370  000156                          156             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13246                                                          ;BY  (013746  000172  000207)
13247
13248  051372  021527  000222                  CMP     (R5),#222
13249  051376  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13250  051400  005215                          INC     (R5)
13251
13252
```

```
13253                                   ;*************************************************************************
13254                                   ;TEST:223       MUL     1000 * #200 = 1 0       PS = 1
13255                                   ;*************************************************************************
13256
13257  051402  010767  126564         TST223: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13258  051406  010501                         MOV     R5,R1           ;SAVE R5
13259  051410  012704  001000                 MOV     #1000,%4        ;LOAD MULTIPLICAND WITH 1000
13260  051414  070427  000200                 MUL     #200,%4         ;MULTIPLY 1000 * #200
13261  051420  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
13262  051424  122737  000001  042142         CMPB    #1,@#PSWORD     ;IS PS = 1
13263  051432  001403                         BEQ     .+10
13264  051434  004767  007066                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13265                                                                  ;PS IS WRONG
13266  051440  000157                         157             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13267                                                          ;BY  (013746  000172  000207)
13268
13269  051442  022704  000001                 CMP     #1,%4           ;IS HIGH ORDER = 1
13270  051446  001403                         BEQ     .+10
13271  051450  004767  007052                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13272                                                                  ;HIGH ORDER IS WRONG
13273  051454  000160                         160             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13274                                                          ;BY  (013746  000172  000207)
13275
13276  051456  022705  000000                 CMP     #0,%4!1         ;IS LOW ORDER = 0
13277  051462  001403                         BEQ     .+10
13278  051464                          1$:
13279  051464  004767  007036                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13280                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13281  051470  000161                         161             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13282                                                          ;BY  (013746  000172  000207)
13283
13284  051472  021127  000223                 CMP     (R1),#223               ;CHECK THE TEST NUMBER
13285  051476  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13286  051500  010105                         MOV     R1,R5           ;RESTORE R5
13287  051502  005215                         INC     (R5)
13288
13289
```

```
13290                                   ;*****************************************************************************
13291                                   ;TEST:224       MUL     2 * #77777 = 0 177776       PS = 1
13292                                   ;*****************************************************************************
13293
13294  051504  010767  126462         TST224: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13295  051510  012700  000002                 MOV     #2,%0           ;LOAD MULTIPLICAND WITH 2
13296  051514  070027  077777                 MUL     #77777,%0       ;MULTIPLY 2 * #77777
13297  051520  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
13298  051524  122737  000001  042142         CMPB    #1,@#PSWORD     ;IS PS = 1
13299  051532  001403                         BEQ     .+10
13300  051534  004767  006766                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13301                                                                  ;PS IS WRONG
13302  051540  000162                         162             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13303                                                          ;BY  (013746  000172  000207)
13304
13305  051542  022700  000000                 CMP     #0,%0           ;IS HIGH ORDER = 0
13306  051546  001403                         BEQ     .+10
13307  051550  004767  006752                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13308                                                                  ;HIGH ORDER IS WRONG
13309  051554  000163                         163             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13310                                                          ;BY  (013746  000172  000207)
13311
13312  051556  022701  177776                 CMP     #177776,%0!1    ;IS LOW ORDER = 177776
13313  051562  001403                         BEQ     .+10
13314  051564                          1$:
13315  051564  004767  006736                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13316                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13317  051570  000164                         164             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13318                                                          ;BY  (013746  000172  000207)
13319
13320  051572  021527  000224                 CMP     (R5),#224
13321  051576  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13322  051600  005215                         INC     (R5)
13323
13324
```

```
13325                                   ;*****************************************************************************
13326                                   ;TEST:225       MUL     7777 * #10 = 0 77770        PS = 0
13327                                   ;*****************************************************************************
13328
13329  051602  010767  126364           TST225: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13330  051606  012702  007777                   MOV     #7777,%2        ;LOAD MULTIPLICAND WITH 7777
13331  051612  070227  000010                   MUL     #10,%2          ;MULTIPLY 7777 * #10
13332  051616  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
13333  051622  122737  000000  042142           CMPB    #0,@#PSWORD     ;IS PS = 0
13334  051630  001403                           BEQ     .+10
13335  051632  004767  006670                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13336                                                                   ;PS IS WRONG
13337  051636  000165                           165             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13338                                                           ;BY  (013746  000172  000207)
13339
13340  051640  022702  000000                   CMP     #0,%2           ;IS HIGH ORDER = 0
13341  051644  001403                           BEQ     .+10
13342  051646  004767  006654                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13343                                                                   ;HIGH ORDER IS WRONG
13344  051652  000166                           166             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13345                                                           ;BY  (013746  000172  000207)
13346
13347  051654  022703  077770                   CMP     #77770,%2!1     ;IS LOW ORDER = 77770
13348  051660  001403                           BEQ     .+10
13349  051662                           1$:
13350  051662  004767  006640                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13351                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE .
13352  051666  000167                           167             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13353                                                           ;BY  (013746  000172  000207)
13354
13355  051670  021527  000225                   CMP     (R5),#225
13356  051674  001372                           BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13357  051676  005215                           INC     (R5)
13358
13359
```

```
13360                                   ;*****************************************************************************
13361                                   ;TEST:226       MUL     77777 * #77777 = 37777 1        PS = 1
13362                                   ;*****************************************************************************
13363
13364  051700  010767  126266          1ST226: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13365  051704  010501                          MOV     R5,R1           ;SAVE R5
13366  051706  012704  077777                  MOV     #77777,%4       ;LOAD MULTIPLICAND WITH 77777
13367  051712  070427  077777                  MUL     #77777,%4       ;MULTIPLY 77777 * #77777
13368  051716  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13369  051722  122737  000001  042142          CMPB    #1,@#PSWORD     ;IS PS = 1
13370  051730  001403                          BEQ     .+10
13371  051732  004767  006570                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13372                                                                  ;PS IS WRONG
13373  051736  000170                          170             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13374                                                          ;BY  (013746  000172  000207)
13375
13376  051740  022704  037777                  CMP     #37777,%4       ;IS HIGH ORDER = 37777
13377  051744  001403                          BEQ     .+10
13378  051746  004767  006554                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13379                                                                  ;HIGH ORDER IS WRONG
13380  051752  000171                          171             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13381                                                          ;BY  (013746  000172  000207)
13382
13383  051754  022705  000001                  CMP     #1,%4!1         ;IS LOW ORDER = 1
13384  051760  001403                          BEQ     .+10
13385  051762                          1$:
13386  051762  004767  006540                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13387                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13388  051766  000172                          172             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13389                                                          ;BY  (013746  000172  000207)
13390
13391  051770  021127  000226                  CMP     (R1),#226       ;CHECK THE TEST NUMBER
13392  051774  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13393  051776  010105                          MOV     R1,R5           ;RESTORE R5
13394  052000  005215                          INC     (R5)
13395
13396
```

```
13397                                   ;*****************************************************************************
13398                                   ;TEST:227       MUL     -1 * #77777 = -1 100001         PS = 10
13399                                   ;*****************************************************************************
13400
13401  052002  010767  126164          TST227: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13402  052006  012702  177777                  MOV     #-1,%2          ;LOAD MULTIPLICAND WITH -1
13403  052012  070227  077777                  MUL     #77777,%2       ;MULTIPLY -1 * #77777
13404  052016  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13405  052022  122737  000010  042142        CMPB    #10,@#PSWORD    ;IS PS = 10
13406  052030  001403                        BEQ     .+10
13407  052032  004767  006470                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13408                                                                  ;PS IS WRONG
13409  052036  000173                        173             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13410                                                         ;BY  (013746  000172  000207)
13411
13412  052040  022702  177777                CMP     #-1,%2          ;IS HIGH ORDER = -1
13413  052044  001403                        BEQ     .+10
13414  052046  004767  006454                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13415                                                                  ;HIGH ORDER IS WRONG
13416  052052  000174                        174             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13417                                                         ;BY  (013746  000172  000207)
13418
13419  052054  022703  100001                CMP     #100001,%2.1    ;IS LOW ORDER = 100001
13420  052060  001403                        BEQ     .+10
13421  052062                          1$:
13422  052062  004767  006440                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13423                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13424  052066  000175                        175             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13425                                                         ;BY  (013746  000172  000207)
13426
13427  052070  021527  000227                CMP     (R5),#227
13428  052074  001372                        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13429  052076  005215                        INC     (R5)
13430
13431
```

```
13432                                   ;*****************************************************************************
13433                                   ;TEST:230       MUL     -2 * #77777 = -1 2      PS - 11
13434                                   ;*****************************************************************************
13435
13436  052100  010767  126066          TST230: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13437  052104  012700  177776                  MOV     #-2,%0          ;LOAD MULTIPLICAND WITH -2
13438  052110  070027  077777                  MUL     #77777,%0       ;MULTIPLY -2 * #77777
13439  052114  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13440  052120  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13441  052126  001403                          BEQ     .+10
13442  052130  004767  006372                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13443                                                                  ;PS IS WRONG
13444  052134  000176                          176             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13445                                                          ;BY  (013746  000172  000207)
13446
13447  052136  022700  177777                  CMP     #-1,%0          ;IS HIGH ORDER = -1
13448  052142  001403                          BEQ     .+10
13449  052144  004767  006356                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13450                                                                  ;HIGH ORDER IS WRONG
13451  052150  000177                          177             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13452                                                          ;BY  (013746  000172  000207)
13453
13454  052152  022701  000002                  CMP     #2,%0.1         ;IS LOW ORDER = 2
13455  052156  001403                          BEQ     .+10
13456  052160                          1$:
13457  052160  004767  006342                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13458                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13459  052164  000200                          200             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13460                                                          ;BY  (013746  000172  000207)
13461
13462  052166  021527  000230                  CMP     (R5),#23C
13463  052172  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13464  052174  005215                          INC     (R5)
13465
13466
```

```
13467                                   ;*************************************************************************
13468                                   ;TEST:231       MUL     125252 * #2 = -1 52524          PS = 11
13469                                   ;*************************************************************************
13470
13471  052176  010767  125770         TST231: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13472  052202  012702  125252                 MOV     #125252,%2      ;LOAD MULTIPLICAND WITH 125252
13473  052206  070227  000002                 MUL     #2,%2           ;MULTIPLY 125252 * #2
13474  052212  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
13475  052216  122737  000011  042142         CMPB    #11,@#PSWORD    ;IS PS = 11
13476  052224  001403                         BEQ     .+10
13477  052226  004767  006274                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13478                                                                  ;PS IS WRONG
13479  052232  000201                         201             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13480                                                          ;BY  (013746  000172  000207)
13481
13482  052234  022702  177777                 CMP     #-1,%2          ;IS HIGH ORDER = -1
13483  052240  001403                         BEQ     .+10
13484  052242  004767  006260                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13485                                                                  ;HIGH ORDER IS WRONG
13486  052246  000202                         202             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13487                                                          ;BY  (013746  000172  000207)
13488
13489  052250  022703  052524                 CMP     #52524,%2!1     ;IS LOW ORDER = 52524
13490  052254  001403                         BEQ     .+10
13491  052256                          1$:
13492  052256  004767  006244                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13493                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13494  052262  000203                         203             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13495                                                          ;BY  (013746  000172  000207)
13496
13497  052264  021527  000231                 CMP     (R5),#231
13498  052270  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13499  052272  005215                         INC     (R5)
13500
13501
```

```
13502                                   ;*************************************************************************
13503                                   ;TEST:232       MUL     125252 * #40000 = 165252 100000         PS = 11
13504                                   ;*************************************************************************
13505
13506  052274  010767  125672          TST232: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13507  052300  010501                          MOV     R5,R1           ;SAVE R5
13508  052302  012704  125252                  MOV     #125252,%4      ;LOAD MULTIPLICAND WITH 125252
13509  052306  070427  040000                  MUL     #40000,%4       ;MULTIPLY 125252 * #40000
13510  052312  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13511  052316  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13512  052324  001403                          BEQ     .+10
13513  052326  004767  006174                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13514                                                                  ;PS IS WRONG
13515  052332  000204                          204             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13516                                                          ;BY  (013746  000172  000207)
13517
13518  052334  022704  165252                  CMP     #165252,%4      ;IS HIGH ORDER = 165252
13519  052340  001403                          BEQ     .+10
13520  052342  004767  006160                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13521                                                                  ;HIGH ORDER IS WRONG
13522  052346  000205                          205             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13523                                                          ;BY  (013746  000172  000207)
13524
13525  052350  022705  100000                  CMP     #100000,%4.1    ;IS LOW ORDER = 100000
13526  052354  001403                          BEQ     .+10
13527  052356                          1$:
13528  052356  004767  006144                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13529                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13530  052362  000206                          206             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13531                                                          ;BY  (013746  000172  000207)
13532
13533  052364  021127  000232                  CMP     (R1),#232               ;CHECK THE TEST NUMBER
13534  052370  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13535  052372  010105                          MOV     R1,R5           ;RESTORE R5
13536  052374  005215                          INC     (R5)
13537
13538
```

```
13539                                   ;*****************************************************************************
13540                                   ;TEST:233       MUL      107070 * #107070 = 31222 26100        PS = 1
13541                                   ;*****************************************************************************
13542
13543  052376  010767  125570          TST233: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13544  052402  012700  107070                  MOV     #107070,%0      ;LOAD MULTIPLICAND WITH 107070
13545  052406  070027  107070                  MUL     #107070,%0      ;MULTIPLY 107070 * #107070
13546  052412  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13547  052416  122737  000001  042142          CMPB    #1,@#PSWORD     ;IS PS = 1
13548  052424  001403                          BEQ     .+10
13549  052426  004767  006074                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13550                                                                  ;PS IS WRONG
13551  052432  000207                                  207             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13552                                                                  ;BY  (013746  000172  000207)
13553
13554  052434  022700  031222                  CMP     #31222,%0       ;IS HIGH ORDER = 31222
13555  052440  001403                          BEQ     .+10
13556  052442  004767  006060                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13557                                                                  ;HIGH ORDER IS WRONG
13558  052446  000210                                  210             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13559                                                                  ;BY  (013746  000172  000207)
13560
13561  052450  022701  026100                  CMP     #26100,%0!1     ;IS LOW ORDER = 26100
13562  052454  001403                          BEQ     .+10
13563  052456                          1$:
13564  052456  004767  006044                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13565                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13566  052462  000211                                  211             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13567                                                                  ;BY  (013746  000172  000207)
13568
13569  052464  021527  000233                  CMP     (R5),#233
13570  052470  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13571  052472  005215                          INC     (R5)
13572
13573
```

```
13574                                   ;*****************************************************************************
13575                                   ;TEST:234       MUL      -1 * #1 = -1 -1        PS = 10
13576                                   ;*****************************************************************************
13577
13578  052474  010767  125472          TST234: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13579  052500  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
13580  052504  070127  000001                  MUL     #1,%1           ;MULTIPLY -1 * #1
13581  052510  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13582  052514  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
13583  052522  001403                          BEQ     .+10
13584  052524  004767  005776                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13585                                                                  ;PS IS WRONG
13586  052530  000212                          212             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13587                                                          ;BY  (013746  000172  000207)
13588
13589  052532  022701  177777                  CMP     #-1,%1          ;IS HIGH ORDER = -1
13590  052536  001403                          BEQ     .+10
13591  052540  004767  005762                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13592                                                                  ;HIGH ORDER IS WRONG
13593  052544  000213                          213             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13594                                                          ;BY  (013746  000172  000207)
13595
13596  052546  022701  177777                  CMP     #-1,%1!1        ;IS LOW ORDER = -1
13597  052552  001403                          BEQ     .+10
13598  052554                          1$:
13599  052554  004767  005746                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13600                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13601  052560  000214                          214             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13602                                                          ;BY  (013746  000172  000207)
13603
13604  052562  021527  000234                  CMP     (R5),#234
13605  052566  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13606  052570  005215                          INC     (R5)
13607
13608
```

```
13609                                   ;*****************************************************************************
13610                                   ;TEST:235       MUL     -1 * #0 = 0 0       PS = 4
13611                                   ;*****************************************************************************
13612
13613  052572  010767  125374          TST235: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13614  052576  012703  177777                  MOV     #-1,%3          ;LOAD MULTIPLICAND WITH -1
13615  052602  070327  000000                  MUL     #0,%3           ;MULTIPLY -1 * #0
13616  052606  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13617  052612  122737  000004  042142          CMPB    #4,@#PSWORD     ;IS PS = 4
13618  052620  001403                          BEQ     .+10
13619  052622  004767  005700                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13620                                                                  ;PS IS WRONG
13621  052626  000215                                  215             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13622                                                                  ;BY  (013746  000172  000207)
13623
13624  052630  022703  000000                  CMP     #0,%3           ;IS HIGH ORDER = 0
13625  052634  001403                          BEQ     .+10
13626  052636  004767  005664                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13627                                                                  ;HIGH ORDER IS WRONG
13628  052642  000216                                  216             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13629                                                                  ;BY  (013746  000172  000207)
13630
13631  052644  022703  000000                  CMP     #0,%3!1         ;IS LOW ORDER = 0
13632  052650  001403                          BEQ     .+10
13633  052652                          1$:
13634  052652  004767  005650                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13635                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13636  052656  000217                                  217             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13637                                                                  ;BY  (013746  000172  000207)
13638
13639  052660  021527  000235                  CMP     (R5),#235
13640  052664  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13641  052666  005215                          INC     (R5)
13642
13643
```

```
13644                                   ;*****************************************************************************
13645                                   ;TEST:236       MUL     77777 * #100000 = 100000 100000        PS = 11
13646                                   ;*****************************************************************************
13647
13648  052670  010767  125276          TST236: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13649  052674  010501                          MOV     R5,R1           ;SAVE R5
13650  052676  012705  077777                  MOV     #77777,%5       ;LOAD MULTIPLICAND WITH 77777
13651  052702  070527  100000                  MUL     #100000,%5      ;MULTIPLY 77777 * #100000
13652  052706  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13653  052712  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13654  052720  001403                          BEQ     .+10
13655  052722  004767  005600                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13656                                                                  ;PS IS WRONG
13657  052726  000220                          220             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13658                                                          ;BY  (013746  000172  000207)
13659
13660  052730  022705  100000                  CMP     #100000,%5      ;IS HIGH ORDER = 100000
13661  052734  001403                          BEQ     .+10
13662  052736  004767  005564                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13663                                                                  ;HIGH ORDER IS WRONG
13664  052742  000221                          221             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13665                                                          ;BY  (013746  000172  000207)
13666
13667  052744  022705  100000                  CMP     #100000,%5!1    ;IS LOW ORDER = 100000
13668  052750  001403                          BEQ     .+10
13669  052752                          1$:
13670  052752  004767  005550                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13671                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13672  052756  000222                          222             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13673                                                          ;BY  (013746  000172  000207)
13674
13675  052760  021127  000236                  CMP     (R1),#236               ;CHECK THE TEST NUMBER
13676  052764  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13677  052766  010105                          MOV     R1,R5           ;RESTORE R5
13678  052770  005215                          INC     (R5)
13679
13680
```

```
13681                                   ;*****************************************************************************
13682                                   ;TEST:237       MUL     -1 * #77777 = 100001 100001        PS = 10
13683                                   ;*****************************************************************************
13684
13685  052772  010767  125174          TST237: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13686  052776  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
13687  053002  070127  077777                  MUL     #77777,%1       ;MULTIPLY -1 * #77777
13688  053006  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13689  053012  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
13690  053020  001403                          BEQ     .+10
13691  053022  004767  005500                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13692                                                                  ;PS IS WRONG
13693  053026  000223                                  223             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13694                                                                  ;BY  (013746  000172  000207)
13695
13696  053030  022701  100001                  CMP     #100001,%1      ;IS HIGH ORDER = 100001
13697  053034  001403                          BEQ     .+10
13698  053036  004767  005464                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13699                                                                  ;HIGH ORDER IS WRONG
13700  053042  000224                                  224             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13701                                                                  ;BY  (013746  000172  000207)
13702
13703  053044  022701  100001                  CMP     #100001,%1!1    ;IS LOW ORDER = 100001
13704  053050  001403                          BEQ     .+10
13705  053052                          1$:
13706  053052  004767  005450                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13707                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13708  053056  000225                                  225             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13709                                                                  ;BY  (013746  000172  000207)
13710
13711  053060  021527  000237                  CMP     (R5),#237
13712  053064  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13713  053066  005215                          INC     (R5)
13714
13715
```

```
13716                                   ;*************************************************************************
13717                                   ;TEST:240       MUL     77777 * #77777 = 1 1        PS = 1
13718                                   ;*************************************************************************
13719
13720  053070  010767  125076          TST240: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13721  053074  012703  077777                  MOV     #77777,%3       ;LOAD MULTIPLICAND WITH 77777
13722  053100  070327  077777                  MUL     #77777,%3       ;MULTIPLY 77777 * #77777
13723  053104  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13724  053110  122737  000001  042142          CMPB    #1,@#PSWORD     ;IS PS = 1
13725  053116  001403                          BEQ     .+10
13726  053120  004767  005402                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13727                                                                  ;PS IS WRONG
13728  053124  000226                          226             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13729                                                          ;BY  (013746  000172  000207)
13730
13731  053126  022703  000001                  CMP     #1,%3           ;IS HIGH ORDER = 1
13732  053132  001403                          BEQ     .+10
13733  053134  004767  005366                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13734                                                                  ;HIGH ORDER IS WRONG
13735  053140  000227                          227             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13736                                                          ;BY  (013746  000172  000207)
13737
13738  053142  022703  000001                  CMP     #1,%3!1         ;IS LOW ORDER = 1
13739  053146  001403                          BEQ     .+10
13740  053150                          1$:
13741  053150  004767  005352                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13742                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13743  053154  000230                          230             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13744                                                          ;BY  (013746  000172  000207)
13745
13746  053156  021527  000240                  CMP     (R5),#240
13747  053162  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13748  053164  005215                          INC     (R5)
13749
13750
```

```
13751                                   ;*************************************************************************
13752                                   ;TEST:241       MUL     2 * #2 = 4 4        PS = 0
13753                                   ;*************************************************************************
13754
13755  053166  010767  125000          TST241: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13756  053172  010501                          MOV     R5,R1           ;SAVE R5
13757  053174  012705  000002                  MOV     #2,%5           ;LOAD MULTIPLICAND WITH 2
13758  053200  070527  000002                  MUL     #2,%5           ;MULTIPLY 2 * #2
13759  053204  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13760  053210  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
13761  053216  001403                          BEQ     .+10
13762  053220  004767  005302                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13763                                                                  ;PS IS WRONG
13764  053224  000231                          231             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13765                                                          ;BY  (013746  000172  000207)
13766
13767  053226  022705  000004                  CMP     #4,%5           ;IS HIGH ORDER = 4
13768  053232  001403                          BEQ     .+10
13769  053234  004767  005266                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13770                                                                  ;HIGH ORDER IS WRONG
13771  053240  000232                          232             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13772                                                          ;BY  (013746  000172  000207)
13773
13774  053242  022705  000004                  CMP     #4,%5!1         ;IS LOW ORDER = 4
13775  053246  001403                          BEQ     .+10
13776  053250                          1$:
13777  053250  004767  005252                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13778                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13779  053254  000233                          233             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13780                                                          ;BY  (013746  000172  000207)
13781
13782  053256  021127  000241          CMP     (R1),#241               ;CHECK THE TEST NUMBER
13783  053262  001372                  BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13784  053264  010105                  MOV     R1,R5           ;RESTORE R5
13785  053266  005215                  INC     (R5)
13786
13787
```

```
13788  053270  012702  040000                  MOV     #40000,%2
13789  053274  012703  042174                  MOV     #S5,%3
13790  053300  012704  042176                  MOV     #S6,%4
13791
13792                                  ;*****************************************************************
13793                                  ;TEST:242       MUL      125252 * S5 = 165252 100000        PS = 11
13794                                  ;*****************************************************************
13795
13796  053304  010767  124662          TST242: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13797  053310  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13798  053314  070067  166654                  MUL     S5,%0           ;MULTIPLY 125252 * S5
13799  053320  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13800  053324  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13801  053332  001403                          BEQ     .+10
13802  053334  004767  005166                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13803                                                                  ;PS IS WRONG
13804  053340  000234                          234             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13805                                                          ;BY  (013746  000172  000207)
13806
13807  053342  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13808  053346  001403                          BEQ     .+10
13809  053350  004767  005152                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13810                                                                  ;HIGH ORDER IS WRONG
13811  053354  000235                          235             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13812                                                          ;BY  (013746  000172  000207)
13813
13814  053356  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13815  053362  001403                          BEQ     .+10
13816  053364                          1$:
13817  053364  004767  005136                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13818                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13819  053370  000236                          236             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13820                                                          ;BY  (013746  000172  000207)
13821
13822  053372  021527  000242                  CMP     (R5),#242
13823  053376  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13824  053400  005215                          INC     (R5)
13825
13826
```

```
13827                                   ;*************************************************************************
13828                                   ;TEST:243       MUL     125252 * @S6 = 165252 100000        PS = 11
13829                                   ;*************************************************************************
13830
13831  053402  010767  124564          TST243: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13832  053406  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13833  053412  070077  166560                  MUL     @S6,%0          ;MULTIPLY 125252 * @S6
13834  053416  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13835  053422  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13836  053430  001403                          BEQ     .+10
13837  053432  004767  005070                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13838                                                                  ;PS IS WRONG
13839  053436  000237                                  237             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13840                                                                  ;BY  (013746  000172  000207)
13841
13842  053440  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13843  053444  001403                          BEQ     .+10
13844  053446  004767  005054                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13845                                                                  ;HIGH ORDER IS WRONG
13846  053452  000240                                  240             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13847                                                                  ;BY  (013746  000172  000207)
13848
13849  053454  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13850  053460  001403                          BEQ     .+10
13851  053462                          1$:
13852  053462  004767  005040                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13853                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13854  053466  000241                                  241             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13855                                                                  ;BY  (013746  000172  000207)
13856
13857  053470  021527  000243                  CMP     (R5),#243
13858  053474  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13859  053476  005215                          INC     (R5)
13860
13861
```

```
13862                                   ;*****************************************************************************
13863                                   ;TEST:244       MUL      125252 * @#S5 = 165252 100000        PS = 11
13864                                   ;*****************************************************************************
13865
13866  053500  010767  124466          TST244: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13867  053504  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13868  053510  070037  042174                  MUL     @#S5,%0         ;MULTIPLY 125252 * @#S5
13869  053514  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
13870  053520  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
13871  053526  001403                          BEQ     .+10
13872  053530  004767  004772                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13873                                                                  ;PS IS WRONG
13874  053534  000242                          242             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13875                                                          ;BY  (013746  000172  000207)
13876
13877  053536  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13878  053542  001403                          BEQ     .+10
13879  053544  004767  004756                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13880                                                                  ;HIGH ORDER IS WRONG
13881  053550  000243                          243             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13882                                                          ;BY  (013746  000172  000207)
13883
13884  053552  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13885  053556  001403                          BEQ     .+10
13886  053560                          1$:
13887  053560  004767  004742                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13888                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13889  053564  000244                          244             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13890                                                          ;BY  (013746  000172  000207)
13891
13892  053566  021527  000244                  CMP     (R5),#244
13893  053572  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVF
13894  053574  005215                          INC     (R5)
13895
13896
```

```
13897                                   ;*************************************************************************
13898                                   ;TEST:245       MUL     125252 * %2 = 165252 100000     PS = 11
13899                                   ;*************************************************************************
13900
13901  053576  010767  124370           TST245: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13902  053602  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13903  053606  070002                           MUL     %2,%0           ;MULTIPLY 125252 * %2
13904  053610  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
13905  053614  122737  000011  042142           CMPB    #11,@#PSWORD    ;IS PS = 11
13906  053622  001403                           BEQ     .+10
13907  053624  004767  004676                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13908                                                                   ;PS IS WRONG
13909  053630  000245                           245             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13910                                                           ;BY  (013746  000172  000207)
13911
13912  053632  022700  165252                   CMP     #165252,%0      ;IS HIGH ORDER = 165252
13913  053636  001403                           BEQ     .+10
13914  053640  004767  004662                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13915                                                                   ;HIGH ORDER IS WRONG
13916  053644  000246                           246             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13917                                                           ;BY  (013746  000172  000207)
13918
13919  053646  022701  100000                   CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13920  053652  001403                           BEQ     .+10
13921  053654                           1$:
13922  053654  004767  004646                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13923                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13924  053660  000247                           247             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13925                                                           ;BY  (013746  000172  000207)
13926
13927  053662  021527  000245                   CMP     (R5),#245
13928  053666  001372                           BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13929  053670  005215                           INC     (R5)
13930
13931
```

```
13932                                   ;*************************************************************************
13933                                   ;TEST:246       MUL     125252 * (3)+ = 165252 100000        PS = 11
13934                                   ;*************************************************************************
13935
13936  053672  010767  124274         TST246: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13937  053676  012700  125252                 MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13938  053702  070023                         MUL     (3)+,%0         ;MULTIPLY 125252 * (3)+
13939  053704  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
13940  053710  122737  000011  042142         CMPB    #11,@#PSWORD    ;IS PS = 11
13941  053716  001403                         BEQ     .+10
13942  053720  004767  004602                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13943                                                                  ;PS IS WRONG
13944  053724  000250                         250             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13945                                                        ;BY  (013746  000172  000207)
13946
13947  053726  022700  165252                 CMP     #165252,%0      ;IS HIGH ORDER = 165252
13948  053732  001403                         BEQ     .+10
13949  053734  004767  004566                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13950                                                                  ;HIGH ORDER IS WRONG
13951  053740  000251                         251             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13952                                                        ;BY  (013746  000172  000207)
13953
13954  053742  022701  100000                 CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13955  053746  001403                         BEQ     .+10
13956  053750                          1$:
13957  053750  004767  004552                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13958                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13959  053754  000252                         252             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13960                                                        ;BY  (013746  000172  000207)
13961
13962  053756  021527  000246                 CMP     (R5),#246
13963  053762  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13964  053764  005215                         INC     (R5)
13965
13966
```

```
13967                                   ;*****************************************************************************
13968                                   ;TEST:247       MUL      125252 * -(3) = 165252 100000        PS = 11
13969                                   ;*****************************************************************************
13970
13971  053766  010767  124200           TST247: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13972  053772  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13973  053776  070043                           MUL     -(3),%0         ;MULTIPLY 125252 * -(3)
13974  054000  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
13975  054004  122737  000011  042142           CMPB    #11,@#PSWORD    ;IS PS = 11
13976  054012  001403                           BEQ     .+10
13977  054014  004767  004506                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13978                                                                   ;PS IS WRONG
13979  054020  000253                           253             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13980                                           ;BY  (013746  000172  000207)
13981
13982  054022  022700  165252                   CMP     #165252,%0      ;IS HIGH ORDER = 165252
13983  054026  001403                           BEQ     .+10
13984  054030  004767  004472                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13985                                                                   ;HIGH ORDER IS WRONG
13986  054034  000254                           254             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13987                                           ;BY  (013746  000172  000207)
13988
13989  054036  022701  100000                   CMP     #100000,%0.1    ;IS LOW ORDER = 100000
13990  054042  001403                           BEQ     .+10
13991  054044                          1$:
13992  054044  004767  004456                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13993                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13994  054050  000255                           255             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13995                                           ;BY  (013746  000172  000207)
13996
13997  054052  021527  000247                   CMP     (R5),#247
13998  054056  001372                           BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13999  054060  005215                           INC     (R5)
14000
14001
```

```
14002                                  ;*************************************************************************
14003                                  ;TEST:250       MUL      125252 * 2(4) = 165252 100000        PS = 11
14004                                  ;*************************************************************************
14005
14006  054062  010767  124104         TST250: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14007  054066  012700  125252                 MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14008  054072  070064  000002                 MUL     2(4),%0         ;MULTIPLY 125252 * 2(4)
14009  054076  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
14010  054102  122737  000011  042142         CMPB    #11,@#PSWORD    ;IS PS = 11
14011  054110  001403                         BEQ     .+10
14012  054112  004767  004410                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14013                                                                  ;PS IS WRONG
14014  054116  000256                         256             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14015                                                          ;BY  (013746  000172  000207)
14016
14017  054120  022700  165252                 CMP     #165252,%0      ;IS HIGH ORDER = 165252
14018  054124  001403                         BEQ     .+10
14019  054126  004767  004374                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14020                                                                  ;HIGH ORDER IS WRONG
14021  054132  000257                         257             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14022                                                          ;BY  (013746  000172  000207)
14023
14024  054134  022701  100000                 CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14025  054140  001403                         BEQ     .+10
14026  054142                         1$:
14027  054142  004767  004360                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14028                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14029  054146  000260                         260             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14030                                                          ;BY  (013746  000172  000207)
14031
14032  054150  021527  000250                 CMP     (R5),#250
14033  054154  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14034  054156  005215                         INC     (R5)
14035
14036
```

```
14037                                   ;*****************************************************************************
14038                                   ;TEST:251       MUL     125252 * @(4) = 165252 100000        PS = 1
14039                                   ;*****************************************************************************
14040
14041  054160  010767  124006          TST251: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14042  054164  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14043  054170  070074  000000                  MUL     @(4),%0         ;MULTIPLY 125252 * @(4)
14044  054174  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14045  054200  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
14046  054206  001403                          BEQ     .+10
14047  054210  004767  004312                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14048                                                                  ;PS IS WRONG
14049  054214  000261                          261             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14050                                                          ;BY  (013746  000172  000207)
14051
14052  054216  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
14053  054222  001403                          BEQ     .+10
14054  054224  004767  004276                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14055                                                                  ;HIGH ORDER IS WRONG
14056  054230  000262                          262             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14057                                                          ;BY  (013746  000172  000207)
14058
14059  054232  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14060  054236  001403                          BEQ     .+10
14061  054240                          1$:
14062  054240  004767  004262                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14063                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14064  054244  000263                          263             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14065                                                          ;BY  (013746  000172  000207)
14066
14067  054246  021527  000251                  CMP     (R5),#251
14068  054252  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14069  054254  005215                          INC     (R5)
14070
14071
```

```
14072                                   ;*****************************************************************************
14073                                   ;TEST:252       MUL     125252 * @(4)+ = 165252 100000          PS - 11
14074                                   ;*****************************************************************************
14075
14076  054256  010767  123710          TST252: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14077  054262  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14078  054266  070034                          MUL     @(4)+,%0        ;MULTIPLY 125252 * @(4)+
14079  054270  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14080  054274  122737  000011  042142          CMPB    #11,@#PSWORD    ;IS PS = 11
14081  054302  001403                          BEQ     .+10
14082  054304  004767  004216                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14083                                                                  ;PS IS WRONG
14084  054310  000264                          264             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14085                                                          ;BY  (013746  000172  000207)
14086
14087  054312  022700  165252                  CMP     #165252,%0      ;IS HICH ORDER = 165252
14088  054316  001403                          BEQ     .+10
14089  054320  004767  004202                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14090                                                                  ;HIGH ORDER IS WRONG
14091  054324  000265                          265             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14092                                                          ;BY  (013746  000172  000207)
14093
14094  054326  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14095  054332  001403                          BEQ     .+10
14096  054334                          1$:
14097  054334  004767  004166                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14098                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14099  054340  000266                          266             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14100                                                          ;BY  (013746  000172  000207)
14101
14102  054342  021527  000252                  CMP     (R5),#252
14103  054346  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14104  054350  005215                          INC     (R5)
14105
14106
```

```
14107                                   ;*****************************************************************************
14108                                   ;TEST:253       MUL      125252 * @-(4) = 165252 100000        PS = 11
14109                                   ;*****************************************************************************
14110
14111  054352  010767  123614           TST253: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14112  054356  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14113  054362  070054                     MUL     @-(4),%0        ;MULTIPLY 125252 * @-(4)
14114  054364  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14115  054370  122737  000011  042142           CMPB    #11,@#PSWORD    ;IS PS = 11
14116  054376  001403                     BEQ     .+10
14117  054400  004767  004122                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14118                                                                   ;PS IS WRONG
14119  054404  000267                             267             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14120                                                                   ;BY  (013746  000172  000207)
14121
14122  054406  022700  165252                   CMP     #165252,%0      ;IS HIGH ORDER = 165252
14123  054412  001403                     BEQ     .+10
14124  054414  004767  004106                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14125                                                                   ;HIGH ORDER IS WRONG
14126  054420  000270                             270             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14127                                                                   ;BY  (013746  000172  000207)
14128
14129  054422  022701  100000                   CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14130  054426  001403                     BEQ     .+10
14131  054430                             1$:
14132  054430  004767  004072                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14133                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14134  054434  000271                             271             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14135                                                                   ;BY  (013746  000172  000207)
14136
14137  054436  021527  000253                   CMP     (R5),#253
14138  054442  001372                     BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14139  054444  005215                     INC     (R5)
14140
14141
```

```
14142                                   ;*********************************
14143                                   ;     DIV INSTRUCTION TESTS
14144                                   ;*********************************
14145
14146
14147
14148                           ;*************************************************************************
14149                           ;TEST:254      DIV     0 4 / #2 = 2        REM = 0      PS = 0
14150                           ;*************************************************************************
14151
14152  054446  010767  123520  TST254: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14153  054452  012700  000000          MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14154  054456  012701  000004          MOV     #4,%0+1         ;LOAD LOW ORDER WITH 4
14155  054462  071027  000002          DIV     #2,%0           ;DIVIDE BY #2
14156  054466  106737  042142          MFPS    @#PSWORD        ;SAVE PS
14157
14158  054472  122737  000000  042142  CMPB    #0,@#PSWORD     ;IS PS = 0
14159  054500  001403                  BEQ     .+10
14160  054502  004767  004020          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14161                                                          ;PS IS WRONG
14162  054506  000272                  272             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14163                                                  ;BY  (013746  000172  000207)
14164
14165
14166  054510  022700  000002          CMP     #2,%0           ;IS QUOTIENT = 2
14167  054514  001403                  BEQ     .+10
14168  054516  004767  004004          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14169                                                          ;QUOTIENT IS WRONG
14170  054522  000273                  273             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14171                                                  ;BY  (013746  000172  000207)
14172
14173
14174  054524  022701  000000          CMP     #0,%0+1         ;IS REMAINDER = 0
14175  054530  001403                  BEQ     .+10
14176  054532  004767  003770          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14177                                                          ;WRONG REMAINDER
14178  054536  000274                  274             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14179                                                  ;BY  (013746  000172  000207)
14180
14181  054540  021527  000254          CMP     (R5),#254
14182  054544  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14183  054546  004767  003754          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14184                                                          ;TEST IS IN WRONG SEQUENCE
14185  054552  000275                  275             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14186                                                  ;BY  (013746  000172  000207)
14187
14188  054554  005215                  INC     (R5)
14189
```

```
14190                                   ;*****************************************************************************
14191                                   ;TEST:255      DIV     -1 -9. / #3 = -3       REM = 0      PS = 10
14192                                   ;*****************************************************************************
14193
14194  054556  010767  123410           TST255: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14195  054562  012702  177777                   MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
14196  054566  012703  177767                   MOV     #-9.,%2+1       ;LOAD LOW ORDER WITH -9.
14197  054572  071227  000003                   DIV     #3,%2           ;DIVIDE BY #3
14198  054576  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14199
14200  054602  122737  000010  042142           CMPB    #10,@#PSWORD    ;IS PS = 10
14201  054610  001403                           BEQ     .+10
14202  054612  004767  003710                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14203                                                                   ;PS IS WRONG
14204  054616  000276                                   276             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14205                                                                   ;BY  (013746  000172  000207)
14206
14207
14208  054620  022702  177775                   CMP     #-3,%2          ;IS QUOTIENT = -3
14209  054624  001403                           BEQ     .+10
14210  054626  004767  003674                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14211                                                                   ;QUOTIENT IS WRONG
14212  054632  000277                                   277             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14213                                                                   ;BY  (013746  000172  000207)
14214
14215
14216  054634  022703  000000                   CMP     #0,%2+1         ;IS REMAINDER = 0
14217  054640  001403                           BEQ     .+10
14218  054642  004767  003660                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14219                                                                   ;WRONG REMAINDER
14220  054646  000300                                   300             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14221                                                                   ;BY  (013746  000172  000207)
14222
14223  054650  021527  000255                   CMP     (R5),#255
14224  054654  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14225  054656  004767  003644                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14226                                                                   ;TEST IS IN WRONG SEQUENCE
14227  054662  000301                                   301             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14228                                                                   ;BY  (013746  000172  000207)
14229
14230  054664  005215                           INC     (R5)
14231
```

```
14232                                   ;*************************************************************************
14233                                   ;TEST:256      DIV     0 9. / #2 = 4       REM = 1      PS = 0
14234                                   ;*************************************************************************
14235
14236  054666  010767  123300           TST256: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14237  054672  010501                           MOV     R5,R1           ;SAVE R5
14238  054674  012704  000000                   MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14239  054700  012705  000011                   MOV     #9.,%4+1        ;LOAD LOW ORDER WITH 9.
14240  054704  071427  000002                   DIV     #2,%4           ;DIVIDE BY #2
14241  054710  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14242
14243  054714  122737  000000  042142           CMPB    #0,@#PSWORD     ;IS PS = 0
14244  054722  001403                           BEQ     .+10
14245  054724  004767  003576                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14246                                                                   ;PS IS WRONG
14247  054730  000302                           302             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14248                                           ;BY  (013746  000172  000207)
14249
14250
14251  054732  022704  000004                   CMP     #4,%4           ;IS QUOTIENT = 4
14252  054736  001403                           BEQ     .+10
14253  054740  004767  003562                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14254                                                                   ;QUOTIENT IS WRONG
14255  054744  000303                           303             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14256                                           ;BY  (013746  000172  000207)
14257
14258
14259  054746  022705  000001                   CMP     #1,%4+1         ;IS REMAINDER = 1
14260  054752  001403                           BEQ     .+10
14261  054754  004767  003546                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14262                                                                   ;WRONG REMAINDER
14263  054760  000304                           304             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14264                                           ;BY  (013746  000172  000207)
14265
14266  054762  010105                   MOV     R1,R5           ;RESTORE R5
14267  054764  021527  000256           CMP     (R5),#256
14268  054770  001403                   BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14269  054772  004767  003530           JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14270                                                           ;TEST IS IN WRONG SEQUENCE
14271  054776  000305                   305             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14272                                   ;BY  (013746  000172  000207)
14273
14274  055000  005215                   INC     (R5)
14275
```

```
14276                                   ;*************************************************************************
14277                                   ;TEST:257       DIV     -1 -9. / #2 = -4        REM = -1        PS = 10
14278                                   ;*************************************************************************
14279
14280  055002  010767  123164          TST257: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14281  055006  012700  177777                  MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
14282  055012  012701  177767                  MOV     #-9.,%0+1       ;LOAD LOW ORDER WITH -9.
14283  055016  071027  000002                  DIV     #2,%0           ;DIVIDE BY #2
14284  055022  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14285
14286  055026  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
14287  055034  001403                          BEQ     .+10
14288  055036  004767  003464                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14289                                                                  ;PS IS WRONG
14290  055042  000306                          306             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14291                                                          ;BY  (013746  000172  000207)
14292
14293
14294  055044  022700  177774                  CMP     #-4,%0          ;IS QUOTIENT = -4
14295  055050  001403                          BEQ     .+10
14296  055052  004767  003450                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14297                                                                  ;QUOTIENT IS WRONG
14298  055056  000307                          307             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14299                                                          ;BY  (013746  000172  000207)
14300
14301
14302  055060  022701  177777                  CMP     #-1,%0+1        ;IS REMAINDER = -1
14303  055064  001403                          BEQ     .+10
14304  055066  004767  003434                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14305                                                                  ;WRONG REMAINDER
14306  055072  000310                          310             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14307                                                          ;BY  (013746  000172  000207)
14308
14309  055074  021527  000257                  CMP     (R5),#257
14310  055100  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14311  055102  004767  003420                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14312                                                                  ;TEST IS IN WRONG SEQUENCE
14313  055106  000311                          311             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14314                                                          ;BY  (013746  000172  000207)
14315
14316  055110  005215                          INC     (R5)
14317
```

```
14318                                   ;*****************************************************************************
14319                                   ;TEST:260      DIV     0 2 / #-3 = 0       REM = 2      PS = 4
14320                                   ;*****************************************************************************
14321
14322  055112  010767  123054          TST260: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14323  055116  012702  000000                  MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
14324  055122  012703  000002                  MOV     #2,%2+1         ;LOAD LOW ORDER WITH 2
14325  055126  071227  177775                  DIV     #-3,%2          ;DIVIDE BY #-3
14326  055132  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14327
14328  055136  122737  000004  042142          CMPB    #4,@#PSWORD     ;IS PS = 4
14329  055144  001403                          BEQ     .+10
14330  055146  004767  003354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14331                                                                  ;PS IS WRONG
14332  055152  000312                          312             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14333                                                          ;BY  (013746  000172  000207)
14334
14335
14336  055154  022702  000000                  CMP     #0,%2           ;IS QUOTIENT = 0
14337  055160  001403                          BEQ     .+10
14338  055162  004767  003340                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14339                                                                  ;QUOTIENT IS WRONG
14340  055166  000313                          313             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14341                                                          ;BY  (013746  000172  000207)
14342
14343
14344  055170  022703  000002                  CMP     #2,%2+1         ;IS REMAINDER = 2
14345  055174  001403                          BEQ     .+10
14346  055176  004767  003324                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14347                                                                  ;WRONG REMAINDER
14348  055202  000314                          314             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14349                                                          ;BY  (013746  000172  000207)
14350
14351  055204  021527  000260          CMP     (R5),#260
14352  055210  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14353  055212  004767  003310          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14354                                                          ;TEST IS IN WRONG SEQUENCE
14355  055216  000315                  315             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14356                                                  ;BY  (013746  000172  000207)
14357
14358  055220  005215                  INC     (R5)
14359
```

```
14360                                   ;*****************************************************************************
14361                                   ;TEST:261      DIV     -1 -2 / #3 = 0       REM = -2      PS - 4
14362                                   ;*****************************************************************************
14363
14364  055222  010767  122744           TST261: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14365  055226  010501                           MOV     R5,R1           ;SAVE R5
14366  055230  012704  177777                   MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
14367  055234  012705  177776                   MOV     #-2,%4+1        ;LOAD LOW ORDER WITH -2
14368  055240  071427  000003                   DIV     #3,%4           ;DIVIDE BY #3
14369  055244  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14370
14371  055250  122737  000004  042142          CMPB    #4,@#PSWORD     ;IS PS = 4
14372  055256  001403                           BEQ     .+10
14373  055260  004767  003242                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14374                                                                   ;PS IS WRONG
14375  055264  000316                           316             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14376                                           ;BY  (013746  000172  000207)
14377
14378
14379  055266  022704  000000                   CMP     #0,%4           ;IS QUOTIENT = 0
14380  055272  001403                           BEQ     .+10
14381  055274  004767  003226                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14382                                                                   ;QUOTIENT IS WRONG
14383  055300  000317                           317             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14384                                           ;BY  (013746  000172  000207)
14385
14386
14387  055302  022705  177776                   CMP     #-2,%4+1        ;IS REMAINDER = -2
14388  055306  001403                           BEQ     .+10
14389  055310  004767  003212                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14390                                                                   ;WRONG REMAINDER
14391  055314  000320                           320             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14392                                           ;BY  (013746  000172  000207)
14393
14394  055316  010105                   MOV     R1,R5           ;RESTORE R5
14395  055320  021527  000261           CMP     (R5),#261
14396  055324  001403                   BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14397  055326  004767  003174           JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14398                                                           ;TEST IS IN WRONG SEQUENCE
14399  055332  000321                   321             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14400                                   ;BY  (013746  000172  000207)
14401
14402  055334  005215                   INC     (R5)
14403
```

```
14404                                   ;*****************************************************************************
14405                                   ;TEST:262      DIV     -1 -1 / #1 = -1       REM = 0      PS = 10
14406                                   ;*****************************************************************************
14407
14408  055336  010767  122630           TST262: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14409  055342  012700  177777                   MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
14410  055346  012701  177777                   MOV     #-1,%0+1        ;LOAD LOW ORDER WITH -1
14411  055352  071027  000001                   DIV     #1,%0           ;DIVIDE BY #1
14412  055356  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14413
14414  055362  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
14415  055370  001403                           BEQ     .+10
14416  055372  004767  003130                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14417                                                                   ;PS IS WRONG
14418  055376  000322                           322             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14419                                           ;BY  (013746  000172  000207)
14420
14421
14422  055400  022700  177777                   CMP     #-1,%0          ;IS QUOTIENT = -1
14423  055404  001403                           BEQ     .+10
14424  055406  004767  003114                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14425                                                                   ;QUOTIENT IS WRONG
14426  055412  000323                           323             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14427                                           ;BY  (013746  000172  000207)
14428
14429
14430  055414  022701  000000                   CMP     #0,%0+1         ;IS REMAINDER = 0
14431  055420  001403                           BEQ     .+10
14432  055422  004767  003100                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14433                                                                   ;WRONG REMAINDER
14434  055426  000324                           324             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14435                                           ;BY  (013746  000172  000207)
14436
14437  055430  021527  000262                   CMP     (R5),#262
14438  055434  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14439  055436  004767  003064                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14440                                                                   ;TEST IS IN WRONG SEQUENCE
14441  055442  000325                           325             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14442                                           ;BY  (013746  000172  000207)
14443
14444  055444  005215                           INC     (R5)
14445
```

```
14446                                   ;*****************************************************************************
14447                                   ;TEST:263      DIV     0 0 / #1 = 0       REM = 0      PS = 4
14448                                   ;*****************************************************************************
14449
14450  055446  010767  122520           TST263: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14451  055452  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14452  055456  012701  000000                   MOV     #0,%0+1         ;LOAD LOW ORDER WITH 0
14453  055462  071027  000001                   DIV     #1,%0           ;DIVIDE BY #1
14454  055466  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
14455
14456  055472  122737  000004  042142         CMPB    #4,@#PSWORD     ;IS PS = 4
14457  055500  001403                 BEQ     .+10
14458  055502  004767  003020                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14459                                                                   ;PS IS WRONG
14460  055506  000326                           326             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14461                                           ;BY  (013746  000172  000207)
14462
14463
14464  055510  022700  000000                   CMP     #0,%0           ;IS QUOTIENT = 0
14465  055514  001403                 BEQ     .+10
14466  055516  004767  003004                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14467                                                                   ;QUOTIENT IS WRONG
14468  055522  000327                           327             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14469                                           ;BY  (013746  000172  000207)
14470
14471
14472  055524  022701  000000                   CMP     #0,%0+1         ;IS REMAINDER = 0
14473  055530  001403                 BEQ     .+10
14474  055532  004767  002770                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14475                                                                   ;WRONG REMAINDER
14476  055536  000330                           330             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14477                                           ;BY  (013746  000172  000207)
14478
14479  055540  021527  000263                   CMP     (R5),#263
14480  055544  001403                 BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14481  055546  004767  002754                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14482                                                                   ;TEST IS IN WRONG SEQUENCE
14483  055552  000331                           331             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14484                                           ;BY  (013746  000172  000207)
14485
14486  055554  005215                   INC     (R5)
14487
```

```
14488                                   ;*****************************************************************************
14489                                   ;TEST:264      DIV     -1 125252 / #2 = 152525       REM = 0      PS - 10
14490                                   ;*****************************************************************************
14491
14492  055556  010767  122410          TST264: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14493  055562  012702  177777                  MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
14494  055566  012703  125252                  MOV     #125252,%2+1    ;LOAD LOW ORDER WITH 125252
14495  055572  071227  000002                  DIV     #2,%2           ;DIVIDE BY #2
14496  055576  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14497
14498  055602  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
14499  055610  001403                          BEQ     .+10
14500  055612  004767  002710                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14501                                                                  ;PS IS WRONG
14502  055616  000332                          332             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14503                                                           ;BY  (013746  000172  000207)
14504
14505
14506  055620  022702  152525                  CMP     #152525,%2      ;IS QUOTIENT = 152525
14507  055624  001403                          BEQ     .+10
14508  055626  004767  002674                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14509                                                                  ;QUOTIENT IS WRONG
14510  055632  000333                          333             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14511                                                           ;BY  (013746  000172  000207)
14512
14513
14514  055634  022703  000000                  CMP     #0,%2+1         ;IS REMAINDER = 0
14515  055640  001403                          BEQ     .+10
14516  055642  004767  002660                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14517                                                                  ;WRONG REMAINDER
14518  055646  000334                          334             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14519                                                           ;BY  (013746  000172  000207)
14520
14521  055650  021527  000264                  CMP     (R5),#264
14522  055654  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14523  055656  004767  002644                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14524                                                                  ;TEST IS IN WRONG SEQUENCE
14525  055662  000335                          335             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14526                                                           ;BY  (013746  000172  000207)
14527
14528  055664  005215                  INC     (R5)
14529
```

```
14530                                   ;*****************************************************************************
14531                                   ;TEST:265       DIV     -1 -1 / #-1 = 1       REM = 0       PS = 0
14532                                   ;*****************************************************************************
14533
14534  055666  010767  122300          TST265: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14535  055672  010501                          MOV     R5,R1           ;SAVE R5
14536  055674  012704  177777                  MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
14537  055700  012705  177777                  MOV     #-1,%4+1        ;LOAD LOW ORDER WITH -1
14538  055704  071427  177777                  DIV     #-1,%4          ;DIVIDE BY #-1
14539  055710  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14540
14541  055714  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14542  055722  001403                          BEQ     .+10
14543  055724  004767  002576                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14544                                                                  ;PS IS WRONG
14545  055730  000336                          336             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14546                                                          ;BY  (013746  000172  000207)
14547
14548
14549  055732  022704  000001                  CMP     #1,%4           ;IS QUOTIENT = 1
14550  055736  001403                          BEQ     .+10
14551  055740  004767  002562                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14552                                                                  ;QUOTIENT IS WRONG
14553  055744  000337                          337             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14554                                                          ;BY  (013746  000172  000207)
14555
14556
14557  055746  022705  000000                  CMP     #0,%4+1         ;IS REMAINDER = 0
14558  055752  001403                          BEQ     .+10
14559  055754  004767  002546                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14560                                                                  ;WRONG REMAINDER
14561  055760  000340                          340             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14562                                                          ;BY  (013746  000172  000207)
14563
14564  055762  010105                  MOV     R1,R5           ;RESTORE R5
14565  055764  021527  000265          CMP     (R5),#265
14566  055770  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14567  055772  004767  002530          JSR     PC,$HLT,SEEN AN ERROR, GO TO THE HALT ROUTINE
14568                                                          ;TEST IS IN WRONG SEQUENCE
14569  055776  000341                  341             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14570                                                  ;BY  (013746  000172  000207)
14571
14572  056000  005215                  INC     (R5)
14573
```

```
14574                                   ;*************************************************************************
14575                                   ;TEST:266      DIV     25253 1 / #125252 = 100000       REM = 1      PS = 10
14576                                   ;*************************************************************************
14577
14578  056002  010767  122164          TST266: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14579  056006  012700  025253                  MOV     #25253,%0       ;LOAD HIGH ORDER WITH 25253
14580  056012  012701  000001                  MOV     #1,%0+1         ;LOAD LOW ORDER WITH 1
14581  056016  071027  125252                  DIV     #125252,%0      ;DIVIDE BY #125252
14582  056022  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14583
14584  056026  122737  000010  042142          CMPB    #10,@#PSWORD    ;IS PS = 10
14585  056034  001403                          BEQ     .+10
14586  056036  004767  002464                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14587                                                                  ;PS IS WRONG
14588  056042  000342                          342             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14589                                                          ;BY  (013746  000172  000207)
14590
14591
14592  056044  022700  100000                  CMP     #100000,%0      ;IS QUOTIENT = 100000
14593  056050  001403                          BEQ     .+10
14594  056052  004767  002450                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14595                                                                  ;QUOTIENT IS WRONG
14596  056056  000343                          343             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14597                                                          ;BY  (013746  000172  000207)
14598
14599
14600  056060  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14601  056064  001403                          BEQ     .+10
14602  056066  004767  002434                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14603                                                                  ;WRONG REMAINDER
14604  056072  000344                          344             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14605                                                          ;BY  (013746  000172  000207)
14606
14607  056074  021527  000266                  CMP     (R5),#266
14608  056100  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14609  056102  004767  002420                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14610                                                                  ;TEST IS IN WRONG SEQUENCE
14611  056106  000345                          345             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14612                                                          ;BY  (013746  000172  000207)
14613
14614  056110  005215                          INC     (R5)
14615
```

```
14616                                   ;*************************************************************************
14617                                   ;TEST:267      DIV     37777 77777 / #77777 = 77777       REM = 77776      PS = 0
14618                                   ;*************************************************************************
14619
14620  056112  010767  122054          TST267: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14621  056116  012702  037777                  MOV     #37777,%2       ;LOAD HIGH ORDER WITH 37777
14622  056122  012703  077777                  MOV     #77777,%2+1     ;LOAD LOW ORDER WITH 77777
14623  056126  071227  077777                  DIV     #77777,%2       ;DIVIDE BY #77777
14624  056132  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14625
14626  056136  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14627  056144  001403                          BEQ     .+10
14628  056146  004767  002354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14629                                                                  ;PS IS WRONG
14630  056152  000346                          346             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14631                                                          ;BY  (013746  000172  000207)
14632
14633
14634  056154  022702  077777                  CMP     #77777,%2       ;IS QUOTIENT = 77777
14635  056160  001403                          BEQ     .+10
14636  056162  004767  002340                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14637                                                                  ;QUOTIENT IS WRONG
14638  056166  000347                          347             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14639                                                          ;BY  (013746  000172  000207)
14640
14641
14642  056170  022703  077776                  CMP     #77776,%2+1     ;IS REMAINDER = 77776
14643  056174  001403                          BEQ     .+10
14644  056176  004767  002324                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14645                                                                  ;WRONG REMAINDER
14646  056202  000350                          350             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14647                                                          ;BY  (013746  000172  000207)
14648
14649  056204  021527  000267                  CMP     (R5),#267
14650  056210  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14651  056212  004767  002310                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14652                                                                  ;TEST IS IN WRONG SEQUENCE
14653  056216  000351                          351             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14654                                                          ;BY  (013746  000172  000207)
14655
14656  056220  005215                          INC     (R5)
14657
```

```
14658                                   ;*************************************************************************
14659                                   ;TEST:270      DIV     0 100000 / #2 = 40000       REM = 0      PS = 0
14660                                   ;*************************************************************************
14661
14662  056222  010767  121744          TST270: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14663  056226  010501                          MOV     R5,R1           ;SAVE R5
14664  056230  012704  000000                  MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14665  056234  012705  100000                  MOV     #100000,%4+1    ;LOAD LOW ORDER WITH 100000
14666  056240  071427  000002                  DIV     #2,%4           ;DIVIDE BY #2
14667  056244  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14668
14669  056250  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14670  056256  001403                          BEQ     .+10
14671  056260  004767  002242                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14672                                                                  ;PS IS WRONG
14673  056264  000352                          352             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14674                                                          ;BY  (013746  000172  000207)
14675
14676
14677  056266  022704  040000                  CMP     #40000,%4       ;IS QUOTIENT = 40000
14678  056272  001403                          BEQ     .+10
14679  056274  004767  002226                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14680                                                                  ;QUOTIENT IS WRONG
14681  056300  000353                          353             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14682                                                          ;BY  (013746  000172  000207)
14683
14684
14685  056302  022705  000000                  CMP     #0,%4+1         ;IS REMAINDER = 0
14686  056306  001403                          BEQ     .+10
14687  056310  004767  002212                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14688                                                                  ;WRONG REMAINDER
14689  056314  000354                          354             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14690                                                          ;BY  (013746  000172  000207)
14691
14692  056316  010105                  MOV     R1,R5           ;RESTORE R5
14693  056320  021527  000270          CMP     (R5),#270
14694  056324  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14695  056326  004767  002174          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14696                                                          ;TEST IS IN WRONG SEQUENCE
14697  056332  000355                  355             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14698                                                  ;BY  (013746  000172  000207)
14699
14700  056334  005215                  INC     (R5)
14701
```

```
14702                                   ;*************************************************************************
14703                                   ;TEST:271      DIV     177777 77777 / #177776 = 40000       REM = 177777      PS = 0
14704                                   ;*************************************************************************
14705
14706  056336  010767  121630          TST271: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14707  056342  012700  177777                  MOV     #177777,%0      ;LOAD HIGH ORDER WITH 177777
14708  056346  012701  077777                  MOV     #77777,%0+1     ;LOAD LOW ORDER WITH 77777
14709  056352  071027  177776                  DIV     #177776,%0      ;DIVIDE BY #177776
14710  056356  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14711
14712  056362  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14713  056370  001403                          BEQ     .+10
14714  056372  004767  002130                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14715                                                                  ;PS IS WRONG
14716  056376  000356                          356             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14717                                                          ;BY  (013746  000172  000207)
14718
14719
14720  056400  022700  040000                  CMP     #40000,%0       ;IS QUOTIENT = 40000
14721  056404  001403                          BEQ     .+10
14722  056406  004767  002114                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14723                                                                  ;QUOTIENT IS WRONG
14724  056412  000357                          357             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14725                                                          ;BY  (013746  000172  000207)
14726
14727
14728  056414  022701  177777                  CMP     #177777,%0+1    ;IS REMAINDER = 177777
14729  056420  001403                          BEQ     .+10
14730  056422  004767  002100                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14731                                                                  ;WRONG REMAINDER
14732  056426  000360                          360             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14733                                                          ;BY  (013746  000172  000207)
14734
14735  056430  021527  000271                  CMP     (R5),#271
14736  056434  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14737  056436  004767  002064                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14738                                                                  ;TEST IS IN WRONG SEQUENCE
14739  056442  000361                          361             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14740                                                          ;BY  (013746  000172  000207)
14741
14742  056444  005215                          INC     (R5)
14743
```

```
14744                                   ;*************************************************************************
14745                                   ;TEST:272      DIV     0 52525 / #52525 = 1       REM = 0      PS = 0
14746                                   ;*************************************************************************
14747
14748  056446  010767  121520          TST272: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14749  056452  012702  000000                  MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
14750  056456  012703  052525                  MOV     #52525,%2+1     ;LOAD LOW ORDER WITH 52525
14751  056462  071227  052525                  DIV     #52525,%2       ;DIVIDE BY #52525
14752  056466  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14753
14754  056472  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14755  056500  001403                          BEQ     .+10
14756  056502  004767  002020                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14757                                                                  ;PS IS WRONG
14758  056506  000362                          362             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14759                                                          ;BY  (013746  000172  000207)
14760
14761
14762  056510  022702  000001                  CMP     #1,%2           ;IS QUOTIENT = 1
14763  056514  001403                          BEQ     .+10
14764  056516  004767  002004                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14765                                                                  ;QUOTIENT IS WRONG
14766  056522  000363                          363             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14767                                                          ;BY  (013746  000172  000207)
14768
14769
14770  056524  022703  000000                  CMP     #0,%2+1         ;IS REMAINDER = 0
14771  056530  001403                          BEQ     .+10
14772  056532  004767  001770                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14773                                                                  ;WRONG REMAINDER
14774  056536  000364                          364             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14775                                                          ;BY  (013746  000172  000207)
14776
14777  056540  021527  000272                  CMP     (R5),#272
14778  056544  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14779  056546  004767  001754                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14780                                                                  ;TEST IS IN WRONG SEQUENCE
14781  056552  000365                          365             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14782                                                          ;BY  (013746  000172  000207)
14783
14784  056554  005215                          INC     (R5)
14785
```

```
14786                                   ;*****************************************************************************
14787                                   ;TEST:273      DIV     0 77777 / #0 = DUMMY       REM = DUMMY      PS = 3
14788                                   ;*****************************************************************************
14789
14790  056556  010767  121410          TST273: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14791  056562  010501                          MOV     R5,R1           ;SAVE R5
14792  056564  012704  000000                  MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14793  056570  012705  077777                  MOV     #77777,%4+1     ;LOAD LOW ORDER WITH 77777
14794  056574  071427  000000                  DIV     #0,%4           ;DIVIDE BY #0
14795  056600  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14796  056604  042737  000014  042142          BIC     #14,@#PSWORD
14797
14798  056612  122737  000003  042142          CMPB    #3,@#PSWORD     ;IS PS = 3
14799  056620  001403                          BEQ     .+10
14800  056622  004767  001700                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14801                                                                  ;PS IS WRONG
14802  056626  000366                          366             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14803                                                          ;BY  (013746  000172  000207)
14804
14805
14806  056630  010105                          MOV     R1,R5           ;RESTORE R5
14807  056632  021527  000273                  CMP     (R5),#273
14808  056636  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14809  056640  004767  001662                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14810                                                                  ;TEST IS IN WRONG SEQUENCE
14811  056644  000367                          367             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14812                                                          ;BY  (013746  000172  000207)
14813
14814  056646  005215                          INC     (R5)
14815
```

```
14816                                   ;*************************************************************************
14817                                   ;TEST:274       DIV     77777 177777 / #2 = DUMMY       REM = DUMMY      PS = 2
14818                                   ;*************************************************************************
14819
14820  056650  010767  121316          TST274: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14821  056654  012700  077777                  MOV     #77777,%0       ;LOAD HIGH ORDER WITH 77777
14822  056660  012701  177777                  MOV     #177777,%0+1    ;LOAD LOW ORDER WITH 177777
14823  056664  071027  000002                  DIV     #2,%0           ;DIVIDE BY #2
14824  056670  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14825  056674  042737  000014  042142          BIC     #14,@#PSWORD
14826
14827  056702  122737  000002  042142          CMPB    #2,@#PSWORD     ;IS PS = 2
14828  056710  001403                          BEQ     .+10
14829  056712  004767  001610                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14830                                                                  ;PS IS WRONG
14831  056716  000370                          370             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14832                                                          ;BY  (013746  000172  000207)
14833
14834
14835  056720  021527  000274                  CMP     (R5),#274
14836  056724  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14837  056726  004767  001574                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14838                                                                  ;TEST IS IN WRONG SEQUENCE
14839  056732  000371                          371             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14840                                                          ;BY  (013746  000172  000207)
14841
14842  056734  005215                          INC     (R5)
14843
```

```
14844  056736  012702  000002                  MOV     #2,%2
14845  056742  012703  042204                  MOV     #S9,%3
14846  056746  012704  042206                  MOV     #S10,%4
14847
14848                                  ;*****************************************************************************
14849                                  ;TEST:275       DIV     0 52525 / S9 = 25252       REM = 1      PS = 0
14850                                  ;*****************************************************************************
14851
14852  056752  010767  121214          TST275: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14853  056756  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14854  056762  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14855  056766  071067  163212                  DIV     S9,%0           ;DIVIDE BY S9
14856  056772  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14857
14858  056776  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14859  057004  001403                          BEQ     .+10
14860  057006  004767  001514                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14861                                                                  ;PS IS WRONG
14862  057012  000372                          372             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14863                                                          ;BY  (013746  000172  000207)
14864
14865
14866  057014  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14867  057020  001403                          BEQ     .+10
14868  057022  004767  001500                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14869                                                                  ;QUOTIENT IS WRONG
14870  057026  000373                          373             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14871                                                          ;BY  (013746  000172  000207)
14872
14873
14874  057030  022701  000001          CMP     #1,%0+1         ;IS REMAINDER = 1
14875  057034  001403                  BEQ     .+10
14876  057036  004767  001464          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14877                                                          ;WRONG REMAINDER
14878  057042  000374                  374             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14879                                                  ;BY  (013746  000172  000207)
14880
14881  057044  021527  000275          CMP     (R5),#275
14882  057050  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14883  057052  004767  001450          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14884                                                          ;TEST IS IN WRONG SEQUENCE
14885  057056  000375                  375             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14886                                                  ;BY  (013746  000172  000207)
14887
14888  057060  005215                  INC     (R5)
14889
```

```
14890                                   ;*************************************************************************
14891                                   ;TEST:276      DIV     0 52525 / @S10 = 25252       REM = 1      PS - 0
14892                                   ;*************************************************************************
14893
14894  057062  010767  121104          TST276: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14895  057066  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14896  057072  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14897  057076  071077  163104                  DIV     @S10,%0         ;DIVIDE BY @S10
14898  057102  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14899
14900  057106  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14901  057114  001403                          BEQ     .+10
14902  057116  004767  001404                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14903                                                                  ;PS IS WRONG
14904  057122  000376                          376             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14905                                                          ;BY  (013746  000172  000207)
14906
14907
14908  057124  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14909  057130  001403                          BEQ     .+10
14910  057132  004767  001370                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14911                                                                  ;QUOTIENT IS WRONG
14912  057136  000377                          377             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14913                                                          ;BY  (013746  000172  000207)
14914
14915
14916  057140  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14917  057144  001403                          BEQ     .+10
14918  057146  004767  001354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14919                                                                  ;WRONG REMAINDER
14920  057152  000400                          400             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14921                                                          ;BY  (013746  000172  000207)
14922
14923  057154  021527  000276                  CMP     (R5),#276
14924  057160  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14925  057162  004767  001340                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14926                                                                  ;TEST IS IN WRONG SEQUENCE
14927  057166  000401                          401             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14928                                                          ;BY  (013746  000172  000207)
14929
14930  057170  005215                          INC     (R5)
14931
```

```
14932                                   ;*****************************************************************************
14933                                   ;TEST:277      DIV     0 52525 / @#S9 = 25252       REM = 1      PS - 0
14934                                   ;*****************************************************************************
14935
14936  057172  010767  120774          TST277: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14937  057176  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14938  057202  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14939  057206  071037  042204                  DIV     @#S9,%0         ;DIVIDE BY @#S9
14940  057212  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
14941
14942  057216  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
14943  057224  001403                          BEQ     .+10
14944  057226  004767  001274                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14945                                                                  ;PS IS WRONG
14946  057232  000402                          402             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14947                                                          ;BY  (013746  000172  000207)
14948
14949
14950  057234  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14951  057240  001403                          BEQ     .+10
14952  057242  004767  001260                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14953                                                                  ;QUOTIENT IS WRONG
14954  057246  000403                          403             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14955                                                          ;BY  (013746  000172  000207)
14956
14957
14958  057250  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14959  057254  001403                          BEQ     .+10
14960  057256  004767  001244                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14961                                                                  ;WRONG REMAINDER
14962  057262  000404                          404             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14963                                                          ;BY  (013746  000172  000207)
14964
14965  057264  021527  000277                  CMP     (R5),#277
14966  057270  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14967  057272  004767  001230                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14968                                                                  ;TEST IS IN WRONG SEQUENCE
14969  057276  000405                          405             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14970                                                          ;BY  (013746  000172  000207)
14971
14972  057300  005215                          INC     (R5)
14973
```

```
14974                                   ;*****************************************************************
14975                                   ;TEST:300      DIV     0 52525 / %2 = 25252       REM   1      PS - 0
14976                                   ;*****************************************************************
14977
14978  057302  010767  120664         TST300: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14979  057306  012700  000000                 MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14980  057312  012701  052525                 MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14981  057316  071002                         DIV     %2,%0           ;DIVIDE BY %2
14982  057320  106737  042142                 MFPS    @#PSWORD        ;SAVE PS
14983
14984  057324  122737  000000  042142         CMPB    #0,@#PSWORD     ;IS PS = 0
14985  057332  001403                         BEQ     .+10
14986  057334  004767  001166                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14987                                                                  ;PS IS WRONG
14988  057340  000406                         406             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14989                                                          ;BY  (013746  000172  000207)
14990
14991
14992  057342  022700  025252                 CMP     #25252,%0       ;IS QUOTIENT = 25252
14993  057346  001403                         BEQ     .+10
14994  057350  004767  001152                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14995                                                                  ;QUOTIENT IS WRONG
14996  057354  000407                         407             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14997                                                          ;BY  (013746  000172  000207)
14998
14999
15000  057356  022701  000001                 CMP     #1,%0+1         ;IS REMAINDER = 1
15001  057362  001403                         BEQ     .+10
15002  057364  004767  001136                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15003                                                                  ;WRONG REMAINDER
15004  057370  000410                         410             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15005                                                          ;BY  (013746  000172  000207)
15006
15007  057372  021527  000300                 CMP     (R5),#300
15008  057376  001403                         BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15009  057400  004767  001122                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15010                                                                  ;TEST IS IN WRONG SEQUENCE
15011  057404  000411                         411             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15012                                                          ;BY  (013746  000172  000207)
15013
15014  057406  005215                         INC     (R5)
15015
```

```
15016                                   ;*************************************************************************
15017                                   ;TEST:301      DIV     0 52525 / (3)+ = 25252       REM = 1      PS = 0
15018                                   ;*************************************************************************
15019
15020  057410  010767  120556          TST301: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15021  057414  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15022  057420  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15023  057424  071023                          DIV     (3)+,%0         ;DIVIDE BY (3)+
15024  057426  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
15025
15026  057432  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
15027  057440  001403                          BEQ     .+10
15028  057442  004767  001060                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15029                                                                  ;PS IS WRONG
15030  057446  000412                          412             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15031                                                          ;BY  (013746  000172  000207)
15032
15033
15034  057450  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15035  057454  001403                          BEQ     .+10
15036  057456  004767  001044                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15037                                                                  ;QUOTIENT IS WRONG
15038  057462  000413                          413             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15039                                                          ;BY  (013746  000172  000207)
15040
15041
15042  057464  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER - 1
15043  057470  001403                          BEQ     .+10
15044  057472  004767  001030                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15045                                                                  ;WRONG REMAINDER
15046  057476  000414                          414             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15047                                                          ;BY  (013746  000172  000207)
15048
15049  057500  021527  000301                  CMP     (R5),#301
15050  057504  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15051  057506  004767  001014                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15052                                                                  ;TEST IS IN WRONG SEQUENCE
15053  057512  000415                          415             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15054                                                          ;BY  (013746  000172  000207)
15055
15056  057514  005215                  INC     (R5)
15057
```

```
15058                                   ;*****************************************************************************
15059                                   ;TEST:302      DIV     0 52525 / -(3) = 25252       REM - 1      PS - 0
15060                                   ;*****************************************************************************
15061
15062  057516  010767  120450           TST302: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15063  057522  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15064  057526  012701  052525                   MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15065  057532  071043                           DIV     -(3),%0         ;DIVIDE BY -(3)
15066  057534  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
15067
15068  057540  122737  000000  042142           CMPB    #0,@#PSWORD     ;IS PS = 0
15069  057546  001403                           BEQ     .+10
15070  057550  004767  000752                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15071                                                                   ;PS IS WRONG
15072  057554  000416                           416             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15073                                           ;BY  (013746  000172  000207)
15074
15075
15076  057556  022700  025252                   CMP     #25252,%0       ;IS QUOTIENT = 25252
15077  057562  001403                           BEQ     .+10
15078  057564  004767  000736                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15079                                                                   ;QUOTIENT IS WRONG
15080  057570  000417                           417             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15081                                           ;BY  (013746  000172  000207)
15082
15083
15084  057572  022701  000001                   CMP     #1,%0+1         ;IS REMAINDER = 1
15085  057576  001403                           BEQ     .+10
15086  057600  004767  000722                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15087                                                                   ;WRONG REMAINDER
15088  057604  000420                           420             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15089                                           ;BY  (013746  000172  000207)
15090
15091  057606  021527  000302                   CMP     (R5),#302
15092  057612  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15093  057614  004767  000706                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15094                                                                   ;TEST IS IN WRONG SEQUENCE
15095  057620  000421                           421             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15096                                           ;BY  (013746  000172  000207)
15097
15098  057622  005215                           INC     (R5)
15099
```

```
15100                                   ;*****************************************************************************
15101                                   ;TEST:303      DIV     0 52525 / 2(4) = 25252       REM = 1      PS = 0
15102                                   ;*****************************************************************************
15103
15104  057624  010767  120342           TST303: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15105  057630  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15106  057634  012701  052525                   MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15107  057640  071064  000002                   DIV     2(4),%0         ;DIVIDE BY 2(4)
15108  057644  106737  042142                   MFPS    @#PSWORD        ;SAVE PS
15109
15110  057650  122737  000000  042142           CMPB    #0,@#PSWORD     ;IS PS = 0
15111  057656  001403                           BEQ     .+10
15112  057660  004767  000642                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15113                                                                   ;PS IS WRONG
15114  057664  000422                           422             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15115                                           ;BY  (013746  000172  000207)
15116
15117
15118  057666  022700  025252                   CMP     #25252,%0       ;IS QUOTIENT = 25252
15119  057672  001403                           BEQ     .+10
15120  057674  004767  000626                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15121                                                                   ;QUOTIENT IS WRONG
15122  057700  000423                           423             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15123                                           ;BY  (013746  000172  000207)
15124
15125
15126  057702  022701  000001                   CMP     #1,%0+1         ;IS REMAINDER = 1
15127  057706  001403                           BEQ     .+10
15128  057710  004767  000612                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15129                                                                   ;WRONG REMAINDER
15130  057714  000424                           424             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15131                                           ;BY  (013746  000172  000207)
15132
15133  057716  021527  000303                   CMP     (R5),#303
15134  057722  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15135  057724  004767  000576                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15136                                                                   ;TEST IS IN WRONG SEQUENCE
15137  057730  000425                           425             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15138                                           ;BY  (013746  000172  000207)
15139
15140  057732  005215                           INC     (R5)
15141
```

```
15142                                   ;*************************************************************************
15143                                   ;TEST:304      DIV     0 52525 / @(4) = 25252       REM = 1      PS - 0
15144                                   ;*************************************************************************
15145
15146  057734  010767  120232          TST304: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15147  057740  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15148  057744  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15149  057750  071074  000000                  DIV     @(4),%0         ;DIVIDE BY @(4)
15150  057754  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
15151
15152  057760  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
15153  057766  001403                          BEQ     .+10
15154  057770  004767  000532                  JSR     %C,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15155                                                                  ;PS IS WRONG
15156  057774  000426                          426             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15157                                                          ;BY  (013746  000172  000207)
15158
15159
15160  057776  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15161  060002  001403                          BEQ     .+10
15162  060004  004767  000516                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15163                                                                  ;QUOTIENT IS WRONG
15164  060010  000427                          427             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15165                                                          ;BY  (013746  000172  000207)
15166
15167
15168  060012  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15169  060016  001403                          BEQ     .+10
15170  060020  004767  000502                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15171                                                                  ;WRONG REMAINDER
15172  060024  000430                          430             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15173                                                          ;BY  (013746  000172  000207)
15174
15175  060026  021527  000304                  CMP     (R5),#304
15176  060032  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15177  060034  004767  000466                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15178                                                                  ;TEST IS IN WRONG SEQUENCE
15179  060040  000431                          431             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15180                                                          ;BY  (013746  000172  000207)
15181
15182  060042  005215                          INC     (R5)
15183
```

```
15184                                   ;*************************************************************************
15185                                   ;TEST:305       DIV     0 52525 / @(4)+ = 25252       REM = 1      PS = 0
15186                                   ;*************************************************************************
15187
15188  060044  010767  120122          TST305: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15189  060050  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15190  060054  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15191  060060  071034                          DIV     @(4)+,%0        ;DIVIDE BY @(4)+
15192  060062  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
15193
15194  060066  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
15195  060074  001403                          BEQ     .+10
15196  060076  004767  000424                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15197                                                                  ;PS IS WRONG
15198  060102  000432                          432             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15199                                          ;BY  (013746  000172  000207)
15200
15201
15202  060104  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15203  060110  001403                          BEQ     .+10
15204  060112  004767  000410                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15205                                                                  ;QUOTIENT IS WRONG
15206  060116  000433                          433             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15207                                          ;BY  (013746  000172  000207)
15208
15209
15210  060120  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15211  060124  001403                          BEQ     .+10
15212  060126  004767  000374                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15213                                                                  ;WRONG REMAINDER
15214  060132  000434                          434             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15215                                          ;BY  (013746  000172  000207)
15216
15217  060134  021527  000305                  CMP     (R5),#305
15218  060140  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15219  060142  004767  000360                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15220                                                                  ;TEST IS IN WRONG SEQUENCE
15221  060146  000435                          435             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15222                                          ;BY  (013746  000172  000207)
15223
15224  060150  005215                          INC     (R5)
15225
```

```
15226                                   ;*************************************************************************
15227                                   ;TEST:306      DIV     0 52525 / @-(4) = 25252       REM = 1      PS = 0
15228                                   ;*************************************************************************
15229
15230  060152  010767  120014          TST306: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15231  060156  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15232  060162  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15233  060166  071054                          DIV     @-(4),%0        ;DIVIDE BY @-(4)
15234  060170  106737  042142                  MFPS    @#PSWORD        ;SAVE PS
15235
15236  060174  122737  000000  042142          CMPB    #0,@#PSWORD     ;IS PS = 0
15237  060202  001403                          BEQ     .+10
15238  060204  004767  000316                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15239                                                                  ;PS IS WRONG
15240  060210  000436                          436             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15241                                                          ;BY  (013746  000172  000207)
15242
15243
15244  060212  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15245  060216  001403                          BEQ     .+10
15246  060220  004767  000302                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15247                                                                  ;QUOTIENT IS WRONG
15248  060224  000437                          437             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15249                                                          ;BY  (013746  000172  000207)
15250
15251
15252  060226  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15253  060232  001403                          BEQ     .+10
15254  060234  004767  000266                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15255                                                                  ;WRONG REMAINDER
15256  060240  000440                          440             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15257                                                          ;BY  (013746  000172  000207)
15258
15259  060242  021527  000306                  CMP     (R5),#306
15260  060246  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15261  060250  004767  000252                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15262                                                                  ;TEST IS IN WRONG SEQUENCE
15263  060254  000441                          441             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15264                                                          ;BY  (013746  000172  000207)
15265
15266  060256  005215                          INC     (R5)
15267
```

```
15268  060260  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOCATION 4 IN STACK
15269  060264  012737  060300  000004          MOV     #1$,@#4         ;SET UP TIME OUT TRAP VECTOR
15270  060272  012737  000001  164000          MOV     #1,@#164000     ;TURN OFF MULTI-TESTER
15271  060300  012637  000004          1$:     MOV     (SP)+,@#4       ;RESTORE LOCATION 4
15272  060304  005227  177777                  INC     #-1             ;TYPE ID. THE FIRST PASS
15273  060310  001002                          BNE     SKPMSG          ;BRANCH AROUND AFTER THE FIRST PASS
15274  060312  104401  060346                  TYPE    ,MSG1
15275  060316  122767  000001  117774  SKPMSG: CMPB    #APTENV,$ENV
15276  060324  001433                          BEQ     $EOP
15277  060326  005267  001752                  INC     PASSPT          ;SHOULD PRINT THIS PASS?
15278  060332  022767  000017  001744          CMP     #17,PASSPT      ;DID 17-OCTAL PASSES YET?
15279  060340  001422                          BEQ     BREOP           ;IF YES, BRANCH TO EOP
15280  060342  000137  001024                  JMP     @#RESTRT        ;JUMP TO RESTART
15281  060346  005015  045103  042113  MSG1:   .ASCIZ  <15><12>*CJKDB-B DCF11-AA CPU DIAG.*<15><12>
15282  060354  026502  020102  041504
15283  060362  030506  026461  040501
15284  060370  041440  052520  042040
15285  060376  040511  027107  005015
15286  060404     000
15287          060406                  .EVEN
15288                                  .SBTTL ** ROUTINES LIST **
15289
15290  060406  012767  177777  001670  BREOP:  MOV     #-1,PASSPT      ;RESET PASSPT
15291
15292                                  .SBTTL  END OF PASS ROUTINE
15293
15294                                  ;*****************************************************************************************
15295                                  ;*INCREMENT THE PASS NUMBER ($PASS)
15296                                  ;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
15297                                  ;*IF THERES A MONITOR GO TO IT
15298                                  ;*IF THERE ISN'T JUMP TO RESTRT
15299
15300  060414                          $EOP:
15301  060414  000240                          SCOPE
15302  060416  005267  117664                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
15303  060422  042767  100000  117656          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
15304  060430  005327                          DEC     (PC)+           ;;LOOP?
15305  060432  000001          $EOPCT: .WORD   1
15306  060434  003022                          BGT     $DOAGN          ;;YES
15307  060436  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
15308  060440  000001          $ENDCT: .WORD   1
15309  060442  060432                          $EOPCT
15310  060444  104401  060511                  TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
15311  060450  016746  117632                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
15312  060454  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
15313  060456  104401  060506                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
15314  060462  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
15315  060466  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
15316  060470  000005                          RESET                   ;;CLEAR THE WORLD
15317  060472  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
15318  060474  000240                          NOP                     ;;SAVE ROOM
15319  060476  000240                          NOP                     ;;FOR
15320  060500  000240                          NOP                     ;;ACT11
15321  060502                          $DOAGN:
15322  060502  000137                          JMP     @(PC)+          ;;RETURN
15323  060504  001024          $RTNAD: .WORD   RESTRT
```

```
 15324  060506     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 15325  060511     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
 15326  060516  050040  051501  020123
 15327  060524  000043
 15328
 15329
 15330                                  .SBTTL  HALT ROUTINE
```

```
15331
15332                                   ;*      HALT ROUTINE
15333                                   ;*      ------------
15334                                   ;*
15335                                   ;*
15336                                   ;*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
15337                                   ;*
15338
15339  060526  017637  000000  000302  $HLT:   MOV     @(SP),@#$FATAL  ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
15340  060534  011637  060714          MOV     (SP),@#CONTIN   ;SAVE ERROR NUMBER ADDRESS
15341  060540  032777  020000  161444  BIT     #20000,@SWR     ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE OUTS
15342  060546  001021                  BNE     6$
15343  060550  104401  042224          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15344  060554  104401  060654          TYPE    ,MSGERR         ;INFORM ERROR
15345  060560  104401  042224          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15346  060564  013746  060714          MOV     @#CONTIN,-(SP)  ;RETREIVE ERROR NUMBER ADDR
15347  060570  162716  000004          SUB     #4,(SP)         ;CALCULATE ERROR PC
15348  060574  104402                  TYPOC                   ;TYPE PC
15349  060576  104401  042224          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15350  060602  013746  000302          MOV     @#$FATAL,-(SP)  ;RETREIVE ERROR NUMBER
15351  060606  104403                  TYPOS                   ;TYPE ERROR NUMBER
15352  060610     003                  .BYTE   3
15353  060611     000                  .BYTE   0
15354  060612  105767  117502  6$:     TSTB    $ENV            ;IF WE ARE NOT UNDER APT. THEN GO TO
15355  060616  001403                  BEQ     8$              ;8$
15356  060620  005237  000300          INC     @#$MSGTY        ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
15357  060624  000777                  BR      .               ;AND LOOP
15358  060626  104401  042224  8$:     TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15359  060632  005777  161354          TST     @SWR    ;IS IT REQUIRED TO HALT ON ERROR ?
15360  060636  100001                  BPL     10$             ;IF NOT THEN GO TO 10$
15361  060640  000000                  HALT
15362  060642  013746  060714  10$:     MOV     @#CONTIN,-(SP)  ;
15363  060646  062716  000002          ADD     #2,(SP)         ;CALCULATE RETURN ADDRESS
15364  060652  000207                  RTS     PC              ;RETURN
15365  060654  042440  051122  051117  MSGERR: .ASCIZ / ERROR!   PC, AND ERROR # ARE: /
15366  060662  020041  020040  041520
15367  060670  020054  047101  020104
15368  060676  051105  047522  020122
15369  060704  020043  051101  035105
15370  060712  000040
15371                           .EVEN
15372  060714  000000           CONTIN: .WORD 0
15373
15374                           .SBTTL  POWER FAIL ROUTINE
15375  060716  012767  060726  117100  PWRDN:  MOV     #PWRUP,24
15376  060724  000000                  HALT
15377
15378  060726  012767  060716  117070  PWRUP:  MOV     #PWRDN,24
15379  060734  012706  001000          MOV     #BUFF,SP
15380  060740  132767  000040  117353          BITB    #40,$ENVM       ;WILL APT ALLOW PRINTING?
15381  060746  001013                  BNE     PFRES           ;NO
15382  060750  012700  061002          MOV     #MSGPWF,R0      ;GET MSG ADDR.
15383  060754  105737  177564  PWAIT:  TSTB    @#TPS           ;TTY READY
15384  060760  100375                  BPL     PWAIT           ;NO WAIT
15385  060762  112037  177566          MOVB    (R0)+,@#TPB     ;PRINT CHARACTER
15386  060766  001372                  BNE     PWAIT           ;NEXT IF NOT DONE.
```

```
15387  060770  105737  177564           PWAIT1: TSTB    @#TPS
15388  060774  100375                           BPL     PWAIT1
15389  060776  000167  120022           PFRES:  JMP     RESTRT
15390  061002  005015  047520  042527   MSGPWF: .ASCIZ  <15><12>.POWER FAILED!.
15391  061010  020122  040506  046111
15392  061016  042105  000041
15393
15394                                    .EVEN
15395                                    .SBTTL  TYPE ROUTINE
15396
15397                                    ;*******************************************************************************************
15398                                    ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
15399                                    ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
15400                                    ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
15401                                    ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
15402                                    ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
15403                                    ;*
15404                                    ;*CALL:
15405                                    ;*1) USING A TRAP INSTRUCTION
15406                                    ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
15407                                    ;*OR
15408                                    ;*      TYPE
15409                                    ;*      MESADR
15410                                    ;*
15411  061022  105767  000261           $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
15412  061026  100002                           BPL     1$              ;;BR IF YES
15413  061030  000000                           HALT                    ;;HALT HERE IF NO TERMINAL
15414  061032  000430                           BR      3$              ;;LEAVE
15415  061034  010046                   1$:     MOV     R0,-(SP)        ;;SAVE R0
15416  061036  017600  000002                   MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
15417  061042  122767  000001  117250           CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
15418  061050  001011                           BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
15419  061052  132767  000100  117241           BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
15420  061060  001405                           BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
15421  061062  010067  000004                   MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
15422  061066  004767  000230                   JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
15423  061072  000000                   61$:    .WORD   0               ;;MESSAGE ADDRESS
15424  061074  132767  000040  117217   62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
15425  061102  001003                           BNE     60$             ;;YES,SKIP TYPE OUT
15426  061104  112046                   2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
15427  061106  001005                           BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
15428  061110  005726                           TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
15429  061112  012600                   60$:    MOV     (SP)+,R0        ;;RESTORE R0
15430  061114  062716  000002           3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
15431  061120  000002                           RTI                     ;;RETURN
15432  061122  122716  000011           4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
15433  061126  001430                           BEQ     8$
15434  061130  122716  000200                   CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
15435  061134  001006                           BNE     5$
15436  061136  005726                           TST     (SP)+           ;;POP  <CR><LF> EQUIV
15437  061140  104401                           TYPE                    ;;TYPE A CR AND LF
15438  061142  042224                           $CRLF
15439  061144  105067  000130                   CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
15440  061150  000755                           BR      2$              ;;GET NEXT CHARACTER
15441  061152  004767  000056           5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
15442  061156  126726  000124           6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
```

```
15443  061162  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
15444  061164  016746  000114                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
15445                                                                  ;;AND THE NULL CHAR.
15446  061170  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
15447  061174  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
15448  061176  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
15449  061202  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
15450  061206  000770                          BR      7$              ;;LOOP
15451
15452                                  ;HORIZONTAL TAB PROCESSOR
15453
15454  061210  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
15455  061214  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
15456  061220  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
15457  061226  001372                          BNE     9$              ;;TAB STOP
15458  061230  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
15459  061232  000724                          BR      2$              ;;GET NEXT CHARACTER
15460  061234  105777  160762          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
15461  061240  100375                          BPL     $TYPEC
15462  061242  116677  000002  160750          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
15463  061250  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
15464  061256  001003                          BNE     1$              ;;BRANCH IF NO
15465  061260  105067  000014                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
15466  061264  000406                          BR      $TYPEX          ;;EXIT
15467  061266  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
15468  061274  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
15469  061276  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
15470  061300  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
15471  061302  000207                  $TYPEX: RTS     PC
15472
15473  061304      000                 $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
15474  061305      002                 $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
15475  061306      012                 $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
15476  061307      000                 $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>-0=YES)
15477  061310      077                 $QUES:  .ASCII  ''?''           ;;QUESTION MARK
15478  061311      012     000         $LF:    .ASCIZ  <12>            ;;LINEFEED
15479          061314                          .EVEN
15480                                  .SBTTL  APT COMMUNICATIONS ROUTINE
15481
15482                                  ;*****************************************************************************************
15483  061314  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
15484  061322  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
15485  061330  000403                          BR      $ATYC
15486  061332  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
15487  061340                          $ATYC:
15488  061340  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
15489  061342  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
15490  061344  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
15491  061350  001450                          BEQ     5$              ;;IF NOT:  BR
15492  061352  122767  000001  116740          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
15493  061360  001031                          BNE     3$              ;;IF NOT:  BR
15494  061362  132767  000100  116731          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
15495  061370  001425                          BEQ     3$              ;;IF NOT:  BR
15496  061372  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
15497  061376  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
15498  061404  005767  116670          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
```

```
15499  061410  001375                        BNE     1$              ;;IF NOT:  WAIT
15500  061412  010067  116676                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
15501  061416  105720                2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
15502  061420  001376                        BNE     2$
15503  061422  166700  116666                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
15504  061426  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
15505  061430  010067  116662                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
15506  061434  012767  000004  116636        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
15507  061442  000413                        BR      5$
15508  061444  017667  000004  000016  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
15509  061452  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
15510  061460  016746  116312                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
15511  061464  004767  177332                JSR     PC,$TYPE        ;;CALL TYPE MACRO
15512  061470  000000                4$:     .WORD   0
15513  061472                        5$:
15514  061472  105767  000062        10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
15515  061476  001416                        BEQ     12$             ;;IF NOT:  BR
15516  061500  005767  116614                TST     $ENV            ;;RUNNING UNDER APT?
15517  061504  001413                        BEQ     12$             ;;IF NOT:  BR
15518  061506  005767  116566        11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
15519  061512  001375                        BNE     11$             ;;IF NOT:  WAIT
15520  061514  017667  000004  116560        MOV     @4(SP),$FATAL   ;;GET ERROR #
15521  061522  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
15522  061530  005267  116544                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
15523  061534  105067  000020        12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
15524  061540  105067  000013                CLRB    $LFLG           ;;CLEAR LOG FLAG
15525  061544  105067  000006                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
15526  061550  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
15527  061552  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
15528  061554  000207                        RTS     PC              ;;RETURN
15529  061556     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
15530  061557     000          $LFLG:  .BYTE   0               ;;LOG FLAG
15531  061560     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
15532          061562                  .EVEN
15533          000200          APTSIZE=200
15534          000001          APTENV=001
15535          000100          APTSPOOL=100
15536          000040          APTCSUP=040
15537                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
15538
15539                          ;*************************************************************************************
15540                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15541                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15542                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15543                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15544                          ;*REPLACED WITH SPACES.
15545                          ;*CALL:
15546                          ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
15547                          ;*      TYPDS                   ;;GO TO THE ROUTINE
15548
15549  061562                  $TYPDS:
15550  061562  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
15551  061564  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
15552  061566  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
15553  061570  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
15554  061572  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
 15555  061574  012746  020200                       MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 15556  061600  016605  000020                       MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 15557  061604  100004                               BPL     1$              ;;BR IF INPUT IS POS.
 15558  061606  005405                               NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 15559  061610  112766  000055  000001               MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 15560  061616  005000                        1$:    CLR     R0              ;;ZERO THE CONSTANTS INDEX
 15561  061620  012703  061776                       MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 15562  061624  112723  000040                       MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 15563  061630  005002                        2$:    CLR     R2              ;;CLEAR THE BCD NUMBER
 15564  061632  016001  061766                       MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 15565  061636  160105                        3$:    SUB     R1,R5           ;;FORM THIS BCD DIGIT
 15566  061640  002402                               BLT     4$              ;;BR IF DONE
 15567  061642  005202                               INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 15568  061644  000774                               BR      3$
 15569  061646  060105                        4$:    ADD     R1,R5           ;;ADD BACK THE CONSTANT
 15570  061650  005702                               TST     R2              ;;CHECK IF BCD DIGIT=0
 15571  061652  001002                               BNE     5$              ;;FALL THROUGH IF 0
 15572  061654  105716                               TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 15573  061656  100407                               BMI     7$              ;;BR IF YES
 15574  061660  106316                        5$:    ASLB    (SP)            ;;MSD?
 15575  061662  103003                               BCC     6$              ;;BR IF NO
 15576  061664  116663  000001  177777               MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 15577  061672  052702  000060                6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 15578  061676  052702  000040                7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 15579  061702  110223                               MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 15580  061704  005720                               TST     (R0)+           ;;JUST INCREMENTING
 15581  061706  020027  000010                       CMP     R0,#10          ;;CHECK THE TABLE INDEX
 15582  061712  002746                               BLT     2$              ;;GO DO THE NEXT DIGIT
 15583  061714  003002                               BGT     8$              ;;GO TO EXIT
 15584  061716  010502                               MOV     R5,R2           ;;GET THE LSD
 15585  061720  000764                               BR      6$              ;;GO CHANGE TO ASCII
 15586  061722  105726                        8$:    TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 15587  061724  100003                               BPL     9$              ;;BR IF NO
 15588  061726  116663  177777  177776               MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 15589  061734  105013                        9$:    CLRB    (R3)            ;;SET THE TERMINATOR
 15590  061736  012605                               MOV     (SP)+,R5        ;;POP STACK INTO R5
 15591  061740  012603                               MOV     (SP)+,R3        ;;POP STACK INTO R3
 15592  061742  012602                               MOV     (SP)+,R2        ;;POP STACK INTO R2
 15593  061744  012601                               MOV     (SP)+,R1        ;;POP STACK INTO R1
 15594  061746  012600                               MOV     (SP)+,R0        ;;POP STACK INTO R0
 15595  061750  104401  061776                       TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 15596  061754  016666  000002  000004               MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 15597  061762  012616                               MOV     (SP)+,(SP)
 15598  061764  000002                               RTI                     ;;RETURN TO USER
 15599  061766  023420                $DTBL:  10000.
 15600  061770  001750                        1000.
 15601  061772  000144                        100.
 15602  061774  000012                        10.
 15603  061776  000004                $DBLK:  .BLKW   4
 15604                                 .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 15605
 15606                                 ;*******************************************************************************
 15607                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 15608                                 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 15609                                 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 15610                                 ;*CALL:
```

```
15611                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15612                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
15613                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
15614                                   ;*      .BYTE   M               ;;M=1 OR 0
15615                                   ;*                                      ;;1=TYPE LEADING ZEROS
15616                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
15617                                   ;*
15618                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
15619                                   ;*$TYPOS OR $TYPOC
15620                                   ;*CALL:
15621                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15622                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
15623                                   ;*
15624                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
15625                                   ;*CALL:
15626                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15627                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
15628
15629  062006  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
15630  062012  116667  000001  000211           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
15631  062020  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
15632  062024  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
15633  062030  000406                          BR      $TYPON
15634  062032  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
15635  062040  112767  000006  000165           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
15636  062046  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
15637  062054  010346                          MOV     R3,-(SP)        ;;SAVE R3
15638  062056  010446                          MOV     R4,-(SP)        ;;SAVE R4
15639  062060  010546                          MOV     R5,-(SP)        ;;SAVE R5
15640  062062  116704  00J145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
15641  062066  005404                          NEG     R4
15642  062070  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
15643  062074  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
15644  062100  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
15645  062104  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
15646  062110  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
15647  062112  006105                   1$:    ROL     R5              ;;ROTATE MSB INTO ''C''
15648  062114  000404                          BR      3$              ;;GO DO MSB
15649  062116  006105                   2$:    ROL     R5              ;;FORM THIS DIGIT
15650  062120  006105                          ROL     R5
15651  062122  006105                          ROL     R5
15652  062124  010503                          MOV     R5,R3
15653  062126  006103                   3$:    ROL     R3              ;;GET LSB OF THIS DIGIT
15654  062130  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
15655  062134  100016                          BPL     7$              ;;BR IF NO
15656  062136  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
15657  062142  001002                          BNE     4$              ;;TEST FOR 0
15658  062144  005704                          TST     R4              ;;SUPPRESS THIS 0?
15659  062146  001403                          BEQ     5$              ;;BR IF YES
15660  062150  005204                   4$:    INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
15661  062152  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
15662  062156  052703  000040           5$:    BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
15663  062162  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
15664  062166  104401  062226                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
15665  062172  105367  000032           7$:    DECB    $OCNT           ;;COUNT BY 1
15666  062176  003347                          BGT     2$              ;;BR IF MORE TO DO
```

```
15667  062200  002402                          BLT     6$              ;;BR IF DONE
15668  062202  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
15669  062204  000744                          BR      2$              ;;GO DO THE LAST DIGIT
15670  062206  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
15671  062210  012604                          MOV     (SP)+,R4        ;;RESTORE R4
15672  062212  012603                          MOV     (SP)+,R3        ;;RESTORE R3
15673  062214  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
15674  062222  012616                          MOV     (SP)+,(SP)
15675  062224  000002                          RTI                     ;;RETURN
15676  062226     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
15677  062227     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
15678  062230     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
15679  062231     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
15680  062232  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
15681                                  .SBTTL  TRAP DECODER
15682
15683                                  ;*******************************************************************************************
15684                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
15685                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
15686                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
15687                                  ;*GO TO THAT ROUTINE.
15688
15689  062234  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
15690  062236  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
15691  062242  005740                          TST     -(R0)           ;;BACKUP BY 2
15692  062244  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
15693  062246  006300                          ASL     R0              ;;POSITION FOR INDEXING
15694  062250  016000  062270                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
15695  062254  000200                          RTS     R0              ;;GO TO ROUTINE
15696
15697
15698                                  ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
15699
15700  062256  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
15701  062260  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
15702  062266  000002                          RTI                     ;;RESTORE THE PSW
15703
15704                                  .SBTTL  TRAP TABLE
15705
15706                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
15707                                  ;*BY THE ''TRAP'' INSTRUCTION.
15708
15709                                  ;       ROUTINE
15710                                  ;       -------
15711  062270  062256                  $TRPAD: .WORD   $TRAP2
15712  062272  061022                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
15713  062274  062032                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
15714  062276  062006                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
15715  062300  062046                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
15716  062302  061562                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
15717
15718
15719  062304  177777                  PASSPT: -1
15720          000001                  .END
```

```
A       026054
ABASE = 000000
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
ADC1    020374
ADC2    020404
ADC3    020424
ADC4    020434
ADC5    020452
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADD1    020210
ADD2    020220
ADD3    020234
ADD4    020244
ADD5    020262
ADD6    020272
ADD7    020304
ADD8    020314
ADD9    020334
ADEVCT= 000000
ADEVM = 000000
AENV  = 000000
AENVM = 000000
AFATAL= 000000
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
APASS = 000000
APRIOR= 000000
APTCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPO= 000100
AROUN   025252
AROUND  041676
ASL1    021710
ASL2    021720
ASL3    021736
ASL4    021746
ASL5    021762
ASL6    021772
ASL7    022016
ASR1    022060
ASR2    022070
ASR3    022112
ASR4    022122
ASR5    022136
ASR6    022146
ASR7    022176
ASTART  042402
ASWREG= 000000
ATESTN= 000000
ATRAP   037502
AUNIT = 000000
AUSWR = 000000
AUTO1   037610
AVECT1= 000000
AVECT2= 000000
BELL  = 000240
BIC1    017340
BIC2    017350
BIC3    017366
BIS1    017430
BIS2    017440
BIS3    017460
BITCHK  025442
BITCLR  025370
BITCON  025524
BITSET  025406
BIT1    017250
BIT2    017260
BIT3    017276
BRA1    001074
BRA2    001104
BRA3    001116
BRA4    001124
BRA5    001134
BRCT    025310
BRC1    003342
BRC2    003352
BRC3    003362
BREOP   060406
BRH     025262
BRMFPD  027074
BRN1    003222
BRN2    003232
BRN3    003242
BRTAB   027170
BRV1    003272
BRV2    003302
BRV3    003312
BRZ1    003152
BRZ2    003162
BRZ3    003172
BR1     027464
BR10    027700
BR11    030004
BR12    030052
BR13    030120
BR14    030166
BR15    030264
BR16    030304
BR17    030324
BR2     027512
BR20    030344
BR21    030364
BR22    030404
BR23    030424
BR3     027534
BR33    030474
BR34    030506
BR35    030520
BR36    030532
BR37    030554
BR4     027560
BR40    030566
BR41    030600
BR45    032126
BR46    032674
BR46A   032704
BR47    033464
BR5     027604
BR51    034232
BR51A   034242
BR6     027630
BR7     027654
BR70    037220
BR71    040150
BTCON   025600
BTERR   025564
BTRAP   037556
BUFF  = 001000
CC    = 177776
CCERR   026020
CC1     025256
CC2     025722
CC3     025706
CLRCD   025704
CLR1    017776
CMP1    020640
CMP2    020650
CMP3    020672
CMP4    020702
CMP5    020726
CMP6    020736
CMP7    020756
COM1    021016
CONT    025312
CONTIN  060714
CON1    025736
CON2    026030
CORH    037542
COUNT   042140
CPUTYP  041524
CR    = 000015
CRLF  = 000200
CTRAP   037520
DAERR   025514
DBE     041340
DBE1    041362
DBE2    041352
DBE3    041402
DBE4    041422
DBE5    041440
DEC1    017636
DEC2    017646
DEC3    017662
DEC4    017672
DEC5    017706
DEC6    017716
DEC7    017740
DISPLA  042214
DISPRE  000174
DL11W   037660
DL11W1  037664
DNMB0A  011036
DNMB0B  011046
DNMB2A  011274
DNMB2B  011304
DNMB2C  011320
DNMB2D  011334
DNMB2E  011344
DNMB2F  011362
DNMB3A  011444
DNMB3B  011454
DNMB3C  011472
DNMB3D  011510
DNMB3E  011520
DNMB4A  011710
DNMB4B  011720
DNMB4C  011736
DNMB4D  011746
DNMB4E  011756
DNMB4F  011772
DNM03A  010130
DNM03B  010140
DNM03C  010150
DNM1    010002
DNM1A   011114
DNM1B   011124
DNM2    010016
DNM2A   011172
DNM2B   011202
DNM2C   011210
DNM2D   011220
DNM3    010034
DNM4    010056
DNM4A   011602
DNM4B   011612
DNM4C   011626
DNM5A   012052
DNM5B   012062
DNM5C   012100
DNM6A   012160
DNM6B   012170
DNM6C   012206
DNM7A   012270
DNM7B   012300
DNM7C   012316
DOPB2A  010542
DOPB2B  010620
DOP0A   007544
DOP0B   007570
DOP0C   007610
DOP0D   007640
DOP03A  007716
DOP03B  007726
DOP1    010354
DOP2    010466
DOP4    013772
DOP5    014056
DTRAP1  037540
DUMMY = 000000
END1    027160
ENT176  046614
ENT51   044154
ER      025274
```

```
ERRNM = 000442
F     = 000063
FINISH  042124
FIRST =%000005
FOVER   036446
FPP     042114
GIN1    041716
GIN2    041734
GIN3    041762
HERE  = 000000
HICORE  037412
HLT   = 000000
HT    = 000011
IJMP    016354
IJMP4   016126
IJMP5   016316
ILLA  = 004700
ILLB  = 000100
INC1    017520
INC2    017530
INC3    017552
INC4    017562
INC5    017576
INST    042126
INSTC   030732
INSTK   035110
ITRAP5= 000004
JMPCK   016356
JMPERR  025634
JMPSEQ  016376
JMPT    025624
JMP2    016130
JMP2A   016146
JMP3    016056
JMP3A   016074
JMP3B   016114
JMP4    016160
JMP4A   016176
JMP4B   016216
JMP5    016264
JMP5A   016304
JMP6    016230
JMP6A   016250
JMP7    016320
JMP7A   016340
JSRCK   017030
JSRCKA  017024
JSRCK1  017046
JSRSEQ  017026
JSR0    016412
JSR1    016416
JSR1A   016440
JSR2    016530
JSR2A   016560
JSR2B   016570
JSR3    016450
JSR3A   016510
JSR3B   016520
JSR4    016612
JSR4A   016630
JSR4B   016640
JSR5    016716
JSR5A   016742
JSR5B   016752
JSR6    016652
JSR6A   016676
JSR6AD  017022
JSR6B   016706
JSR7    016764
JSR7A   017002
JSR7B   017012
K1      027404
K10     027422
K11     027424
K12     027426
K2      027406
K3      027410
K4      027412
K5      027414
K6      027416
K7      027420
LAST  =%000001
LF    = 000012
LPADR   000172
MBDM2A  012534
MBDM2B  012544
MBDM2C  012562
MBDM2D  012574
MBDM2E  012604
MBDM2F  012622
MBDM4A  013156
MBDM4B  013174
MBDM4C  013206
MBDM4D  013216
MBDM4E  013232
MDM1A   012366
MDM1B   012376
MDM2A   012442
MDM2B   012452
MDM2C   012460
MDM2D   012470
MDM3A   012676
MDM3B   012706
MDM3C   012724
MDM3D   012744
MDM3E   012772
MDM4A   013052
MDM4B   013062
MDM4C   013076
MDM5A   013302
MDM5B   013312
MDM5C   013330
MDM5D   013346
MDM5E   013374
MDM6A   013444
MDM6B   013454
MDM6C   013472
MDM6D   013512
MDM6E   013542
MDM7A   013616
MDM7B   013626
MDM7C   013644
MDM7D   013664
MDM7E   013710
MFPI0   024764
MFPI0A  025012
MFPS1   023524
MFPS2A  023614
MFPS2B  023624
MFPS2C  023644
MFPS3A  023722
MFPS3B  023732
MFPS3C  023752
MFPS4A  024030
MFPS4B  024040
MFPS4C  024060
MFPS5A  024136
MFPS5B  024146
MFPS5C  024166
MFPS6A  024246
MFPS6B  024256
MFPS6C  024276
MFPS7A  024356
MFPS7B  024366
MFPS7C  024406
MFPT  = 000007
MOR0    026062
MOR1    026122
MOR2    026202
MOR3    026274
MOR4    026314
MOR5    026334
MOR6    026424
MOR7    026444
MOR8    026464
MOV1    017160
MOV2    017170
MOV3    017206
MRK1    022564
MRK2    022606
MRK3    022616
MRK4    022640
MRK5    022652
MRK6    022666
MSGERR  060654
MSGPWF  061002
MSG1    060346
MTPI0   025076
MTPS1   022736
MTPS1A  022756
MTPS2   023032
MTPS3   023122
MTPS4   023210
MTPS5   023270
MTPS6   023360
MTPS7   023450
N     = 000307
NBR     025266
NEGAT   043554
NEG00   004312
NEG01   004322
NEG02   004336
NEG03   004352
NEG04   004362
NEG1    020514
NEG10   004426
NEG11   004436
NEG12   004454
NEG13   004470
NEG14   004500
NEG2    020524
NEG20   004546
NEG21   004556
NEG22   004604
NEG3    020546
NEG30   005126
NEG31   005136
NEG32   005152
NEG33   005176
NEG34   005212
NEG4    020556
NEG40   005604
NEG41   005614
NEG42   005630
NEG5    020576
NEG50   005704
NEG51   005714
NEG52   005730
NEG60   006006
NEG61   006016
NEG70   006066
NEG71   006076
NODL    040172
NODL1   040356
NODL2   040440
NODL3   040710
NOP   = 000240
NOR     037406
NOSUB   037524
PASSPT  062304
PCN01   026516
PCN1    041116
PCN2    026552
PCN3    026620
PCN4    026662
PCN5    026722
PFRES   060776
POWER   042232
PROMTE  037656
PS    = 177776
PSWORD  042142
PUSRM = 030000
PWAIT   060754
PWAIT1  060770
PWRDN   060716
PWRUP   060726
RA      033474
RA1     032136
RB      033462
RB1     032124
RC      033456
RC1     032120
REGR1   042550
REGR2   042716
REGR23  045426
REGR3   043064
REGR4   043230
REGR5   043400
REG01   045226
REG1    002032
REG1A   002076
REG1E   002044
REG2    002146
REG2A   002174
REG2B   002222
REG2C   002254
REG3    002302
REG3A   002346
REG3E   002314
REG4    002416
REG4A   002462
REG4E   002430
REG45   045626
REG5    002532
REG5A   002576
```

```
REG5E   002544
REG6    002646
REG6A   002712
REG6E   002660
RESET2  040510
RESET3  040500
REST    024464
RESTRT  001024
RET     042006
RETA    030636
RETAH   030646
RETAT   036536
RETA1   031412
RETA2   032220
RETA3   032752
RETA4   033556
RETA5   034310
RETB    030672
RETBT   036604
RETB1   031436
RETB2   032244
RETB3   032776
RETB4   033602
RETB5   034334
RETC    030734
RETCT   036656
RETC1   031500
RETC2   032306
RETC3   033040
RETC4   033644
RETC5   034376
RETD    031006
RETD1   031552
RETD2   032360
RETD3   033112
RETD4   033716
RETD5   034450
RETE    031052
RETE1   031614
RETE2   032424
RETE3   033156
RETE4   033762
RETE5   034514
RETF    031122
RETF1   031664
RETF2   032474
RETF3   033226
RETF4   034032
RETF5   034564
RETG    031234
RETG1   031776
RETG2   032606
RETG3   033340
RETG4   034144
RETG5   034676
RETH5   035024
RETJ    035050
RETK    035112
RETL    035164
RETM    035230
RETN    035300
RETO    035410
RETR1   041132
RETR2   041210
RETR3   041256
RET1    042024
RET2    042044
RET3    042062
RET4    042052
RITSH   046036
ROL1    021400
ROL2    021410
ROL3    021426
ROL4    021436
ROL5    021452
ROL6    021462
ROL7    021504
ROR1    021546
ROR2    021556
ROR3    021574
ROR4    021604
ROR5    021622
ROR6    021632
ROR7    021646
ROTX    015164
ROTXAD  015310
ROT0A   014302
ROT0B   014312
ROT0C   014334
ROT1A   014402
ROT1B   014412
ROT1C   014436
ROT1D   014446
ROT1E   014476
ROT2A   014550
ROT2B   014560
ROT2C   014610
ROT2D   014620
ROT2E   014654
ROT3A   014722
ROT3B   014732
ROT3C   014760
ROT3D   014770
ROT3E   015016
ROT4    015072
ROT5    015154
ROT6    015224
ROT7    015300
RTI1    037106
RTI2    037120
RTRAP =  000010
RTRAP1=  000034
RTRAP2=  000020
RTRAP3=  000030
RTRAP4=  000014
RTRAP5=  000004
RTS1    017114
RTT1    036734
RTT2    036750
RTT3    037012
RTT4    037030
RTT5    036764
RTT6    037050
ROTRAP  037440
R1ERR   002114
R2ERR   002244
R3ERR   002364
R4ERR   002500
R5ERR   002614
R6      =%000006
R6ERR   002730
R7      =%000007
R7TRX   040026
SBC1    021232
SBC2    021242
SBC3    021260
SBC4    021270
SBC5    021306
SBC6    021316
SBC7    021336
SB0     015342
SB2     015464
SB4     015610
SB5     015676
SB5A    015670
SB5X    015706
SB5XAD  015710
SB6     015750
SB6X    015760
SB7     016020
SB7X    016030
SB7XAD  016032
SCOPE =  000240
SC3     026000
SC4     026014
SECPRT  027400
SETBR   025146
SETCC   025166
SETCD   025776
SETUP   025130
SET2BR  025236
SHL     001502
SHLE    001516
SHR     001616
SHRE    001632
SKPMSG  060316
SKP104  037672
SKTST2  040526
SNMB0A  006272
SNMB1A  006376
SNMB1B  006406
SNMB1C  006430
SNMB2A  006552
SNMB2B  006562
SNMB2C  006576
SNMB2D  006616
SNMB2E  006626
SNMB3A  006770
SNMB3B  007000
SNMB3C  007016
SNMB3D  007026
SNM0A   006232
SNM1A   006334
SNM2A   006472
SNM2B   006502
SNM3A   006702
SNM3B   006712
SNM4A   007076
SNM4B   007106
SNM5A   007160
SNM5B   007170
SNM6A   007244
SNM6B   007254
SNM7A   007326
SNM7B   007336
SOB1    022446
SOB2    022454
SOB3    022464
SOB4    022504
SOPA    006150
SOPB    006170
SOPB0A  003560
SOPB0B  003570
SOPB1A  003702
SOPB1B  003720
SOPB1C  003764
SOPB1D  004004
SOPB2A  004140
SOPB2B  004160
SOPB2C  004230
SOPB2D  004254
SOPB3A  004742
SOPB3B  004766
SOPB3C  005034
SOPB3D  005056
SOPX    006134
SOPXAD  006200
SOPZA   004052
SOP0A   003416
SOP0B   003436
SOP0C   003500
SOP0D   003524
SOP1A   003626
SOP1B   003640
SOP2B   004072
SOP3A   004656
SOP3B   004672
SOP4A   005270
SOP4B   005310
SOP5A   005366
SOP5B   005402
SOP6A   005446
SOP6B   005462
SOP7A   005530
SOP7B   005544
START   001002
STATUS= 177776
STBOT   001000
STP     036456
STPP    031322
STPPA   031332
STP3    037314
STP3D   037324
STP4    041064
STP4E   041074
SUB0    007472
SUB0A   007502
SUB1    021060
SUB2    021070
SUB3    021112
SUB4    021122
SUB5    021140
SUB6    021150
SUB7    021170
SWB1    020122
SWB2    020132
SWB3    020150
SWR     042212
SWREG   000176
SW09  = 001000
SW10  = 002000
SW11  = 004000
SW12  = 010000
SXT0    022244
SXT1    022254
```

```
SXT2    022304          TRAP10  041672          TST232  052274          TST47   044100          TS143   012226
S0      042162          TRAP24  041662          TST233  052376          TST50   044130          TS144   012336
S1      042164          TRCSR = 177560          TST234  052474          TST51   044206          TS145   012412
S10     042206          TRC1    037304          TST235  052572          TST52   044270          TS146   012506
S11     042210          TRPADR  037430          TST236  052670          TST53   044352          TS147   012642
S2      042166          TRT   = 000003          TST237  052772          TST54   044434          TS15    001724
S3      042170          TR0     040140          TST240  053070          TST55   044514          TS150   013020
S4      042172          TR2     040160          TST241  053166          TST56   044574          TS151   013114
S5      042174          TR3     040306          TST242  053304          TST57   044654          TS152   013250
S6      042176          TR4     040320          TST243  053402          TST60   044736          TS153   013412
S7      042200          TR5     040310          TST244  053500          TST61   045020          TS154   013562
S8      042202          TST160  046072          TST245  053576          TST62   045100          TS155   013730
S9      042204          TST161  046120          TST246  053672          TS1     001056          TS156   014014
TAB   =%000003          TST162  046136          TST247  053766          TS10    001456          TS157   014100
TABLE   042074          TST163  046154          TST250  054062          TS100   006440          TS16    001756
TBL1    014012          TST164  046216          TST251  054160          TS101   006520          TS160   014164
TBL2    014076          TST165  046254          TST252  054256          TS102   006644          TS161   014250
TDEC1   035536          TST166  046300          TST253  054352          TS103   006730          TS162   014344
TDEC2   035562          TST167  046330          TST254  054446          TS104   007044          TS163   014506
TDEC3   035630          TST170  046366          TST255  054556          TS105   007122          TS164   014664
TDEC4   035676          TST171  046412          TST256  054666          TS106   007206          TS165   015026
TDEC6   035712          TST172  046440          TST257  055002          TS107   007270          TS166   015102
TDEC7   035722          TST173  046462          TST260  055112          TS11    001526          TS167   015166
TDEC77  040010          TST174  046520          TST261  055222          TS110   007352          TS17    002010
TDEC8   040000          TST175  046554          TST262  055336          TS111   007406          TS170   015234
TEMP1   042144          TST176  046632          TST263  055446          TS112   007442          TS171   015312
TEMP2   042146          TST177  046730          TST264  055556          TS113   007516          TS172   015360
TEMP3   042150          TST200  047026          TST265  055666          TS114   007660          TS173   015422
TEMP4   042152          TST201  047130          TST266  056002          TS115   007750          TS174   015502
TEMP5   042154          TST202  047226          TST267  056112          TS116   010074          TS175   015544
TEMP6   042156          TST203  047324          TST270  056222          TS117   010160          TS176   015624
TENSAV  041674          TST204  047426          TST271  056336          TS12    001572          TS177   015712
TESTN1  027430          TST205  047524          TST272  056446          TS120   010216          TS2     001146
TEST1   020034          TST206  047652          TST273  056556          TS121   010254          TS20    002054
TEST2   020044          TST207  047756          TST274  056650          TS122   010312          TS200   015762
TEST3   020062          TST210  050062          TST275  056752          TS123   010372          TS201   016034
THRPRT  042242          TST211  050166          TST276  057062          TS124   010432          TS202   016400
TKB   = 177562          TST212  050270          TST277  057172          TS125   010504          TS203   017056
TKS   = 177560          TST213  050372          TST300  057302          TS126   010560          TS204   017132
TONT1   037230          TST214  050474          TST301  057410          TS127   010636          TS205   017216
TO10    027310          TST215  050600          TST302  057516          TS13    001642          TS206   017306
TO114   027350          TST216  050704          TST303  057624          TS130   010700          TS207   017376
TO14    027320          TST217  051006          TST304  057734          TS131   010742          TS21    002124
TO244   027360          TST220  051110          TST305  060044          TS132   011004          TS210   017470
TO250   027370          TST221  051206          TST306  060152          TS133   011062          TS211   017606
TO30    027330          TST222  051304          TST37   043602          TS134   011140          TS212   017750
TO34    027340          TST223  051402          TST40   043636          TS135   011236          TS213   020006
TO4     027300          TST224  051504          TST41   043652          TS136   011400          TS214   020072
TPB   = 177566          TST225  051602          TST42   043670          TS137   011540          TS215   020160
TPS   = 177564          TST226  051700          TST43   043724          TS14    001672          TS216   020344
TRACE   037210          TST227  052002          TST44   043756          TS140   011646          TS217   020462
TRAPA = 000077          TST230  052100          TST45   044010          TS141   012010          TS22    002200
TRAPB   037620          TST231  052176          TST46   044042          TS142   012120          TS220   020606
```

```
TS221   020766
TS222   021026
TS223   021200
TS224   021346
TS225   021514
TS226   021656
TS227   022026
TS23    002260
TS230   022206
TS231   022314
TS232   022426
TS233   022514
TS234   022676
TS235   022766
TS236   023050
TS237   023140
TS24    002324
TS240   023224
TS241   023306
TS242   023376
TS243   023466
TS244   023560
TS245   023662
TS246   023770
TS247   024076
TS25    002374
TS250   024204
TS251   024314
TS252   024424
TS253   024464
TS254   024542
TS255   024706
TS256   025012
TS257   025120
TS26    002440
TS260   025360
TS261   025530
TS262   025604
TS263   025644
TS264   026062
TS265   026142
TS266   026222
TS267   026352
TS27    002510
TS270   026502
TS271   026536
TS272   026576
TS273   026644
TS274   026706
TS275   026746
TS276   027014
TS277   027074
TS3     001202
TS30    002554
TS300   027430
TS301   027726
TS302   030234
TS303   030444
TS304   030612
TS305   030646
TS306   030710
TS307   030754
TS31    002624
TS310   031072
TS311   031344
TS312   031412
TS313   031454
TS314   031520
TS315   031634
TS316   032074
TS317   032164
TS32    002670
TS320   032220
TS321   032262
TS322   032326
TS323   032444
TS324   032716
TS325   032752
TS326   033014
TS327   033060
TS33    002740
TS330   033176
TS331   033432
TS332   033522
TS333   033556
TS334   033620
TS335   033664
TS336   034002
TS337   034254
TS34    003000
TS340   034310
TS341   034352
TS342   034416
TS343   034534
TS344   034770
TS345   035024
TS346   035066
TS347   035132
TS35    003036
TS350   035250
TS351   035502
TS352   035536
TS353   035600
TS354   035722
TS355   035772
TS356   036042
TS357   036112
TS36    003074
TS360   036162
TS361   036232
TS362   036320
TS363   036370
TS364   036470
TS365   036536
TS366   036622
TS367   036676
TS37    003132
TS370   036750
TS371   037050
TS372   037134
TS373   037244
TS374   037332
TS375   037672
TS376   040026
TS377   040172
TS4     001240
TS40    003202
TS400   040356
TS401   040440
TS402   040526
TS403   040710
TS404   041074
TS405   041152
TS406   041220
TS407   041274
TS41    003252
TS410   041444
TS411   041532
TS412   041616
TS42    003322
TS43    003372
TS44    003452
TS45    003534
TS46    003600
TS47    003650
TS5     001276
TS50    003730
TS51    004014
TS52    004102
TS53    004170
TS54    004264
TS55    004376
TS56    004516
TS57    004622
TS6     001334
TS60    004702
TS61    004776
TS62    005066
TS63    005236
TS64    005320
TS65    005412
TS66    005472
TS67    005554
TS7     001400
TS70    005644
TS71    005746
TS72    006032
TS73    006122
TS74    006202
TS75    006242
TS76    006302
TS77    006344
TTCSR = 177564
TTT37   040704
TTYOUT  042216
TTY11   040674
TTY3    040614
TTY4    040666
TYPCNT  042160
TYPDS = 104405
TYPE  = 104401
TYPOC = 104402
TYPON = 104404
TYPOS = 104403
USP1    024510
USP1A   024534
USP2    024610
USP3    024634
USP4    024670
USRM  = 140000
USTBOT  027300
VDEC1   035764
VDEC10  036214
VDEC11  036264
VDEC12  036274
VDEC13  036352
VDEC14  036362
VDEC2   035754
VDEC3   036034
VDEC4   036024
VDEC5   036104
VDEC6   036074
VDEC7   036154
VDEC8   036144
VDEC9   036224
WATE    041036
WATE1   040762
WATE2   040776
WATE3   041024
XOR1    022360
XOR2    022370
XOR3    022416
XXT     041526
YBR     025272
YNTAB   027226
$APTHD  000330
$ATYC   061340
$ATY1   061314
$ATY3   061322
$ATY4   061332
$CHARC  061300
$CPUOP  000326
$CRLF   042224
$DBLK   061776
$DEVCT  000310
$DOAGN  060502
$DTBL   061766
$ENDAD  060472
$ENDCT  060440
$ENDMG  060511
$ENULL  060506
$ENV    000320
$ENVM   000321
$EOP    060414
$EOPCT  060432
$ERN  = 001162
$ERROR= 000302
$ETABL  000320
$ETEND  000330
$FATAL  000302
$FFLG   061560
$FILLC  061306
$FILLS  061305
$GET42  060462
$HIBTS  000330
$HLT    060526
$LF     061311
$LFLG   061557
$MAIL   000300
$MBADR  000332
$MFLG   061556
$MSGAD  000314
$MSGLG  000316
$MSGTY  000300
$NULL   061304
$OCNT   062230
$OMODE  062232
$PASS   000306
$PASTM  000336
$QUES   061310
$RTNAD  060504
$SETUP= 000020
$STUP = 177777
$SVPC = 000400
$SWR  = 000000
```

```
$SWREG  000322          $TPS    042222          $TSTNM= 000304          $TYPON  062046          $XX    = 177666
$TESTN  000304          $TRAP   062234          $TYPDS  061562          $TYPOS  062006          $XXX   = 000665
$TN    = 000413         $TRAP2  062256          $TYPE   061022          $UNIT   000312          $$GET4= 000000
$TPB    042220          $TRP   = 000006         $TYPEC  061234          $UNITM  000340          $OFILL  062231
$TPCNT  042161          $TRPAD  062270          $TYPEX  061302          $USWR   000324          .      = 062306
$TPFLG  061307          $TSTM   000334          $TYPOC  062032          $X     - 041626         .$X    - 000330


. ABS.  062306      000


 ERRORS DETECTED:  0

 MULE:CJKDBB,MULE:CJKDBB.SEQ/SOL/NL:TOC-CJKDBB.P11
 RUN-TIME: 145 181 4 SECONDS
 RUN-TIME RATIO: 712/331-2.1
 CORE USED:  27K  (53 PAGES)
```