PDP11/23

F-11 CPU DIAG
CJKDBA0

AH-F140A-MC

FICHE 1 OF 2

MAR 1979
digital
MADE IN USA

PDP11/23

F-11 CPU DIAG

CJKDBA0

AH-F140A-MC

FICHE 2 OF 2

MAR 1979

digital

MADE IN USA

```
000000                          .REPT 0
```

IDENTIFICATION

PRODUCT CODE: AC-F141A-MC

PRODUCT NAME: CJKDBA0 F11 CPU DIAG

DATE: 24-JAN-79

MAINTAINER: DIAGNOSTIC ENGINEERING

CONTENTS

1.0 GENERAL INFORMATION.
1.1 HISTORY.
1.2 PROGRAM DESCRIPTION.
1.3 ABSTRACTS OF PART ONE, TWO AND THREE.

2.0 HARDWARE REQUIREMENT.

3.0 RELATED DOCUMENTS AND STANDARDS.

4.0 STARTING PROCEDURES.

5.0 TRAPCATCHER ABSTRACTS.

6.0 ERROR HANDLING.
6.1 ERROR HANDLING IN PART ONE AND TWO.
6.2 ERROR HANDLING IN PART THREE.

7.0 SWITCH SETTING (APPLICABLE ONLY TO PART THREE).

8.0 EXECUTION TIMES.

9.0 ROUTINES ABSTRACT.
9.1 HALT ROUTINE.
9.2 POWER FAIL ROUTINE.

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 KDF11-A PROCESSOR WITH 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 APT 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  MAINDEC-11-CJKDBA-A  F-11 CPU DIAG.
                                        .ENABLE ABS
                                        .NLIST  CND,MC,MD
                                        .LIST   ME
                        000240          SCOPE=NOP
                        000007          R7=%7
                        000006          R6=%6
                        177776          PS=177776
                        177564          TPS=177564
                        177566          TPB=177566
                        140000          USRM=140000
                        030000          PUSRM=30000
                                .SBTTL  ACT11 HOOKS

                                ;*******************************************************************************
                                ;HOOKS REQUIRED BY ACT11
                        000400          $SVPC=.                 ;SAVE PC
                        000046          .=46
000046  060606                          $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.

000330                          $APTHD:
000330  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000332  000300                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000334  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
000336  000020                  $PASTM: .WORD   20      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000340  000005                  $UNITM: .WORD   5       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000342  000014                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
                                ;*****************************************************************************************
                                ;SOME POINTERS TO CPU TRAP HANDLERS
                                ;*****************************************************************************************
000004                                  .=4
000004  027352                          TO4
000006  000000                          0
000010  027362                          TO10
000012  000000                          0
000014  027372                          TO14
000030                                  .=30
000030  027402                          TO30
000032  000000                          0
000034  027412                          TO34
000036  000000                          0
000114                                  .=114
000114  027422                          TO114
000116  000000                          0
000244                                  .=244
000244  027432                          TO244
000246  000000                          0
000250  027442                          TO250
000252  000000                          0

000172                          .=172
000172  000000                  LPADR:          0       ;LOOP ADDRESS (EIS TEST)
000174  000000                  DISPREG:        0       ;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:          0       ;SOFTWARE SWITCH REGISTER

                                ;*****************************************************************************************
                                ;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
                                ;*****************************************************************************************
000370                                  .=370
000370  000000  000000  000000         0,0,0,0,0,0
000376  000000  000000  000000
000404  000001  000001  177777         1,1,-1
                                ;*****************************************************************************************
```

```
                                                    ;SET UP STARTING ADDRESS
                    000302                          $ERROR=$FATAL
                    000304                          $TSTNM=$TESTN
                    001000                          .=1000
001000  000000                                      STBOT:  .WORD 0                         ;STACK POINTER

                    000200                                  .=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

                    001002                          .=1002
                                                    .SBTTL **STARTING OF CPU TEST **
001002  012737  061032  000024                      START:  MOV     #PWRDN,@#24             ;SET UP FOR POWER FAIL
001010  012737  000000  000306                              MOV     #0,@#$PASS      ;CLEAR PASS COUNT
001016  012737  000016  062420                              MOV     #16,@#PASSPT    ;SET PRINT COUNTER
001024  012737  001046  000004                      RESTRT: MOV     #1$,@#4         ;SET UP FOR TIMEOUT IF NO MULTI TESTER
001032  012737  000340  000006                              MOV     #340,@#6
001040  012737  000002  164000                              MOV     #2,@#164000     ;SET BIT1 FOR MULTI TESTER
001046  012737  000006  000004                      1$:     MOV     #6,@#4          ;SET TRAP CATCHER
001054  005037  000006                                      CLR     @#6             ;SET HALT
001060  012706  001000                                      MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
001064  012702  000304                                      MOV     #$TESTN,R2      ;SET UP POINTER TO MESSAGE TYPE
001070  012737  000000  000304                              MOV     #0,@#$TSTNM     ;CLEAR TEST NUMBER
001076  012737  000000  000302                              MOV     #0,@#$ERROR     ;CLEAR ERROR NUMBER
001104  012737  000000  000300                              MOV     #0,@#$MSGTY     ;CLEAR MESSAGE TYPE(FOR APT)
                                                    ;*****************************************************************************
                                                    ;TEST 1 CHECK BRANCHES ON Z BIT
                                                    ;*****************************************************************************
001112  005212                                      TS1:    INC     (R2)            ;UPDATE TEST NUMBER
001114  022712  000001                                      CMP     #1,(R2)         ;SEQUENCE ERROR?
001120  001024                                              BNE     TS2-10  ;BR TO ERROR HALT ON SEQ ERROR
001122  000257                                              CCC                     ;CLEAR ALL CONDITION CODES
001124  001401                                              BEQ     BRA1            ;SHOULD BRANCH
001126  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               <====
001130                                          BRA1:
001130  012742  000001                                      MOV     #1,-(R2)        ;MOVE TO MAILBOX #  *******  1  *******
001134  005242                                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001136  000000                                              HALT                    ;SHOULD HAVE BRANCHED: Z=0
001140                                          BRA2:
001140  001004                                              BNE     BRA3
                                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
                                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                                    ;           WHICH FOLLOWS W/ 767           <====
001142  012742  000002                                      MOV     #2,-(R2)        ;MOVE TO MAILBOX #  *******  2  *******
001146  005242                                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001150  000000                                              HALT                    ;
001152  000264                                  BRA3:       SEZ
001154  001001                                              BNE     BRA4
```

```
001156 000404                       BR      BRA5
                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS     <====
                                                    ;           BRANCH INSTRUCTION AND           <====
                                                    ;           REPLACE THE MOVE INSTRUCTION     <====
                                                    ;           FOLLOWING W/ 760                 <====
001160                      BRA4:
001160 012742 000003                MOV     #3,-(R2)        ;MOVE TO MAILBOX #  *******  3  *******
001164 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001166 000000                       HALT                    ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
001170                      BRA5:
001170 001404                       BEQ     TS2
                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS     <====
                                                    ;           CONDITIONAL BRANCH INST. AND     <====
                                                    ;           REPLACE THE MOVE INSTRUCTION     <====
                                                    ;           WHICH FOLLOWS W/ 753             <====
001172 012742 000004                MOV     #4,-(R2)        ;MOVE TO MAILBOX #  *******  4  *******
001176 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001200 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
                            ;*******************************************************************************
001202 005212               TS2:    INC     (R2)            ;UPDATE TEST NUMBER
001204 022712 000002                CMP     #2,(R2)         ;SEQUENCE ERROR?
001210 001006                       BNE     TS3-10  ;BR TO ERROR HALT ON SEQ ERROR
001212 012737 000000 000000         MOV     #0,@#0          ;MOVE ZEROES THRU ADDRESS LINES, DATA
                                                            ;LINES AND INTERNAL PATHS
001220 005737 000000                TST     @#0             ;SUCCESSFUL?
001224 001404                       BEQ     TS3
                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS     <====
                                                    ;           CONDITIONAL BRANCH INST. AND     <====
                                                    ;           REPLACE THE MOVE INSTRUCTION     <====
                                                    ;           WHICH FOLLOWS W/ 771             <====
001226 012742 000005                MOV     #5,-(R2)        ;MOVE TO MAILBOX #  *******  5  *******
001232 005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001234 000000                       HALT                    ;DATA INCORRECT
                                                            ;  OR SEQUENCE ERROR

                            ;*******************************************************************************
                            ;TEST 3 TEST OF PATTERN 125252 IN DATA PATH
                            ;*******************************************************************************
001236 005212               TS3:    INC     (R2)            ;UPDATE TEST NUMBER
001240 022712 000003                CMP     #3,(R2)         ;SEQUENCE ERROR?
001244 001007                       BNE     TS4-10  ;BR TO ERROR HALT ON SEQ ERROR
```

```
001246  012737  125252  000000          MOV     #125252,@#0       ;MOVE ALTERNATING ONES AND ZEROES
                                                                  ;THRU DATA PATHS
001254  022737  125252  000000          CMP     #125252,@#0       ;SUCCESSFUL
001262  001404                          BEQ     TS4
                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                  ;           WHICH FOLLOWS W/ 770            <====
001264  012742  000006                  MOV     #6,-(R2)          ;MOVE TO MAILBOX #  *******  6  *******
001270  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
001272  000000                          HALT                      ;DATA INCORRECT
                                                                  ;  OR SEQUENCE ERROR


                                ;*******************************************************************
                                ;TEST 4 TEST OF PATTERN 052525 IN DATA PATH
                                ;*******************************************************************
001274  005212                  TS4:    INC     (R2)              ;UPDATE TEST NUMBER
001276  022712  000004                  CMP     #4,(R2)           ;SEQUENCE ERROR?
001302  001007                          BNE     TS5-10  ;BR TO ERROR HALT ON SEQ ERROR
001304  012737  052525  000000          MOV     #052525,@#0       ;MOVE ALTERNATING ZEROES AND ONES
                                                                  ;THRU DATA PATH
001312  022737  052525  000000          CMP     #052525,@#0       ;SUCCESSFUL?
001320  001404                          BEQ     TS5
                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                  ;           WHICH FOLLOWS W/ 770            <====
001322  012742  000007                  MOV     #7,-(R2)          ;MOVE TO MAILBOX #  *******  7  *******
001326  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
001330  000000                          HALT                      ;DATA INCORRECT
                                                                  ;  OR SEQUENCE ERROR


                                ;*******************************************************************
                                ;TEST 5 TEST OF ALL ONES IN DATA PATH
                                ;*******************************************************************
001332  005212                  TS5:    INC     (R2)              ;UPDATE TEST NUMBER
001334  022712  000005                  CMP     #5,(R2)           ;SEQUENCE ERROR?
001340  001007                          BNE     TS6-10  ;BR TO ERROR HALT ON SEQ ERROR
001342  012737  177777  000000          MOV     #177777,@#0       ;MOVE ONES THRU DATA PATH
001350  022737  177777  000000          CMP     #177777,@#0       ;SUCCESSFUL
001356  001404                          BEQ     TS6
                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                  ;           WHICH FOLLOWS W/ 770            <====
001360  012742  000010                  MOV     #10,-(R2)         ;MOVE TO MAILBOX #  *******  10  *******
001364  005242                          INC     -(R2)             ;SET MSGTYP TO FATAL ERROR
001366  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
                                ;*********************************************************************************
001370  005212                TS6:    INC     (R2)            ;UPDATE TEST NUMBER
001372  022712  000006              CMP     #6,(R2)         ;SEQUENCE ERROR?
001376  001012                      BNE     TS7-10  ;BR TO ERROR HALT ON SEQ ERROR
001400  000241                      CLC                     ;CLEAR CARRY BIT
001402  012737  000001  000000      MOV     #1,@#0          ;LOAD A 1
001410  006137  000000              ROL     @#0             ;SHIFT LEFT
001414  022737  000002  000000      CMP     #2,@#0          ;SUCCESSFUL
001422  001404                      BEQ     TS7
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
001424  012742  000011              MOV     #11,-(R2)       ;MOVE TO MAILBOX #  *******  11  *******
001430  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001432  000000                      HALT                    ;BIT 1 NOT SET
                                                        ;  OR SEQUENCE ERROR

                                ;*********************************************************************************
                                ;TEST 7 SHIFT CARRY INTO BIT 0
                                ;*********************************************************************************
001434  005212                TS7:    INC     (R2)            ;UPDATE TEST NUMBER
001436  022712  000007              CMP     #7,(R2)         ;SEQUENCE ERROR?
001442  001017                      BNE     TS10-10         ;BR TO ERROR HALT ON SEQ ERROR
001444  012737  000000  000000      MOV     #0,@#0          ;CLEAR LOCATION
001452  000261                      SEC                     ;SET CARRY
001454  006137  000000              ROL     @#0             ;ROTATE CARRY BIT TO BIT 0
001460  103014                      BCC     TS10
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
001462  012742  000012              MOV     #12,-(R2)       ;MOVE TO MAILBOX #  *******  12  *******
001466  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001470  000000                      HALT                    ;CARRY CLEAR
                                                        ;  OR SEQUENCE ERROR
001472  022737  000001  000000      CMP     #1,@#0          ;BIT 0 SET
001500  001404                      BEQ     TS10
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
001502  012742  000013              MOV     #13,-(R2)       ;MOVE TO MAILBOX #  *******  13  *******
001506  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001510  000000                      HALT                    ;BIT 0 NOT SET
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;TEST 10         LEFT SHIFT FROM BIT 0 TO C-BIT
                                ;*****************************************************************************************
001512  005212                  TS10:   INC     (R2)            ;UPDATE TEST NUMBER
001514  022712  000010                  CMP     #10,(R2)        ;SEQUENCE ERROR?
001520  001014                          BNE     TS11-10         ;BR TO ERROR HALT ON SEQ ERROR
001522  012737  000001  000000          MOV     #1,@#0          ;SET BIT 0
001530  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
001534  000241                          CLC                     ;CLEAR C-BIT
001536  005200                  SHL:    INC     R0              ;INCREMENT BIT COUNTER
001540  001404                          BEQ     SHLE            ;BR TO ERROR HALT IF BIT IS LOST
001542  006137  000000                  ROL     @#0             ;SHIFT LEFT ONE POSITION
001546  103373                          BCC     SHL             ;BRANCH IF C-BIT NOT SET
001550  001404                          BEQ     TS11
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 763           <====
001552                          SHLE:
001552  012742  000014                  MOV     #14,-(R2)       ;MOVE TO MAILBOX #  *******  14  *******
001556  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001560  000000                          HALT                    ;LEFT SHIFTING LOGIC FAILED
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************************
                                ;TEST 11         SHIFT BIT 15 TO BIT 14
                                ;*****************************************************************************************
001562  005212                  TS11:   INC     (R2)            ;UPDATE TEST NUMBER
001564  022712  000011                  CMP     #11,(R2)        ;SEQUENCE ERROR?
001570  001012                          BNE     TS12-10         ;BR TO ERROR HALT ON SEQ ERROR
001572  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
001600  000241                          CLC                     ;CLEAR CARRY
001602  006037  000000                  ROR     @#0             ;SHIFT BIT 15 TO BIT 14
001606  022737  040000  000000          CMP     #40000,@#0      ;SUCCESSFUL
001614  001404                          BEQ     TS12
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <====
001616  012742  000015                  MOV     #15,-(R2)       ;MOVE TO MAILBOX #  *******  15  *******
001622  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001624  000000                          HALT                    ;BIT 14 NOT SET
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************************
                                ;TEST 12         RIGHT SHIFT FROM BIT 15 TO C-BIT
                                ;*****************************************************************************************
001626  005212                  TS12:   INC     (R2)            ;UPDATE TEST NUMBER
001630  022712  000012                  CMP     #12,(R2)        ;SEQUENCE ERROR?
001634  001014                          BNE     TS13-10         ;BR TO ERROR HALT ON SEQ ERROR
001636  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
001644  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
001650  000241                          CLC                     ;CLEAR C-BIT
001652  005200                  SHR:    INC     R0              ;INCREMENT BIT COUNTER
001654  001404                          BEQ     SHRE            ;BR TO ERROR HALT IF BIT IS LOST
```

```
                        ROR     @#0             ;ROTATE RIGHT ONE POSITION
                        BCC     SHR             ;BRANCH IF C-BIT CLEAR
                        BEQ     TS13
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 763           <====
SHRE:
                        MOV     #16,-(R2)       ;MOVE TO MAILBOX #  *******  16  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        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
;******************************************************************************
TS13:   INC     (R2)            ;UPDATE TEST NUMBER
        CMP     #13,(R2)        ;SEQUENCE ERROR?
        BNE     TS14-10         ;BR TO ERROR HALT ON SEQ ERROR

        MOV     #0,R0           ;MOVE ZEROES TO R0
        TST     R0              ;SUCCESSFUL?
        BEQ     TS14
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 773           <====
        MOV     #17,-(R2)       ;MOVE TO MAILBOX #  *******  17  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;R0 NOT 0
                                ;  OR SEQUENCE ERROR

;******************************************************************************
;TEST 14        TEST IF R0 CAN HOLD ONES AND ZEROES
;******************************************************************************
TS14:   INC     (R2)            ;UPDATE TEST NUMBER
```

```
001730  022712  000014          CMP     #14,(R2)        ;SEQUENCE ERROR?
001734  001005                  BNE     TS15-10         ;BR TO ERROR HALT ON SEQ ERROR
001736  012700  125252          MOV     #125252,R0      ;MOVE ALTERNATING ONES AND ZEROES TO R0
001742  020027  125252          CMP     R0,#125252      ;SUCCESSFUL?
001746  001404                  BEQ     TS15
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
001750  012742  000020          MOV     #20,-(R2)       ;MOVE TO MAILBOX #  *******  20  *******
001754  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001756  000000                  HALT                    ;R0 NOT 125252
                                                        ;  OR SEQUENCE ERROR


                        ;*******************************************************************************
                        ;TEST 15        TEST IF R0 CAN HOLD ZEROES AND ONES
                        ;*******************************************************************************
001760  005212          TS15:   INC     (R2)            ;UPDATE TEST NUMBER
001762  022712  000015          CMP     #15,(R2)        ;SEQUENCE ERROR?
001766  001005                  BNE     TS16-10         ;BR TO ERROR HALT ON SEQ ERROR
001770  012700  052525          MOV     #052525,R0      ;MOVE ALTERNATING ZEROES AND ONES TO R0
001774  020027  052525          CMP     R0,#052525      ;SUCCESSFUL?
002000  001404                  BEQ     TS16
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
002002  012742  000021          MOV     #21,-(R2)       ;MOVE TO MAILBOX #  *******  21  *******
002006  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002010  000000                  HALT                    ;R0 NOT 52525
                                                        ;  OR SEQUENCE ERROR


                        ;*******************************************************************************
                        ;TEST 16        TEST IF R0 CAN HOLD ALL ONES
                        ;*******************************************************************************
002012  005212          TS16:   INC     (R2)            ;UPDATE TEST NUMBER
002014  022712  000016          CMP     #16,(R2)        ;SEQUENCE ERROR?
002020  001005                  BNE     TS17-10         ;BR TO ERROR HALT ON SEQ ERROR
002022  012700  177777          MOV     #177777,R0      ;MOVE ALL ONES TO R0
002026  020027  177777          CMP     R0,#177777      ;SUCCESSFUL?
002032  001404                  BEQ     TS17
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
002034  012742  000022          MOV     #22,-(R2)       ;MOVE TO MAILBOX #  *******  22  *******
002040  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002042  000000                  HALT                    ;R0 NOT 177777
                                                        ;  OR SEQUENCE ERROR


                        ;*******************************************************************************
                        ;TEST 17        TEST IF R1 CAN HOLD A ONE IN ALL BITS
                        ;*******************************************************************************
002044  005212          TS17:   INC     (R2)            ;UPDATE TEST NUMBER
002046  022712  000017          CMP     #17,(R2)        ;SEQUENCE ERROR?
002052  001012                  BNE     TS20-10         ;BR TO ERROR HALT ON SEQ ERROR
```

```
                                        MOV     #1,R1           ;SET BIT 0
                                        MOV     #-21,R0         ;SET BIT COUNTER
                                        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               <====
002214  012702  000304          MOV     #$TESTN,R2      ;RESTORE POINTER
002220  012742  000025          MOV     #25,-(R2)       ;MOVE TO MAILBOX #  *******  25  *******
002224  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002226  000000                  HALT                    ;FAILURE WITH R2
002230  012702  000304  REG2A:  MOV     #$TESTN,R2              ;RESTORE POINTER

                        ;*******************************************************************************************
                        ;TEST 22        TEST IF R2 CAN HOLD A ZERO IN ALL BITS
                        ;*******************************************************************************************
002234  005212          TS22:   INC     (R2)            ;UPDATE TEST NUMBER
002236  022712  000022          CMP     #22,(R2)        ;SEQUENCE ERROR?
002242  001020                  BNE     TS23-10         ;BR TO ERROR HALT ON SEQ ERROR
002244  012702  177776          MOV     #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
002250  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002254  000261                  SEC                     ;SET C-BIT
002256  005200  REG2B:  INC     R0              ;INCREMENT BIT COUNTER
002260  001407                  BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
002262  006102                  ROL     R2              ;ROTATE 1 POSITION
002264  103774                  BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
002266  022702  177777          CMP     #-1,R2          ;CHECK DATA IN R2
002272  001406                  BEQ     REG2C
002274  012702  000304          MOV     #$TESTN,R2      ;RESTORE POINTER
002300          R2ERR:
002300  012742  000026          MOV     #26,-(R2)       ;MOVE TO MAILBOX #  *******  26  *******
002304  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002306  000000                  HALT                    ;FAILURE WITH R2
002310  012702  000304  REG2C:  MOV     #$TESTN,R2      ;RESTORE POINTER

                        ;*******************************************************************************************
                        ;TEST 23        TEST IF R3 CAN HOLD A ONE IN ALL BITS
                        ;*******************************************************************************************
002314  005212          TS23:   INC     (R2)            ;UPDATE TEST NUMBER
002316  022712  000023          CMP     #23,(R2)        ;SEQUENCE ERROR?
002322  001012                  BNE     TS24-10         ;BR TO ERROR HALT ON SEQ ERROR
002324  012703  000001          MOV     #1,R3           ;SET BIT 0
002330  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002334  000241                  CLC                     ;CLEAR C-BIT
002336  005200  REG3:   INC     R0              ;INCREMENT BIT COUNTER
002340  001403                  BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
002342  006103                  ROL     R3              ;ROTATE 1 POSITION
002344  103374                  BCC     REG3            ;ALL DONE
002346  001404                  BEQ     TS24

                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 765           <====
002350          REG3E:
002350  012742  000027          MOV     #27,-(R2)       ;MOVE TO MAILBOX #  *******  27  *******
002354  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002356  000000                  HALT                    ;FAILURE WITH R3
                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;TEST 24         TEST IF R3 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
002360  005212          TS24:   INC     (R2)            ;UPDATE TEST NUMBER
002362  022712  000024          CMP     #24,(R2)        ;SEQUENCE ERROR?
002366  001014                  BNE     TS25-10         ;BR TO ERROR HALT ON SEQ ERROR
002370  012703  177776          MOV     #-2,R3          ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
002374  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002400  000261                  SEC                     ;SET C-BIT
002402  005200          REG3A:  INC     R0              ;INCREMENT BIT COUNTER
002404  001405                  BEQ     R3ERR           ;BR TO ERROR HALT IF COUNTER=0
002406  006103                  ROL     R3              ;ROTATE 1 POSITION
002410  103774                  BCS     REG3A           ;CONTINUE UNTIL C-BIT IS CLEAR
002412  022703  177777          CMP     #-1,R3          ;CHECK DATA
002416  001404                  BEQ     TS25
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
002420          R3ERR:
002420  012742  000030          MOV     #30,-(R2)       ;MOVE TO MAILBOX #  *******  30  *******
002424  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002426  000000                  HALT                    ;FAILURE WITH R3
                                                        ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 25         TEST IF R4 CAN HOLD A ONE IN ALL BITS
                                ;*****************************************************************************
002430  005212          TS25:   INC     (R2)            ;UPDATE TEST NUMBER
002432  022712  000025          CMP     #25,(R2)        ;SEQUENCE ERROR?
002436  001012                  BNE     TS26-10         ;BR TO ERROR HALT ON SEQ ERROR
002440  012704  000001          MOV     #1,R4           ;SET BIT 0
002444  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002450  000241                  CLC                     ;CLEAR C-BIT
002452  005200          REG4:   INC     R0              ;INCREMENT BIT COUNTER
002454  001403                  BEQ     REG4E           ;BR TO ERROR HALT IF BIT IS LOST
002456  006104                  ROL     R4              ;ROTATE 1 POSITION
002460  103374                  BCC     REG4            ;ALL DONE
002462  001404                  BEQ     TS26
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
002464          REG4E:
002464  012742  000031          MOV     #31,-(R2)       ;MOVE TO MAILBOX #  *******  31  *******
002470  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002472  000000                  HALT                    ;FAILURE WITH R4
                                                        ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 26         TEST IF R4 CAN HOLD A ZERO IN ALL BITS
                                ;*****************************************************************************
002474  005212          TS26:   INC     (R2)            ;UPDATE TEST NUMBER
002476  022712  000026          CMP     #26,(R2)        ;SEQUENCE ERROR?
002502  001014                  BNE     TS27-10         ;BR TO ERROR HALT ON SEQ ERROR
002504  012704  177776          MOV     #-2,R4          ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
```

```
  975  002510  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
  976  002514  000261                  SEC                     ;SET C-BIT
  977  002516  005200          REG4A:  INC     R0              ;INCREMENT BIT COUNTER
  978  002520  001405                  BEQ     R4ERR           ;BR TO ERROR HALT IF COUNTER=0
  979  002522  006104                  ROL     R4              ;ROTATE 1 POSITION
  980  002524  103774                  BCS     REG4A           ;CONTINUE UNTIL C-BIT IS CLEAR
  981  002526  022704  177777          CMP     #-1,R4          ;CHECK DATA
  982  002532  001404                  BEQ     TS27
  983                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
  984                                                          ;           CONDITIONAL BRANCH INST. AND   <====
  985                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
  986                                                          ;           WHICH FOLLOWS W/ 763           <====
  987  002534                  R4ERR:
  988  002534  012742  000032          MOV     #32,-(R2)       ;MOVE TO MAILBOX #  *******  32  *******
  989  002540  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
  990  002542  000000                  HALT                    ;FAILURE WITH R4
  991                                                          ;  OR SEQUENCE ERROR
  992
  993
  994                          ;*****************************************************************************************
  995                          ;TEST 27        TEST IF R5 CAN HOLD A ONE IN ALL BITS
  996                          ;*****************************************************************************************
  997  002544  005212          TS27:   INC     (R2)            ;UPDATE TEST NUMBER
  998  002546  022712  000027          CMP     #27,(R2)        ;SEQUENCE ERROR?
  999  002552  001012                  BNE     TS30-10         ;BR TO ERROR HALT ON SEQ ERROR
 1000  002554  012705  000001          MOV     #1,R5           ;SET BIT 0
 1001  002560  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 1002  002564  000241                  CLC                     ;CLEAR C-BIT
 1003  002566  005200          REG5:   INC     R0              ;INCREMENT BIT COUNTER
 1004  002570  001403                  BEQ     REG5E           ;BR TO ERROR HALT IF BIT IS LOST
 1005  002572  006105                  ROL     R5              ;ROTATE 1 POSITION
 1006  002574  103374                  BCC     REG5            ;ALL DONE
 1007  002576  001404                  BEQ     TS30
 1008                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 1009                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 1010                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 1011                                                          ;           WHICH FOLLOWS W/ 765           <====
 1012  002600                  REG5E:
 1013  002600  012742  000033          MOV     #33,-(R2)       ;MOVE TO MAILBOX #  *******  33  *******
 1014  002604  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 1015  002606  000000                  HALT                    ;FAILURE WITH R5
 1016                                                          ;  OR SEQUENCE ERROR
 1017
 1018                          ;*****************************************************************************************
 1019                          ;TEST 30        TEST IF R5 CAN HOLD A ZERO IN ALL BITS
 1020                          ;*****************************************************************************************
 1021  002610  005212          TS30:   INC     (R2)            ;UPDATE TEST NUMBER
 1022  002612  022712  000030          CMP     #30,(R2)        ;SEQUENCE ERROR?
 1023  002616  001014                  BNE     TS31-10         ;BR TO ERROR HALT ON SEQ ERROR
 1024  002620  012705  177776          MOV     #-2,R5          ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
 1025  002624  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 1026  002630  000261                  SEC                     ;SET C-BIT
 1027  002632  005200          REG5A:  INC     R0              ;INCREMENT BIT COUNTER
 1028  002634  001405                  BEQ     R5ERR           ;BR TO ERROR HALT IF COUNTER=0
 1029  002636  006105                  ROL     R5              ;ROTATE 1 POSITION
 1030  002640  103774                  BCS     REG5A           ;CONTINUE UNTIL C-BIT IS C;EAR
```

```
1031  002642  022705  177777          CMP     #-1,R5          ;CHECK DATA
1032  002646  001404                  BEQ     TS31
1033                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1034                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1035                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1036                                                          ;           WHICH FOLLOWS W/ 763           <====
1037  002650                      R5ERR:
1038  002650  012742  000034            MOV     #34,-(R2)       ;MOVE TO MAILBOX #  *******  34  *******
1039  002654  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1040  002656  000000                    HALT                    ;FAILURE WITH R5
1041                                                          ;  OR SEQUENCE ERROR
1042
1043                              ;*****************************************************************************
1044                              ;TEST 31        TEST IF R6 CAN HOLD A ONE IN ALL BITS
1045                              ;*****************************************************************************
1046  002660  005212              TS31:   INC     (R2)            ;UPDATE TEST NUMBER
1047  002662  022712  000031            CMP     #31,(R2)        ;SEQUENCE ERROR?
1048  002666  001012                    BNE     TS32-10         ;BR TO ERROR HALT ON SEQ ERROR
1049  002670  012706  000001            MOV     #1,R6           ;SET BIT 0
1050  002674  012700  177757            MOV     #-21,R0         ;SET BIT COUNTER
1051  002700  000241                    CLC                     ;CLEAR C-BIT
1052  002702  005200              REG6:   INC     R0              ;INCREMENT BIT COUNTER
1053  002704  001403                    BEQ     REG6E           ;BR TO ERROR HALT IF BIT IS LOST
1054  002706  006106                    ROL     R6              ;ROTATE 1 POSITION
1055  002710  103374                    BCC     REG6            ;ALL DONE
1056  002712  001404                    BEQ     TS32
1057                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1058                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1059                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1060                                                          ;           WHICH FOLLOWS W/ 765           <====
1061  002714                      REG6E:
1062  002714  012742  000035            MOV     #35,-(R2)       ;MOVE TO MAILBOX #  *******  35  *******
1063  002720  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1064  002722  000000                    HALT                    ;FAILURE WITH R6
1065                                                          ;  OR SEQUENCE ERROR
1066
1067                              ;*****************************************************************************
1068                              ;TEST 32        TEST IF R6 CAN HOLD A ZERO IN ALL BITS
1069                              ;*****************************************************************************
1070  002724  005212              TS32:   INC     (R2)            ;UPDATE TEST NUMBER
1071  002726  022712  000032            CMP     #32,(R2)        ;SEQUENCE ERROR?
1072  002732  001014                    BNE     TS33-10         ;BR TO ERROR HALT ON SEQ ERROR
1073  002734  012706  177776            MOV     #-2,R6          ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
1074  002740  012700  177757            MOV     #-21,R0         ;SET BIT COUNTER
1075  002744  000261                    SEC                     ;SET C-BIT
1076  002746  005200              REG6A:  INC     R0              ;INCREMENT BIT COUNT
1077  002750  001405                    BEQ     R6ERR           ;BR TO ERROR HALT IF COUNTER=0
1078  002752  006106                    ROL     R6              ;ROTATE 1 POSITION
1079  002754  103774                    BCS     REG6A           ;CONTINUE UNTIL C-BIT IS CLEAR
1080  002756  022706  177777            CMP     #-1,R6          ;CHECK DATA
1081  002762  001404                    BEQ     TS33
1082                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1083                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1084                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1085                                                          ;           WHICH FOLLOWS W/ 763           <====
1086  002764                      R6ERR:
```

```
1087  002764  012742  000036          MOV     #36,-(R2)       ;MOVE TO MAILBOX #  *******  36  *******
1088  002770  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1089  002772  000000                  HALT                    ;FAILURE WITH R6
1090                                                          ;  OR SEQUENCE ERROR
1091
1092                            ;*************************************************************************************
1093                            .SBTTL  PSW TESTS
1094                            ;
1095                            ;       THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
1096                            ;PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE
1097                            ;PSW ADDRESSING LOGIC IS FUNCTIONING.  MOVE AND COMPARE INSTRUCTIONS
1098                            ;ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.
1099                            ;EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR
1100                            ;SCOPING.
1101                            ;       THE PSW REGISTER IS TESTED, THE CC INPUTS ARE TESTED
1102                            ;LATER IN THE MICROCODE TESTS.  SETTING OF THE T-BIT BY THE
1103                            ;TEST PATTERNS IS PURPOSELY AVOIDED. TESTING OF THE
1104                            ;T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.
1105                            ;
1106                            ;*************************************************************************************
1107                            ;TEST 33        TEST IF PSW WILL HOLD ZEROES
1108                            ;*************************************************************************************
1109  002774  005212            TS33:   INC     (R2)            ;UPDATE TEST NUMBER
1110  002776  022712  000033            CMP     #33,(R2)        ;SEQUENCE ERROR?
1111  003002  001010                    BNE     TS34-10         ;BR TO ERROR HALT ON SEQ ERROR
1112  003004  012706  001000            MOV     #STBOT,R6
1113  003010  012737  000000  177776    MOV     #0,@#PS         ;SET PSW TO ZERO
1114  003016  005737  177776            TST     @#PS            ;SUCCESSFUL
1115  003022  001404                    BEQ     TS34
1116                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1117                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1118                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1119                                                          ;           WHICH FOLLOWS W/ 767           <====
1120  003024  012742  000037          MOV     #37,-(R2)       ;MOVE TO MAILBOX #  *******  37  *******
1121  003030  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1122  003032  000000                  HALT                    ;PSW NOT 0
1123                                                          ;  OR SEQUENCE ERROR
1124
1125                            ;*************************************************************************************
1126                            ;TEST 34        TEST IF PSW WILL HOLD ONES AND ZEROES
1127                            ;*************************************************************************************
1128  003034  005212            TS34:   INC     (R2)            ;UPDATE TEST NUMBER
1129  003036  022712  000034            CMP     #34,(R2)        ;SEQUENCE ERROR?
1130  003042  001007                    BNE     TS35-10         ;BR TO ERROR HALT ON SEQ ERROR
1131  003044  012737  000252  177776    MOV     #252,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
1132  003052  023727  177776  000252    CMP     @#PS,#252       ;SUCCESSFUL?
1133  003060  001404                    BEQ     TS35
1134                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1135                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1136                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1137                                                          ;           WHICH FOLLOWS W/ 770           <====
1138  003062  012742  000040          MOV     #40,-(R2)       ;MOVE TO MAILBOX #  *******  40  *******
1139  003066  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1140  003070  000000                  HALT                    ;PSW NOT 252
1141                                                          ;  OR SEQUENCE ERROR
1142
```

```
1143
1144                                    ;*****************************************************************************
1145                                    ;TEST 35        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
1146                                    ;*****************************************************************************
1146  003072  005212                TS35:   INC     (R2)            ;UPDATE TEST NUMBER
1147  003074  022712  000035               CMP     #35,(R2)        ;SEQUENCE ERROR?
1148  003100  001007                       BNE     TS36-10         ;BR TO ERROR HALT ON SEQ ERROR
1149  003102  012737  000105  177776       MOV     #105,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
1150  003110  023727  177776  000105       CMP     @#PS,#105       ;SUCCESSFUL?
1151  003116  001404                       BEQ     TS36
1152                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1153                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1154                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1155                                                                ;           WHICH FOLLOWS W/ 770           <====
1156  003120  012742  000041               MOV     #41,-(R2)       ;MOVE TO MAILBOX #  *******  41  *******
1157  003124  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1158  003126  000000                       HALT                    ;PSW NOT 105
1159                                                                ;  OR SEQUENCE ERROR
1160
1161                                    ;*****************************************************************************
1162                                    ;TEST 36        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
1163                                    ;*****************************************************************************
1164  003130  005212                TS36:   INC     (R2)            ;UPDATE TEST NUMBER
1165  003132  022712  000036               CMP     #36,(R2)        ;SEQUENCE ERROR?
1166  003136  001007                       BNE     TS37-10         ;BR TO ERROR HALT ON SEQ ERROR
1167  003140  012737  000357  177776       MOV     #357,@#PS       ;MOVE ONES TO PSW
1168  003146  023727  177776  000357       CMP     @#PS,#357       ;SUCCESSFUL
1169  003154  001404                       BEQ     TS37
1170                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1171                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1172                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1173                                                                ;           WHICH FOLLOWS W/ 770           <====
1174  003156  012742  000042               MOV     #42,-(R2)       ;MOVE TO MAILBOX #  *******  42  *******
1175  003162  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1176  003164  000000                       HALT                    ;PSW NOT 357
1177                                                                ;  OR SEQUENCE ERROR
1178                                  .SBTTL  CONDITION CODE TEST
1179
1180                                    ;*****************************************************************************
1181                                    ;
1182                                    ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.
1183                                    ;THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1184                                    ;BEQ AND BNE ARE TESTED FOR PROPER EXECUTION.  THEN THE Z-BIT IS
1185                                    ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1186                                    ;AGAIN FOR PROPER OPERATION.
1187                                    ;       THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1188                                    ;CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1189                                    ;BRANCH ROM.  THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1190                                    ;LEAVING THE PC UNALTERED IS TESTED.  ONLY THOSE ROM ADDRESSES SPECIFICALLY
1191                                    ;USED IN THE TEST ARE VERIFIED HERE.
1192                                    ;
1193                                    ;*****************************************************************************
1194                                    ;TEST 37        TEST BRANCHES AROUND Z-BIT
1195                                    ;*****************************************************************************
1196  003166  005212                TS37:   INC     (R2)            ;UPDATE TEST NUMBER
1197  003170  022712  000037               CMP     #37,(R2)        ;SEQUENCE ERROR?
1198  003174  001014                       BNE     TS40-10         ;BR TO ERROR HALT ON SEQ ERROR
```

```
                                    ;FIRST WITH Z-BIT ON
003176  000257                      CCC                     ;CC=0100: JUST Z-BIT
003200  000264                      SEZ
003202  001001                      BNE     BRZ1            ;CHECK OPPOSITE CONDITION
003204  001404                      BEQ     BRZ2
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 773            <====
003206                      BRZ1:
003206  012742  000043              MOV     #43,-(R2)       ;MOVE TO MAILBOX #  *******  43  *******
003212  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003214  000000                      HALT                    ;IMPROPER BR W/ Z=1
                                    ;CHECK WITH Z-BIT OFF
003216  000277              BRZ2:   SCC                     ;CC=1011: ALL BUT Z-BIT
003220  000244                      CLZ
003222  001401                      BEQ     BRZ3
003224  001004                      BNE     TS40
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 763            <====
003226                      BRZ3:
003226  012742  000044              MOV     #44,-(R2)       ;MOVE TO MAILBOX #  *******  44  *******
003232  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003234  000000                      HALT                    ;IMPROPER BR W/ Z=0
                                                            ;  OR SEQUENCE ERROR

                            ;*******************************************************************************
                            ;
                            ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
                            ;THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
                            ;BMI AND BPL ARE TESTED FOR PROPER EXECUTION.  THEN THE N-BIT IS
                            ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
                            ;AGAIN FOR PROPER OPERATION.
                            ;       THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
                            ;CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
                            ;BRANCH ROM.  THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
                            ;LEAVING THE PC UNALTERED IS TESTED.  ONLY THOSE ROM ADDRESSES SPECIFICALLY
                            ;USED IN THE TEST ARE VERIFIED HERE.
                            ;
                            ;*******************************************************************************
                            ;TEST 40        TEST BRANCHES AROUND N-BIT
                            ;*******************************************************************************
003236  005212              TS40:   INC     (R2)            ;UPDATE TEST NUMBER
003240  022712  000040              CMP     #40,(R2)        ;SEQUENCE ERROR?
003244  001014                      BNE     TS41-10         ;BR TO ERROR HALT ON SEQ ERROR
                                    ;FIRST WITH N-BIT ON
003246  000257                      CCC                     ;CC=1000: JUST N-BIT
003250  000270                      SEN
003252  100001                      BPL     BRN1            ;CHECK OPPOSITE CONDITION
003254  100404                      BMI     BRN2
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 773            <====
```

```
1255  003256
1256  003256  012742  000045        BRN1:   MOV     #45,-(R2)       ;MOVE TO MAILBOX #  *******  45  *******
1257  003262  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1258  003264  000000                        HALT                    ;IMPROPER BR W/ N=1
1259                                        ;CHECK WITH N-BIT OFF
1260  003266  000277                BRN2:   SCC                     ;CC=0111
1261  003270  000250                        CLN
1262  003272  100401                        BMI     BRN3            ;CHECK OPPOSITE CONDITION
1263  003274  100004                        BPL     TS41
1264                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1265                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1266                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1267                                                                ;           WHICH FOLLOWS W/ 763           <====
1268  003276                        BRN3:
1269  003276  012742  000046              MOV     #46,-(R2)       ;MOVE TO MAILBOX #  *******  46  *******
1270  003302  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1271  003304  000000                        HALT                    ;IMPROPER BR W/ N=0
1272                                                                ;  OR SEQUENCE ERROR
1273
1274                                ;*****************************************************************************
1275                                ;
1276                                ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.
1277                                ;THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1278                                ;BVS AND BVC ARE TESTED FOR PROPER EXECUTION.  THEN THE V-BIT IS
1279                                ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1280                                ;AGAIN FOR PROPER OPERATION.
1281                                ;
1282                                ;*****************************************************************************
1283                                ;TEST 41        TEST BRANCHES AROUND V-BIT
1284                                ;*****************************************************************************
1285  003306  005212                TS41:   INC     (R2)            ;UPDATE TEST NUMBER
1286  003310  022712  000041               CMP     #41,(R2)        ;SEQUENCE ERROR?
1287  003314  001014                        BNE     TS42-10         ;BR TO ERROR HALT ON SEQ ERROR
1288                                        ;FIRST WITH V-BIT ON
1289  003316  000257                        CCC                     ;CC=0010: JUST V-BIT
1290  003320  000262                        SEV
1291  003322  102001                        BVC     BRV1            ;CHECK OPPOSITE CONDITION
1292  003324  102404                        BVS     BRV2
1293                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1294                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1295                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1296                                                                ;           WHICH FOLLOWS W/ 773           <====
1297  003326                        BRV1:
1298  003326  012742  000047              MOV     #47,-(R2)       ;MOVE TO MAILBOX #  *******  47  *******
1299  003332  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1300  003334  000000                        HALT                    ;IMPROPER BR W/ V=1
1301                                        ;CHECK WITH V-BIT OFF
1302  003336  000277                BRV2:   SCC                     ;CC=1101: ALL BVT V-BIT
1303  003340  000242                        CLV
1304  003342  102401                        BVS     BRV3            ;CHECK OPPOSITE CONDITION
1305  003344  102004                        BVC     TS42
1306                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1307                                                                ;           CONDITIONAL BRANCH INST. AND   <====
1308                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1309                                                                ;           WHICH FOLLOWS W/ 763           <====
1310  003346                        BRV3:
```

```
1311  003346  012742  000050          MOV     #50,-(R2)       ;MOVE TO MAILBOX #  *******  50  *******
1312  003352  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1313  003354  000000                  HALT                    ;IMPROPER BR W/ V=0
1314                                                          ;  OR SEQUENCE ERROR
1315
1316                          ;*****************************************************************************
1317                          ;
1318                          ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.
1319                          ;THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1320                          ;BCS AND BCC ARE TESTED FOR PROPER EXECUTION.  THEN THE C-BIT IS
1321                          ;SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1322                          ;AGAIN FOR PROPER OPERATION.
1323                          ;
1324                          ;*****************************************************************************
1325                          ;TEST 42        TEST BRANCHES AROUND C-BIT
1326                          ;*****************************************************************************
1327  003356  005212          TS42:   INC     (R2)            ;UPDATE TEST NUMBER
1328  003360  022712  000042          CMP     #42,(R2)        ;SEQUENCE ERROR?
1329  003364  001014                  BNE     TS43-10         ;BR TO ERROR HALT ON SEQ ERROR
1330                                  ;FIRST WITH C-BIT ON
1331  003366  000257                  CCC                     ;CC=0001: JUST C-BIT
1332  003370  000261                  SEC
1333  003372  103001                  BCC     BRC1            ;CHECK OPPOSITE CONDITION
1334  003374  103404                  BCS     BRC2
1335                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1336                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1337                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1338                                                          ;           WHICH FOLLOWS W/ 773           <====
1339  003376                  BRC1:
1340  003376  012742  000051          MOV     #51,-(R2)       ;MOVE TO MAILBOX #  *******  51  *******
1341  003402  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1342  003404  000000                  HALT                    ;IMPROPER BR W/ C=1
1343                                  ;CHECK WITH C-BIT OFF
1344  003406  000277          BRC2:   SCC                     ;CC=1110
1345  003410  000241                  CLC
1346  003412  103401                  BCS     BRC3            ;CHECK OPPOSITE CONDITION
1347  003414  100404                  BMI     TS43
1348                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1349                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1350                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1351                                                          ;           WHICH FOLLOWS W/ 763           <====
1352  003416                  BRC3:
1353  003416  012742  000052          MOV     #52,-(R2)       ;MOVE TO MAILBOX #  *******  52  *******
1354  003422  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1355  003424  000000                  HALT                    ;IMPROPER BR W/ C=0
1356                                                          ;  OR SEQUENCE ERROR
1357
1358                          ;*****************************************************************************
1359                          .SBTTL  MICROCODE TESTS
1360                          ;
1361                          ;       THE TEST EXERCISES  BRANCHES IN THE MICROCODE BY
1362                          ;TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
1363                          ;ALL POSSIBLE MODES.  FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
1364                          ;AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
1365                          ;ADDRESSING MODES.  BYTE MODES ARE ALSO TESTED.  AS EACH NEW
1366                          ;MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
```

```
                                ;A SMALL LOOP CONVENIENT FOR SCOPING.  THE TEST IS SET UP USING
                                ;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
                                ;VERIFIED.
                                ;       IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
                                ;FAULT.
                                ;
                                ;*****************************************************************************************



                                ;*****************************************************************************************
                                ;
                                ;       THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
                                ;MODE WITH THE SINGLE OPERAND INSTRUCTION.  FOLLOWING THE SEQUENCE CHECK,
                                ;THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
                                ;CHECKS THAT THE Z-BIT WAS PROPERLY SET.  THIS TEST CAN CHECK IR DECODE
                                ;AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0.  FOLLOWING THIS TEST
                                ;SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0.  THESE
                                ;INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
                                ;OF THE SOP INSTRUCTIONS IN THIS TEST.  THE DATA IN THIS TEST IS
                                ;OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
                                ;
                                ;*****************************************************************************************
                                ;TEST 43         TEST MODE 0 USING SOP INST.
                                ;*****************************************************************************************
003426  005212          TS43:   INC     (R2)            ;UPDATE TEST NUMBER
003430  022712  000043          CMP     #43,(R2)        ;SEQUENCE ERROR?
003434  001020                  BNE     TS44-10         ;BR TO ERROR HALT ON SEQ ERROR
003436  005000                  CLR     R0              ;TRY THE CLEAR INST.
003440  001404                  BEQ     SOP0A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 775           <====
003442  012742  000053                  MOV     #53,-(R2)       ;MOVE TO MAILBOX #  *******  53  *******
003446  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003450  000000                  HALT                    ;CLR DID NOT SET Z-BIT
003452  005200          SOP0A:  INC     R0              ;TRY THE INCREMENT INST.
003454  005100                  COM     R0              ;TRY COMPLEMENT
003456  005200                  INC     R0
003460  100404                  BMI     SOP0B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
003462  012742  000054                  MOV     #54,-(R2)       ;MOVE TO MAILBOX #  *******  54  *******
003466  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003470  000000                  HALT                    ;NEGATE DID NOT SET N-BIT
003472  005100          SOP0B:  COM     R0              ;TRY COMPLEMENT INST.
003474  001404                  BEQ     TS44
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 757           <====
003476  012742  000055                  MOV     #55,-(R2)       ;MOVE TO MAILBOX #  *******  55  *******
003502  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
1423  003504  000000                        HALT                    ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
1424                                                                  ;  OR SEQUENCE ERROR
1425
1426
1427                                  ;*****************************************************************************
1428                                  ;
1429                                  ;       THIS TEST INTRODUCES THE REMAINING SOP INSTRUCIONS AND TESTS
1430                                  ;THEM IN MODE 0.  THE PURPOSE IS TO PROVIDE A BASELINE OF
1431                                  ;INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS.  SINCE THE MICROCODE FOR
1432                                  ;THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
1433                                  ;SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
1434                                  ;FUNCTIONING.
1435                                  ;
1436                                  ;*****************************************************************************
1437                                  ;TEST 44        TEST REMAINDER OF SOP INSTS IN MODE 0
1438                                  ;*****************************************************************************
1439  003506  005212                  TS44:   INC     (R2)            ;UPDATE TEST NUMBER
1440  003510  022712  000044                  CMP     #44,(R2)        ;SEQUENCE ERROR?
1441  003514  001021                          BNE     TS45-10         ;BR TO ERROR HALT ON SEQ ERROR
1442  003516  005000                          CLR     R0              ;INITIALIZE
1443  003520  005300                          DEC     R0              ;TRY DECREMENT INST.
1444  003522  100404                          BMI     SOPOC
1445                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1446                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1447                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1448                                                                  ;           WHICH FOLLOWS W/ 774           <====
1449  003524  012742  000056                      MOV     #56,-(R2)   ;MOVE TO MAILBOX #  *******  56  *******
1450  003530  005242                              INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
1451  003532  000000                              HALT                ;N-BIT NOT SET ON DEC
1452  003534  000261                  SOPOC:  SEC                     ;INITIALIZE CARRY
1453  003536  005500                          ADC     R0              ;TRY ADD CARRY INST
1454  003540  001007                          BNE     SOPOD
1455  003542  000261                          SEC                     ;INITIALIZE CARRY
1456  003544  005600                          SBC     R0              ;TRY SUBTRACT-CARRY INST
1457  003546  100004                          BPL     SOPOD
1458  003550  005100                          COM     R0
1459  003552  005200                          INC     R0
1460  003554  005300                          DEC     R0
1461  003556  001404                          BEQ     TS45
1462                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1463                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1464                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1465                                                                  ;           WHICH FOLLOWS W/ 756           <====
1466  003560                          SOPOD:
1467  003560  012742  000057                  MOV     #57,-(R2)       ;MOVE TO MAILBOX #  *******  57  *******
1468  003564  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1469  003566  000000                          HALT                    ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F
1470                                                                  ;  OR SEQUENCE ERROR
1471
1472                                  ;*****************************************************************************
1473                                  ;
1474                                  ;       THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
1475                                  ;THE MODE 0 BYTE MICROCODE IS TESTED.  THE METHOD AND SEQUENCE
1476                                  ;OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
1477                                  ;
1478                                  ;*****************************************************************************
```

```
1479
1480
1481  003570  005212                ;TEST 45        TEST MODE 0 EVEN BYTE USING SOP INST
1482  003572  022712  000045        ;*****************************************************************************
1483  003576  001012        TS45:   INC     (R2)            ;UPDATE TEST NUMBER
1484  003600  105000                CMP     #45,(R2)        ;SEQUENCE ERROR?
1485  003602  001404                BNE     TS46-10         ;BR TO ERROR HALT ON SEQ ERROR
1486                                CLRB    R0              ;TRY CLEARING EVEN BYTE OF REGISTER
1487                                BEQ     SOPB0A
1488                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1489                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 775           <====
1490  003604  012742  000060                MOV     #60,-(R2)       ;MOVE TO MAILBOX #  *******  60  *******
1491  003610  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1492  003612  000000                HALT                    ;CLRB DID NOT SET Z-BIT
1493  003614  105100        SOPB0A: COMB    R0              ;TRY SETTING EVEN BYTE OF REGISTER
1494  003616  100002                BPL     SOPB0B
1495  003620  105200                INCB    R0              ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
1496  003622  001404                BEQ     TS46
1497                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1498                                                        ;           CONDITIONAL BRANCH INST. AND   <====
1499                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
1500                                                        ;           WHICH FOLLOWS W/ 765           <====
1501  003624                    SOPB0B:
1502  003624  012742  000061                MOV     #61,-(R2)       ;MOVE TO MAILBOX #  *******  61  *******
1503  003630  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1504  003632  000000                HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
1505                                                        ;  OR SEQUENCE ERROR
1506
1507                            ;*****************************************************************************
1508                            ;
1509                            ;       THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
1510                            ;SINGLE OPERAND MODE 1 INSTRUCTIONS.  AGAIN, THE CLR INSTRUCTION
1511                            ;IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
1512                            ;CONDITION CODES ARE SET.  OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
1513                            ;COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
1514                            ;
1515                            ;*****************************************************************************
1516                            ;TEST 46        TEST MODE 1 USING SOP INST.
1517                            ;*****************************************************************************
1518  003634  005212        TS46:   INC     (R2)            ;UPDATE TEST NUMBER
1519  003636  022712  000046        CMP     #46,(R2)        ;SEQUENCE ERROR?
1520  003642  001014                BNE     TS47-10         ;BR TO ERROR HALT ON SEQ ERROR
1521  003644  005000                CLR     R0              ;INITIALIZE R0
1522  003646  005010                CLR     (R0)            ;TRY CLEAR INST W/MODE 1
1523  003650  001404                BEQ     SOP1A
1524                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1525                                                        ;           CONDITIONAL BRANCH INST. AND   <====
1526                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
1527                                                        ;           WHICH FOLLOWS W/ 774           <====
1528  003652  012742  000062                MOV     #62,-(R2)       ;MOVE TO MAILBOX #  *******  62  *******
1529  003656  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1530  003660  000000                HALT                    ;CLR DID NOT SET Z-BIT
1531  003662  005310        SOP1A:  DEC     (R0)            ;TRY DECREMENT INST W/MODE 1
1532  003664  100003                BPL     SOP1B
1533  003666  000261                SEC                     ;INITIALIZE CARRY
1534  003670  005510                ADC     (R0)            ;TRY ADD-CARRY W/MODE 1
```

```
1535  003672  001404                    BEQ     TS47
1536                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1537                                                ;           CONDITIONAL BRANCH INST. AND   <====
1538                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1539                                                ;           WHICH FOLLOWS W/ 763           <====
1540  003674                    SOP1B:
1541  003674  012742  000063          MOV     #63,-(R2)       ;MOVE TO MAILBOX #  *******  63  *******
1542  003700  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1543  003702  000000                   HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST
1544                                                          ;  OR SEQUENCE ERROR
1545
1546                              ;*************************************************************************************
1547                              ;
1548                              ;       THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
1549                              ;SINGLE OPERAND INSTRUCTIONS.
1550                              ;       THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
1551                              ;AND VERIFIED.
1552                              ;
1553                              ;*************************************************************************************
1554                              ;TEST 47        TEST MODE 1 EVEN BYTE USING SOP INST
1555                              ;*************************************************************************************
1556  003704  005212              TS47:   INC     (R2)            ;UPDATE TEST NUMBER
1557  003706  022712  000047            CMP     #47,(R2)        ;SEQUENCE ERROR?
1558  003712  001020                    BNE     TS50-10         ;BR TO ERROR HALT ON SEQ ERROR
1559  003714  005000                    CLR     R0              ;INITIALIZE R0
1560  003716  005010                    CLR     (R0)            ;INITIALIZE LOC. 0
1561  003720  005110                    COM     (R0)
1562  003722  105010                    CLRB    (R0)            ;TRY TO CLEAR BYTE 0
1563  003724  001404                    BEQ     SOPB1A
1564                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1565                                                ;           CONDITIONAL BRANCH INST. AND   <====
1566                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1567                                                ;           WHICH FOLLOWS W/ 772           <====
1568  003726  012742  000064          MOV     #64,-(R2)       ;MOVE TO MAILBOX #  *******  64  *******
1569  003732  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1570  003734  000000                   HALT                    ;CLRB DID NOT SET Z-BIT
1571  003736  005210            SOPB1A: INC     (R0)            ;INCREMENT TO TEST WORD
1572  003740  100005                    BPL     SOPB1B
1573  003742  105110                    COMB    (R0)            ;COMPLEMENT:  ODD BYTE = 376
1574  003744  105210                    INCB    (R0)            ;INC:  ODD BYTE = 377
1575  003746  100002                    BPL     SOPB1B
1576  003750  105210                    INCB    (R0)            ;INCREMENT ODD BYTE=0
1577  003752  001404                    BEQ     TS50
1578                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1579                                                ;           CONDITIONAL BRANCH INST. AND   <====
1580                                                ;           REPLACE THE MOVE INSTRUCTION   <====
1581                                                ;           WHICH FOLLOWS W/ 757           <====
1582  003754                    SOPB1B:
1583  003754  012742  000065          MOV     #65,-(R2)       ;MOVE TO MAILBOX #  *******  65  *******
1584  003760  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1585  003762  000000                   HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST
1586                                                          ;  OR SEQUENCE ERROR
1587
1588
1589                              ;*************************************************************************************
1590                              ;
```

```
                        ;       THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL
                        ;FUNCTION CORRECTLY FOR ODD BYTES.
                        ;       THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN
                        ;EXERCISED.  CHECKS ARE MADE THAT THE PROPER BYTE IS MODIFIED AND
                        ;THE CONDITION CODES ARE CHECKED.  IT IS ALSO VERIFIED THAT THE UNADDRESSED
                        ;BYTE IS NOT ALTERED BY THE INSTRUCTION.
                        ;
                        ;*****************************************************************************
                        ;TEST 50        TEST MODE 1 ODD BYTE USING SOP INST
                        ;*****************************************************************************
003764  005212  1S50:    INC     (R2)            ;UPDATE TEST NUMBER
003766  022712  000050    CMP     #50,(R2)        ;SEQUENCE ERROR?
003772  001022            BNE     TS51-10         ;BR TO ERROR HALT ON SEQ ERROR
003774  005000            CLR     R0              ;INITIALIZE R0
003776  005010            CLR     (R0)            ;INITIALIZE LOC. 0
004000  005110            COM     (R0)
004002  005200            INC     R0              ;R0=ODD BYTE
004004  105010            CLRB    (R0)            ;TRY TO CLEAR BYTE 1
004006  001404            BEQ     SOPB1C
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 771           <====
004010  012742  000066            MOV     #66,-(R2)       ;MOVE TO MAILBOX #  *******  66  *******
004014  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004016  000000            HALT                    ;CLRB DID NOT SET Z-BIT
004020  005300  SOPB1C:   DEC     R0              ;R0=WORD ADDR.
004022  005210            INC     (R0)            ;INCREMENT TO TEST WORD
004024  005200            INC     R0              ;R0=ODD BYTE
004026  105110            COMB    (R0)            ;TRY TO COMPLEMENT BYTE 1
004030  105210            INCB    (R0)
004032  100002            BPL     SOPB1D
004034  105210            INCB    (R0)            ;TRY TO INCREMENT BYTE 1
004036  001404            BEQ     TS51
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 755           <====
004040  SOPB1D:
004040  012742  000067            MOV     #67,-(R2)       ;MOVE TO MAILBOX #  *******  67  *******
004044  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004046  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.
```

```
                                ;*************************************************************************
004050  005212                  TS51:   INC     (R2)            ;UPDATE TEST NUMBER
004052  022712  000051                  CMP     #51,(R2)        ;SEQUENCE ERROR?
004056  001023                          BNE     TS52-10         ;BR TO ERROR HALT ON SEQ ERROR
004060  005000                          CLR     R0              ;SET R0=400
004062  105100                          COMB    R0
004064  005200                          INC     R0
004066  005010                          CLR     (R0)            ;CLEAR 400
004070  005110                          COM     (R0)            ;INITIALIZE: 400=-1
004072  005020                          CLR     (R0)+           ;TRY CLEARING WITH MODE 2
004074  001404                          BEQ     SOPZA
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
004076  012742  000070                  MOV     #70,-(R2)       ;MOVE TO MAILBOX #  *******  70  *******
004102  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004104  000000                          HALT                    ;CLR INST DID NOT SET Z-BIT
004106  005300                  SOPZA:  DEC     R0              ;RESET R0
004110  005300                          DEC     R0
004112  005120                          COM     (R0)+           ;TRY COMPLEMENTING WITH MODE 2
004114  100004                          BPL     SOP2B
004116  005300                          DEC     R0              ;RESET R0
004120  005300                          DEC     R0
004122  005220                          INC     (R0)+           ;TRY INCREMENTING WITH MODE 2
004124  001404                          BEQ     TS52
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 754           <====
004126                          SOP2B:
004126  012742  000071                  MOV     #71,-(R2)       ;MOVE TO MAILBOX #  *******  71  *******
004132  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004134  000000                          HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST
                                                                ;  OR SEQUENCE ERROR

                                ;*************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
                                ;ADDRESS EVEN BYTES.  R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION
                                ;400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
                                ;MODE 2.
                                ;       R0 IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS
                                ;WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST.  THIS PROCEDURE ALSO
                                ;VERIFIES THE PROPER INCREMENTING OF THE REGISTER.
                                ;
                                ;*************************************************************************
                                ;TEST 52        TEST MODE 2 EVEN BYTE USING SOP INST.
                                ;*************************************************************************
004136  005212                  TS52:   INC     (R2)            ;UPDATE TEST NUMBER
004140  022712  000052                  CMP     #52,(R2)        ;SEQUENCE ERROR?
004144  001023                          BNE     TS53-10         ;BR TO ERROR HALT ON SEQ ERROR
004146  005000                          CLR     R0              ;SET R0=400
004150  105100                          COMB    R0
004152  005200                          INC     R0
004154  005010                          CLR     (R0)            ;CLEAR 400
```

```
1703  004156  005110                    COM     (R0)            ;INITIALIZE: 400=-1
1704  004160  105020                    CLRB    (R0)+           ;TRY TO CLEAT 400 W/MODE 2
1705  004162  001404                    BEQ     SOPB2A
1706                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1707                                                            ;           CONDITIONAL BRANCH INST. AND   <====
1708                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
1709                                                            ;           WHICH FOLLOWS W/ 770           <====
1710  004164  012742  000072              MOV     #72,-(R2)       ;MOVE TO MAILBOX #  *******  72  *******
1711  004170  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1712  004172  000000                    HALT                    ;CLR DID NOT SET Z-BIT
1713  004174  005300              SOPB2A: DEC     R0              ;RESULT R0=400
1714  004176  005210                    INC     (R0)            ;INC 400 TO TEST WORD
1715  004200  105110                    COMB    (R0)
1716  004202  105220                    INCB    (R0)+           ;TRY TO INC EVEN BYTE
1717  004204  100003                    BPL     SOPB2B
1718  004206  005300                    DEC     R0              ;RESET R0=400
1719  004210  105220                    INCB    (R0)+           ;TRY INCREMENT OF EVEN BYTE
1720  004212  001404                    BEQ     TS53
1721                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1722                                                            ;           CONDITIONAL BRANCH INST. AND   <====
1723                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
1724                                                            ;           WHICH FOLLOWS W/ 754           <====
1725  004214                        SOPB2B:
1726  004214  012742  000073              MOV     #73,-(R2)       ;MOVE TO MAILBOX #  *******  73  *******
1727  004220  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1728  004222  000000                    HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1729                                                            ;  OR SEQUENCE ERROR
1730
1731                                ;*************************************************************************************
1732                                ;
1733                                ;       THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
1734                                ;TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
1735                                ;
1736                                ;*************************************************************************************
1737                                ;TEST 53        TEST MODE 2 ODD BYTE USING SOP INST.
1738                                ;*************************************************************************************
1739  004224  005212                TS53:   INC     (R2)            ;UPDATE TEST NUMBER
1740  004226  022712  000053              CMP     #53,(R2)        ;SEQUENCE ERROR?
1741  004232  001026                    BNE     TS54-10         ;BR TO ERROR HALT ON SEQ ERROR
1742  004234  005000                    CLR     R0              ;SET R0=400
1743  004236  105100                    COMB    R0
1744  004240  005200                    INC     R0
1745  004242  005010                    CLR     (R0)            ;CLEAR LOC 400
1746  004244  005110                    COM     (R0)            ;INITIALIZE: 400=-1
1747  004246  005200                    INC     R0              ;R0=ODD BYTE
1748  004250  105020                    CLRB    (R0)+           ;TRY TO CLEAR ODD BYTE
1749  004252  001404                    BEQ     SOPB2C
1750                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1751                                                            ;           CONDITIONAL BRANCH INST. AND   <====
1752                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
1753                                                            ;           WHICH FOLLOWS W/ 767           <====
1754  004254  012742  000074              MOV     #74,-(R2)       ;MOVE TO MAILBOX #  *******  74  *******
1755  004260  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1756  004262  000000                    HALT                    ;CLRB DID NOT SET Z-BIT
1757  004264  005300              SOPB2C: DEC     R0              ;R0=WORD ADDR.
1758  004266  005300                    DEC     R0
```

```
1759  004270  005220                          INC     (R0)+           ;INCREMENT WORD
1760  004272  005300                          DEC     R0              ;POINT TO ODD BYTE
1761  004274  105110                          COMB    (R0)            ;COMPLEMENT ODD BYTE
1762  004276  105220                          INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
1763  004300  100003                          BPL     SOPB2D
1764  004302  005300                          DEC     R0              ;RESET R0 TO ODD BYTE
1765  004304  105220                          INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
1766  004306  001404                          BEQ     TS54
1767                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1768                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1769                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1770                                                                  ;           WHICH FOLLOWS W/ 751           <====
1771  004310                          SOPB2D:
1772  004310  012742  000075                 MOV     #75,-(R2)       ;MOVE TO MAILBOX #  *******  75  *******
1773  004314  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1774  004316  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1775                                                                  ;  OR SEQUENCE ERROR
1776
1777                                    ;*************************************************************************************
1778                                    ;
1779                                    ;       THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES.  PREVIOUSLY
1780                                    ;TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
1781                                    ;
1782                                    ;*************************************************************************************
1783                                    ;TEST 54        TEST MODE 0 USING NEGATE INSTRUCTION
1784                                    ;*************************************************************************************
1785  004320  005212                    TS54:   INC     (R2)            ;UPDATE TEST NUMBER
1786  004322  022712  000054                   CMP     #54,(R2)        ;SEQUENCE ERROR?
1787  004326  001035                          BNE     TS55-10         ;BR TO ERROR HALT ON SEQ ERROR
1788  004330  005000                          CLR     R0              ;SET R0=0
1789  004332  005200                          INC     R0              ;    R0=1
1790  004334  005400                          NEG     R0              ;TRY NEGATE MODE 0:  R0=-1
1791  004336  100003                          BPL     NEG00           ;CC=1001?
1792  004340  001402                          BEQ     NEG00
1793  004342  102401                          BVS     NEG00
1794  004344  103404                          BCS     NEG01
1795                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1796                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1797                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1798                                                                  ;           WHICH FOLLOWS W/ 770           <====
1799  004346                          NEG00:
1800  004346  012742  000076                 MOV     #76,-(R2)       ;MOVE TO MAILBOX #  *******  76  *******
1801  004352  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1802  004354  000000                          HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1803
1804  004356  005200                    NEG01:  INC     R0              ;TEST DATA RESULT
1805  004360  001404                          BEQ     NEG02
1806                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1807                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
1808                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
1809                                                                  ;           WHICH FOLLOWS W/ 762           <====
1810  004362  012742  000077                 MOV     #77,-(R2)       ;MOVE TO MAILBOX #  *******  77  *******
1811  004366  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1812  004370  000000                          HALT                    ;DATA RESULT OF NEGATE INCORRECT
1813
1814  004372  105100                    NEG02:  COMB    R0              ;R0=377
```

```
1815  004374  105400                  NEGB    R0              ;R0=1
1816  004376  100403                  BMI     NEG03           ;CC=0001?
1817  004400  001402                  BEQ     NEG03
1818  004402  102401                  BVS     NEG03
1819  004404  103404                  BCS     NEG04
1820                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1821                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1822                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1823                                                          ;           WHICH FOLLOWS W/ 750           <====
1824  004406                  NEG03:
1825  004406  012742  000100          MOV     #100,-(R2)      ;MOVE TO MAILBOX #  *******  100  *******
1826  004412  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1827  004414  000000                  HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
1828  004416  005300          NEG04:  DEC     R0              ;TEST DATA RESULT
1829  004420  001404                  BEQ     TS55
1830                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1831                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1832                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1833                                                          ;           WHICH FOLLOWS W/ 742           <====
1834  004422  012742  000101          MOV     #101,-(R2)      ;MOVE TO MAILBOX #  *******  101  *******
1835  004426  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1836  004430  000000                  HALT                    ;DATA RESULT OF NEGB INCORRECT
1837                                                          ;  OR SEQUENCE ERROR
1838                  ;*****************************************************************************
1839                  ;TEST 55        TEST MODE 1 USING NEGATE INST.
1840                  ;*****************************************************************************
1841  004432  005212          TS55:   INC     (R2)            ;UPDATE TEST NUMBER
1842  004434  022712  000055          CMP     #55,(R2)        ;SEQUENCE ERROR?
1843  004440  001040                  BNE     TS56-10         ;BR TO ERROR HALT ON SEQ ERROR
1844  004442  005000                  CLR     R0              ;POINT TO LOC. 0
1845  004444  005010                  CLR     (R0)            ;CLEAR LOC. 0
1846  004446  005210                  INC     (R0)            ;LOC. 0=1
1847  004450  005410                  NEG     (R0)            ;TRY NEG. LOC. 0=-1
1848  004452  100003                  BPL     NEG10           ;CC=1001
1849  004454  001402                  BEQ     NEG10
1850  004456  102401                  BVS     NEG10
1851  004460  103404                  BCS     NEG11
1852                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1853                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1854                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1855                                                          ;           WHICH FOLLOWS W/ 767           <====
1856  004462                  NEG10:
1857  004462  012742  000102          MOV     #102,-(R2)      ;MOVE TO MAILBOX #  *******  102  *******
1858  004466  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1859  004470  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1860
1861  004472  005237  000000  NEG11:  INC     @#0             ;TEST DATA RESULT
1862  004476  001404                  BEQ     NEG12
1863                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1864                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1865                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1866                                                          ;           WHICH FOLLOWS W/ 760           <====
1867  004500  012742  000103          MOV     #103,-(R2)      ;MOVE TO MAILBOX #  *******  103  *******
1868  004504  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1869  004506  000000                  HALT                    ;DATA RESULT OF NEGATE INCORRECT
1870  004510  105110          NEG12:  COMB    (R0)            ;LOC. 0=377
```

```
1871  004512  105410                  NEGB    (R0)            ;TRY NEGB LOC. 0=1
1872  004514  100403                  BMI     NEG13           ;CC=0001?
1873  004516  001402                  BEQ     NEG13
1874  004520  102401                  BVS     NEG13
1875  004522  103404                  BCS     NEG14
1876                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1877                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1878                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1879                                                          ;           WHICH FOLLOWS W/ 746           <====
1880  004524                  NEG13:
1881  004524  012742  000104          MOV     #104,-(R2)      ;MOVE TO MAILBOX #  *******  104  *******
1882  004530  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1883  004532  000000                  HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
1884  004534  005337  000000  NEG14:  DEC     @#0             ;TEST DATA RESULT
1885  004540  001404                  BEQ     TS56
1886                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1887                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1888                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1889                                                          ;           WHICH FOLLOWS W/ 737           <====
1890  004542  012742  000105          MOV     #105,-(R2)      ;MOVE TO MAILBOX #  *******  105  *******
1891  004546  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1892  004550  000000                  HALT                    ;DATA RESULT OF NEGB INCORRECT
1893                                                          ;  OR SEQUENCE ERROR
1894                  ;*****************************************************************************************
1895                  ;TEST 56        TEST MODE 2 USING NEGATE INSTRUCTION
1896                  ;*****************************************************************************************
1897  004552  005212  TS56:   INC     (R2)            ;UPDATE TEST NUMBER
1898  004554  022712  000056          CMP     #56,(R2)        ;SEQUENCE ERROR?
1899  004560  001032                  BNE     TS57-10         ;BR TO ERROR HALT ON SEQ ERROR
1900  004562  005000                  CLR     R0              ;POINT TO LOC. 0
1901  004564  005010                  CLR     (R0)            ;CLEAR LOC. 0
1902  004566  005210                  INC     (R0)            ;LOC. 0=1
1903  004570  005420                  NEG     (R0)+           ;TRY NEG.: LOC. 0=-1
1904  004572  100003                  BPL     NEG20           ;CC=1001?
1905  004574  001402                  BEQ     NEG20
1906  004576  102401                  BVS     NEG20
1907  004600  103404                  BCS     NEG21
1908                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1909                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1910                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1911                                                          ;           WHICH FOLLOWS W/ 767           <====
1912  004602                  NEG20:
1913  004602  012742  000106          MOV     #106,-(R2)      ;MOVE TO MAILBOX #  *******  106  *******
1914  004606  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1915  004610  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1916  004612  105300  NEG21:  DECB    R0              ;R0=LOC. 0
1917  004614  105300          DECB    R0
1918  004616  105420          NEGB    (R0)+           ;BYTE 0=1   R0=1
1919  004620  105420          NEGB    (R0)+           ;BYTE 1=1   R0=2
1920  004622  105340          DECB    -(R0)           ;R0=1  LOC. 0=01
1921  004624  005300          DEC     R0              ;R0=0
1922  004626  001404          BEQ     NEG22
1923                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1924                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1925                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1926                                                          ;           WHICH FOLLOWS W/ 754           <====
```

```
1927  004630  012742  000107          MOV     #107,-(R2)      ;MOVE TO MAILBOX #  *******  107  *******
1928  004634  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1929  004636  000000                  HALT                    ;REGISTER NOT INCREMENTED CORRECTLY
1930  004640  005337  000000  NEG22:  DEC     @#0             ;LOC. 0=0
1931  004644  001404                  BEQ     TS57
1932                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1933                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1934                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1935                                                          ;           WHICH FOLLOWS W/ 745           <====
1936  004646  012742  000110          MOV'    #110,-(R2)      ;MOVE TO MAILBOX #  *******  110  *******
1937  004652  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1938  004654  000000                  HALT                    ;NEG BYTE INSTRUCTIONS FAILED
1939                                                          ;  OR SEQUENCE ERROR
1940
1941                          ;*****************************************************************************
1942                          ;
1943                          ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS.  IT
1944                          ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 400
1945                          ;THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
1946                          ;INSTRUCTIONS UNDER TEST.
1947                          ;       R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
1948                          ;INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0.  THEN R0
1949                          ;IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
1950                          ;LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST.  THE PROPER INCREMENTING
1951                          ;OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
1952                          ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
1953                          ;(LOC. 400-402) HAS THE PROPER VALUES (0).
1954                          ;
1955                          ;*****************************************************************************
1956                          ;TEST 57        TEST MODE 3 USING SOP INST.
1957                          ;*****************************************************************************
1958  004656  005212  TS57:   INC     (R2)            ;UPDATE TEST NUMBER
1959  004660  022712  000057          CMP     #57,(R2)        ;SEQUENCE ERROR?
1960  004664  001020                  BNE     TS60-10         ;BR TO ERROR HALT ON SEQ ERROR
1961  004666  005000                  CLR     R0              ;SET R0=400
1962  004670  105100                  COMB    R0
1963  004672  005200                  INC     R0
1964  004674  005010                  CLR     (R0)            ;CLEAR LOC 400
1965  004676  005030                  CLR     @(R0)+          ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402
1966  004700  001404                  BEQ     SOP3A
1967                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1968                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1969                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1970                                                          ;           WHICH FOLLOWS W/ 771           <====
1971  004702  012742  000111          MOV     #111,-(R2)      ;MOVE TO MAILBOX #  *******  111  *******
1972  004706  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1973  004710  000000                  HALT                    ;CLR DID NOT SET Z-BIT
1974  004712  005300  SOP3A:  DEC     R0              ;RESET R0=400
1975  004714  005300                  DEC     R0
1976  004716  005130                  COM     @(R0)+          ;TRY TO COMPLEMENT LOC 0 OF MODE 3  ;R0=402
1977  004720  100002                  BPL     SOP3B
1978  004722  005230                  INC     @(R0)+          ;TRY TO INCREMENT LOC 0 W/MODE 3  ;R0=404
1979  004724  001404                  BEQ     TS60
1980                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1981                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1982                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                ;           WHICH FOLLOWS W/ 757          <====
SOP3B:
        MOV     #112,-(R2)      ;MOVE TO MAILBOX #  *******  112  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;CUMMULATIVE RESULT OF ABOVE INST FAILED
                                ;  OR SEQUENCE ERROR

;*****************************************************************************************
;
;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
;WHICH ADDRESS EVEN BYTES.  AGAIN, THE TARGET LOCATION 0 IS USED
;AND THE SAME TABLE AT 400 IS EMPLOYED.
;       AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION
;0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.
;       SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE
;TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.
;IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS
;THE PROPER VALUES (0).
;
;*****************************************************************************************
;TEST 60        TEST MODE 3 EVEN BYTE USING SOP INST.
;*****************************************************************************************
TS60:   INC     (R2)            ;UPDATE TEST NUMBER
        CMP     #60,(R2)        ;SEQUENCE ERROR?
        BNE     TS61-10         ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R4              ;SET R4=400
        COMB    R4
        INC     R4
        CLR     R0              ;INITIALIZE LOC. 0=-1
        CLR     (R0)
        COM     (R0)            ;LOC. 0=-1
        CLRB    @(R4)+          ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400  R4=402

                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 767           <====
        MOV     #113,-(R2)      ;MOVE TO MAILBOX #  *******  113  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;CLRB DID NOT SET Z-BIT
SOPB3A: DEC     R4              ;RESET POINTER  R4=400
        DEC     R4
        INC     @(R4)+          ;TRY INCREMENTING WORD  LOC.0=177401  R4=402
        BPL     SOPB3B
        NEGB    @(R4)+          ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
        BPL     SOPB3B
        DEC     R4              ;R4=402
        DEC     R4
        INCB    @(R4)+          ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
        BEQ     TS61

                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 751           <====
SOPB3B:
        MOV     #114,-(R2)      ;MOVE TO MAILBOX #  *******  114  *******
```

```
2039  005026  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2040  005030  000000                        HALT                    ;CUMMULATIVE RESULT OF ABOVE INST FAILED
2041                                                                  ;  OR SEQUENCE ERROR
2042
2043                                  ;*****************************************************************************
2044                                  ;
2045                                  ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
2046                                  ;WHICH ADDRESS ODD BYTES.  THE TARGET IS BYTE 1.  A TABLE AT
2047                                  ;LOC. 400-406 IS USED.  R0 SERVES AS THE TABLE POINTER.
2048                                  ;       R0 IS INITIALIZED TO 400.  LOC. 0 IS SET TO -1 USING THE
2049                                  ;FIRST TWO TABLE ENTRIES.  A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING
2050                                  ;TABLE ADDRESS AT 404.  R0 IS DECREMENTED TO 402 AND SEVERAL SOP
2051                                  ;MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER
2052                                  ;REGISTER INCREMENTING.
2053                                  ;       THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND
2054                                  ;AFTER THE TEST IS RUN.
2055                                  ;
2056                                  ;*****************************************************************************
2057                                  ;TEST 61        TEST MODE 3 ODD BYTE USING SOP INST.
2058                                  ;*****************************************************************************
2059  005032  005212                  TS61:   INC     (R2)            ;UPDATE TEST NUMBER
2060  005034  022712  000061                  CMP     #61,(R2)        ;SEQUENCE ERROR?
2061  005040  001024                          BNE     TS62-10         ;BR TO ERROR HALT ON SEQ ERROR
2062  005042  005000                          CLR     R0              ;SET R0=400
2063  005044  105100                          COMB    R0
2064  005046  005200                          INC     R0
2065  005050  005030                          CLR     @(R0)+          ;INITIALIZE
2066  005052  005130                          COM     @(R0)+          ;LOC 0=-1 R0=404
2067  005054  105030                          CLRB    @(R0)+          ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406
2068  005056  001404                          BEQ     SOPB3C
2069                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2070                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2071                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2072                                                                  ;           WHICH FOLLOWS W/ 770           <====
2073  005060  012742  000115                          MOV     #115,-(R2)      ;MOVE TO MAILBOX #  *******  115  *******
2074  005064  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2075  005066  000000                          HALT                    ;CLRB DID NOT SET Z-BIT
2076  005070  005300                  SOPB3C: DEC     R0              ;RESET R0=402
2077  005072  005300                          DEC     R0
2078  005074  005300                          DEC     R0              ;POINT TO EVEN BYTE ADDR.
2079  005076  005300                          DEC     R0
2080  005100  005230                          INC     @(R0)+          ;INCREMENT WORD LOC. 0=400 R0=404
2081  005102  105430                          NEGB    @(R0)+          ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406
2082  005104  100002                          BPL     SOPB3D
2083  005106  105230                          INCB    @(R0)+          ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410
2084  005110  001404                          BEQ     TS62
2085                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2086                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2087                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2088                                                                  ;           WHICH FOLLOWS W/ 753           <====
2089  005112                          SOPB3D:
2090  005112  012742  000116                  MOV     #116,-(R2)      ;MOVE TO MAILBOX #  *******  116  *******
2091  005116  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2092  005120  000000                          HALT                    ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED
2093                                                                  ;  OR SEQUENCE ERROR
2094                                  ;*****************************************************************************
```

```
                                        ;TEST 62        TEST MODE 3 USING NEGATE INSTRUCTION
                                        ;*************************************************************************************
005122  005212                TS62:   INC     (R2)            ;UPDATE TEST NUMBER
005124  022712  000062                CMP     #62,(R2)        ;SEQUENCE ERROR?
005130  001054                        BNE     TS63-10         ;BR TO ERROR HALT ON SEQ ERROR
005132  005000                        CLR     R0              ;R0=400
005134  105100                        COMB    R0
005136  005200                        INC     R0
005140  005010                        CLR     (R0)            ;LOC. 400=0
005142  005004                        CLR     R4              ;R4=0
005144  005014                        CLR     (R4)            ;LOC. 0=0
005146  005214                        INC     (R4)            ;LOC. 0=1
005150  005430                        NEG     @(R0)+          ;TRY NEGATE   LOC. 0=-1  R0=402
005152  100003                        BPL     NEG30           ;CC=1001?
005154  001402                        BEQ     NEG30
005156  102401                        BVS     NEG30
005160  103404                        BCS     NEG31
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 763           <====
005162                  NEG30:
005162  012742  000117                MOV     #117,-(R2)      ;MOVE TO MAILBOX #  *******  117  *******
005166  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005170  000000                        HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005172  005214                NEG31:  INC     (R4)            ;LOC. 0=0
005174  001404                        BEQ     NEG32
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 755           <====
005176  012742  000120                MOV     #120,-(R2)      ;MOVE TO MAILBOX #  *******  120  *******
005202  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005204  000000                        HALT                    ;DATA RESULT OF NEG INCORRECT
005206  105137  000001        NEG32:  COMB    @#1             ;LOC 0=177400
005212  005237  000000                INC     @#0             ;LOC. 0=177401
005216  105430                        NEGB    @(R0)+          ;TRY NEGB LOC. 0=177777  R0=404
005220  100404                        BMI     NEG33
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 743           <====
005222  012742  000121                MOV     #121,-(R2)      ;MOVE TO MAILBOX #  *******  121  *******
005226  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005230  000000                        HALT                    ;NEGB FAILED WITH EVEN BYTE
005232  105430                NEG33:  NEGB    @(R0)+          ;TRY NEGB  LOC.0=777  R0=406
005234  100004                        BPL     NEG34
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 735           <====
005236  012742  000122                MOV     #122,-(R2)      ;MOVE TO MAILBOX #  *******  122  *******
005242  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005244  000000                        HALT                    ;NEGB FAILED WITH ODD BYTE
005246  105137  000001        NEG34:  COMB    @#1             ;LOC. 0=177377
005252  105237  000001                INCB    @#1             ;LOC. 0=177777
```

```
2151  005256  005214                        INC     (R4)            ;LOC. 0=0
2152  005260  001404                        BEQ     TS63
2153                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2154                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2155                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2156                                                                  ;           WHICH FOLLOWS W/ 723           <====
2157  005262  012742  000123                MOV     #123,-(R2)      ;MOVE TO MAILBOX #  *******  123  *******
2158  005266  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2159  005270  000000                        HALT                    ;DATA RESULT OF NEGB'S INCORRECT
2160                                                                  ;  OR SEQUENCE ERROR
2161
2162                                ;*****************************************************************************
2163                                ;
2164                                ;       THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
2165                                ;R0 IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
2166                                ;LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
2167                                ;COMPLEMENTS LOC.376.
2168                                ;       TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
2169                                ;TO COMPLETE THE TEST.
2170                                ;*****************************************************************************
2171                                ;TEST 63        TEST MODE 4 USING SOP INSTS
2172                                ;*****************************************************************************
2173  005272  005212                TS63:   INC     (R2)            ;UPDATE TEST NUMBER
2174  005274  022712  000063                CMP     #63,(R2)        ;SEQUENCE ERROR?
2175  005300  001021                        BNE     TS64-10         ;BR TO ERROR HALT ON SEQ ERROR
2176  005302  005000                        CLR     R0              ;SET R0=400
2177  005304  105100                        COMB    R0
2178  005306  005200                        INC     R0
2179  005310  005040                        CLR     -(R0)           ;TRY TO CLEAR USING MODE 4
2180  005312  001404                        BEQ     SOP4A
2181                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2182                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2183                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2184                                                                  ;           WHICH FOLLOWS W/ 772           <====
2185  005314  012742  000124                MOV     #124,-(R2)      ;MOVE TO MAILBOX #  *******  124  *******
2186  005320  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2187  005322  000000                        HALT                    ;CLR DID NOT SET Z-BIT
2188  005324  005200                SOP4A:  INC     R0              ;RESET R0
2189  005326  005200                        INC     R0
2190  005330  005140                        COM     -(R0)           ;TRY TO COMPLEMENT USING MODE 4
2191  005332  100004                        BPL     SOP4B
2192  005334  005200                        INC     R0              ;MOVE POINTER
2193  005336  005200                        INC     R0
2194  005340  005240                        INC     -(R0)
2195  005342  001404                        BEQ     TS64
2196                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2197                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2198                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2199                                                                  ;           WHICH FOLLOWS W/ 756           <====
2200  005344                        SOP4B:
2201  005344  012742  000125                MOV     #125,-(R2)      ;MOVE TO MAILBOX #  *******  125  *******
2202  005350  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2203  005352  000000                        HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
2204                                                                  ;  OR SEQUENCE ERROR
2205
2206                                ;*****************************************************************************
```

```
 2207
 2208                                   ;
 2209                                   ;       THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS.  IT
 2210                                   ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 372
 2211                                   ;THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
 2212                                   ;INSTRUCTIONS UNDER TEST.
 2213                                   ;       R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
 2214                                   ;AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
 2215                                   ;LOC. 0.  THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
 2216                                   ;INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
 2217                                   ;THE TEST.  THE PROPER DECREMENTING OF THE REGISTER IS ALSO
 2218                                   ;VERIFIED IN THIS MANNER.
 2219                                   ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
 2220                                   ;(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
 2221                                   ;
 2222                                   ;*****************************************************************************************
 2223                                   ;TEST 64        TEST MODE 5 USING SOP INSTS
 2224                                   ;*****************************************************************************************
 2225 005354  005212           TS64:    INC     (R2)            ;UPDATE TEST NUMBER
 2226 005356  022712  000064             CMP     #64,(R2)        ;SEQUENCE ERROR?
 2227 005362  001025                     BNE     TS65-10         ;BR TO ERROR HALT ON SEQ ERROR
 2228 005364  012700  000370             MOV     #370,R0         ;CLEAR LOCATION 370-376
 2229 005370  005020                     CLR     (R0)+           ;370
 2230 005372  005020                     CLR     (R0)+           ;372
 2231 005374  005020                     CLR     (R0)+           ;374
 2232 005376  005010                     CLR     (R0)            ;376
 2233 005400  005000                     CLR     R0              ;SET R0=376 (LOW BYTE)
 2234 005402  005020                     CLR     (R0)+
 2235 005404  105400                     NEGB    R0
 2236 005406  005050                     CLR     @-(R0)          ;TRY TO CLEAR LOC 0 W/MODE 5
 2237 005410  001404                     BEQ     SOP5A
 2238                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 2239                                                           ;           CONDITIONAL BRANCH INST. AND   <====
 2240                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
 2241                                                           ;           WHICH FOLLOWS W/ 764           <====
 2242 005412  012742  000126             MOV     #126,-(R2)      ;MOVE TO MAILBOX #  *******  126  *******
 2243 005416  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 2244 005420  000000                     HALT                    ;CLR DID NOT SET Z-BIT
 2245 005422  005200           SOP5A:   INC     R0              ;RESET R0
 2246 005424  005200                     INC     R0
 2247 005426  005150                     COM     @-(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
 2248 005430  100002                     BPL     SOP5B
 2249 005432  005250                     INC     @-(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 5
 2250 005434  001404                     BEQ     TS65
 2251                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 2252                                                           ;           CONDITIONAL BRANCH INST. AND   <====
 2253                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
 2254 005436                             SOP5B:                 ;           WHICH FOLLOWS W/ 752           <====
 2255 005436  012742  000127             MOV     #127,-(R2)      ;MOVE TO MAILBOX #  *******  127  *******
 2256 005442  005242                     INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 2257 005444  000000                     HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
 2258                                                           ;  OR SEQUENCE ERROR
 2259
 2260                                   ;*****************************************************************************************
 2261                                   ;
 2262                                   ;       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
                                ;*****************************************************************************
005446  005212                TS65:   INC     (R2)            ;UPDATE TEST NUMBER
005450  022712  000065                CMP     #65,(R2)        ;SEQUENCE ERROR?
005454  001020                        BNE     TS66-10         ;BR TO ERROR HALT ON SEQ ERROR
005456  005000                        CLR     R0              ;SET R0=400
005460  105100                        COMB    R0
005462  005200                        INC     R0
005464  005060  177400                CLR     -400(R0)        ;TRY TO CLEAR LOCATION 0 W/MODE 6
005470  001404                        BEQ     SOP6A
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 771           <====
005472  012742  000130                MOV     #130,-(R2)      ;MOVE TO MAILBOX #  *******  130  *******
005476  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005500  000000                        HALT                    ;CLR DID NOT SET Z-BIT
005502  005160  177400        SOP6A:  COM     -400(R0)        ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
005506  100003                        BPL     SOP6B
005510  005260  177400                INC     -400(R0)        ;TRY TO INCREMENT LOCATION 0 W/MODE 6
005514  001404                        BEQ     TS66
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 757           <====
005516                          SOP6B:
005516  012742  000131                MOV     #131,-(R2)      ;MOVE TO MAILBOX #  *******  131  *******
005522  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005524  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.
                                ;*****************************************************************************
005526  005212                TS66:   INC     (R2)            ;UPDATE TEST NUMBER
005530  022712  000066                CMP     #66,(R2)        ;SEQUENCE ERROR?
005534  001021                        BNE     TS67-10         ;BR TO ERROR HALT ON SEQ ERROR
005536  005000                        CLR     R0              ;SET R0=400
005540  105100                        COMB    R0
005542  005200                        INC     R0
005544  005210                        INC     (R0)            ;R0=1
```

```
2319  005546  005070  000002          CLR     @2(R0)          ;TRY TO CLEAR LOC. 0 W/MODE 7
2320  005552  001404                  BEQ     SOP7A
2321                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2322                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2323                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2324                                                          ;           WHICH FOLLOWS W/ 770           <====
2325  005554  012742  000132          MOV     #132,-(R2)      ;MOVE TO MAILBOX #  *******  132  *******
2326  005560  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2327  005562  000000                  HALT                    ;CLR DID NOT SET Z-BIT
2328  005564  005170  000002  SOP7A:  COM     @2(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
2329  005570  100003                  BPL     SOP7B
2330  005572  005270  000002          INC     @2(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 7
2331  005576  001404                  BEQ     TS67
2332                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2333                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2334                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2335                                                          ;           WHICH FOLLOWS W/ 756           <====
2336  005600          SOP7B:
2337  005600  012742  000133          MOV     #133,-(R2)      ;MOVE TO MAILBOX #  *******  133  *******
2338  005604  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2339  005606  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
2340                                                          ;  OR SEQUENCE ERROR
2341
2342                                  ;*****************************************************************************
2343                                  ;TEST 67        TEST MODE 4 WITH NEGATE INSTRUCTION
2344                                  ;*****************************************************************************
2345  005610  005212  TS67:   INC     (R2)            ;UPDATE TEST NUMBER
2346  005612  022712  000067          CMP     #67,(R2)        ;SEQUENCE ERROR?
2347  005616  001024                  BNE     TS70-10         ;BR TO ERROR HALT ON SEQ ERROR
2348  005620  005000                  CLR     R0
2349  005622  005010                  CLR     (R0)
2350  005624  005120                  COM     (R0)+           ;LOC. 0=177777,  R0=2
2351  005626  005440                  NEG     -(R0)           ;TRY NEGATE, LOC. 0=1
2352  005630  100403                  BMI     NEG40           ;CC=0001?
2353  005632  001402                  BEQ     NEG40
2354  005634  102401                  BVS     NEG40
2355  005636  103404                  BCS     NEG41
2356                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2357                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2358                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2359                                                          ;           WHICH FOLLOWS W/ 767           <====
2360  005640          NEG40:
2361  005640  012742  000134          MOV     #134,-(R2)      ;MOVE TO MAILBOX #  *******  134  *******
2362  005644  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2363  005646  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2364  005650  005400          NEG41:  NEG     R0              ;TST R0 WITH A NEG.
2365  005652  001404                  BEQ     NEG42
2366                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2367                                                          ;           CONDITIONAL BRANCH INST. AND   <====
2368                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
2369                                                          ;           WHICH FOLLOWS W/ 761           <====
2370  005654  012742  000135          MOV     #135,-(R2)      ;MOVE TO MAILBOX #  *******  135  *******
2371  005660  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2372  005662  000000                  HALT                    ;R0 NOT DECREMENTED PROPERLY
2373  005664  005310          NEG42:  DEC     (R0)            ;TEST DTA RESULT OF NEG
2374  005666  001404                  BEQ     TS70
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 753           <====
005670  012742  000136          MOV     #136,-(R2)      ;MOVE TO MAILBOX #  *******  136  *******
005674  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005676  000000                  HALT                    ;DATA RESULT OF NEG INCORRECT
                                                ;  OR SEQUENCE ERROR
;*************************************************************************************************
;TEST 70        TEST MODE 5 WITH NEGATE INSTRUCTION
;*************************************************************************************************
005700  005212  TS70:   INC     (R2)            ;UPDATE TEST NUMBER
005702  022712  000070          CMP     #70,(R2)        ;SEQUENCE ERROR?
005706  001031                  BNE     TS71-10         ;BR TO ERROR HALT ON SEQ ERROR
005710  005000                  CLR     R0              ;R0=0
005712  005010                  CLR     (R0)            ;LOC. 0=0
005714  105100                  COMB    R0              ;R0=377
005716  005200                  INC     R0              ;R0=400
005720  005010                  CLR     (R0)            ;SET 400 = 0
005722  005004                  CLR     R4              ;R4=0
005724  005314                  DEC     (R4)            ;LOC. 0=177777
005726  005450                  NEG     @-(R0)          ;TRY NEGATE:  LOC. 0=1
005730  100403                  BMI     NEG50           ;CC=0001?
005732  001402                  BEQ     NEG50
005734  102401                  BVS     NEG50
005736  103404                  BCS     NEG51
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 763           <====
005740  NEG50:
005740  012742  000137          MOV     #137,-(R2)      ;MOVE TO MAILBOX #  *******  137  *******
005744  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005746  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005750  005314  NEG51:  DEC     (R4)
005752  001404                  BEQ     NEG52
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 755           <====
005754  012742  000140          MOV     #140,-(R2)      ;MOVE TO MAILBOX #  *******  140  *******
005760  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005762  000000                  HALT                    ;DATA RESULT OF NEG INCORRECT
005764  105100  NEG52:  COMB    R0
005766  005300                  DEC     R0
005770  001404                  BEQ     TS71
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 746           <====
005772  012742  000141          MOV     #141,-(R2)      ;MOVE TO MAILBOX #  *******  141  *******
005776  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006000  000000                  HALT                    ;REGISTER NOT DECREMENTED PROPERLY
                                                ;  OR SEQUENCE ERROR
;*************************************************************************************************
;TEST 71        TEST MODE 6 WITH NEGATE
```

```
2431                                    ;*****************************************************************************
2432  006002  005212                    TS71:   INC     (R2)            ;UPDATE TEST NUMBER
2433  006004  022712  000071                    CMP     #71,(R2)        ;SEQUENCE ERROR?
2434  006010  001022                            BNE     TS72-10         ;BR TO ERROR HALT ON SEQ ERROR
2435  006012  005000                            CLR     R0              ;R0=0
2436  006014  005004                            CLR     R4              ;R4=0
2437  006016  105100                            COMB    R0              ;R0=377
2438  006020  005014                            CLR     (R4)            ;LOC. 0=0
2439  006022  105024                            CLRB    (R4)+           ;LOC. 0=177777,  R4=1
2440  006024  105114                            COMB    (R4)            ;LOC. 0=177400
2441  006026  005460  177401                    NEG     -377(R0)        ;LOC. 0=400
2442  006032  100403                            BMI     NEG60           ;CC=0001
2443  006034  001402                            BEQ     NEG60
2444  006036  102401                            BVS     NEG60
2445  006040  103404                            BCS     NEG61
2446                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2447                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2448                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2449                                                                    ;           WHICH FOLLOWS W/ 763           <====
2450  006042                            NEG60:
2451  006042  012742  000142                    MOV     #142,-(R2)      ;MOVE TO MAILBOX #  *******  142  *******
2452  006046  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2453  006050  000000                            HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2454  006052  105314                    NEG61:  DECB    (R4)
2455  006054  001404                            BEQ     TS72
2456                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2457                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2458                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2459                                                                    ;           WHICH FOLLOWS W/ 755           <====
2460  006056  012742  000143                    MOV     #143,-(R2)      ;MOVE TO MAILBOX #  *******  143  *******
2461  006062  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2462  006064  000000                            HALT                    ;DATA RESULT OF NEG INCORRECT
2463                                                                    ;  OR SEQUENCE ERROR
2464                                    ;*****************************************************************************
2465                                    ;TEST 72        TEST MODE 7 W/ NEGATE
2466                                    ;*****************************************************************************
2467  006066  005212                    TS72:   INC     (R2)            ;UPDATE TEST NUMBER
2468  006070  022712  000072                    CMP     #72,(R2)        ;SEQUENCE ERROR?
2469  006074  001024                            BNE     TS73-10         ;BR TO ERROR HALT ON SEQ ERROR
2470  006076  005000                            CLR     R0              ;R0=0
2471  006100  005010                            CLR     (R0)            ;LOC. 0=0
2472  006102  005110                            COM     (R0)            ;LOC. 0=177777
2473  006104  105100                            COMB    R0              ;R0=377
2474  006106  105470  000005                    NEGB    @5(R0)          ;R0+5=404,  404=1,  LOC. 0=777
2475  006112  100403                            BMI     NEG70           ;CC=0001?
2476  006114  001402                            BEQ     NEG70
2477  006116  102401                            BVS     NEG70
2478  006120  103404                            BCS     NEG71
2479                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2480                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2481                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2482                                                                    ;           WHICH FOLLOWS W/ 765           <====
2483  006122                            NEG70:
2484  006122  012742  000144                    MOV     #144,-(R2)      ;MOVE TO MAILBOX #  *******  144  *******
2485  006126  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2486  006130  000000                            HALT                    ;NEG DID NOT SET CC'S CORRECTLY
```

```
2487  006132  105100                  NEG71:  COMB    R0              ;R0=0
2488  006134  105120                          COMB    (R0)+           ;LOC. 0=400,  R0=1
2489  006136  105310                          DECB    (R0)            ;LOC. 0=0
2490  006140  005467  171634                  NEG     0               ;USE NEG MODE 67 TO TST FOR ZERO
2491  006144  001404                          BEQ     TS73
2492                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2493                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2494                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2495                                                                  ;           WHICH FOLLOWS W/ 753           <====
2496  006146  012742  000145                      MOV     #145,-(R2)  ;MOVE TO MAILBOX #  *******  145  *******
2497  006152  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2498  006154  000000                            HALT                ;DATA RESULT OF NEG WAS INCORRECT
2499                                                                  ;  OR SEQUENCE ERROR
2500
2501                                  ;*************************************************************************************
2502                                  ;
2503                                  ;       THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
2504                                  ;INSTRUCTIONS.  CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
2505                                  ;INSTRUCTION (SOPX).  THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
2506                                  ;77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
2507                                  ;OF THESE INSTRUCTIONS.
2508                                  ;
2509                                  ;*************************************************************************************
2510                                  ;TEST 73        TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
2511                                  ;*************************************************************************************
2512  006156  005212                  TS73:   INC     (R2)            ;UPDATE TEST NUMBER
2513  006160  022712  000073                  CMP     #73,(R2)        ;SEQUENCE ERROR?
2514  006164  001017                          BNE     SOPB            ;BR TO ERROR HALT ON SEQ ERROR
2515  006166  005027                          CLR     (R7)+           ;CLEAR NEXT LOCATION: (SOPX)
2516  006170  177777                  SOPX:   -1                      ;USE MODE 27
2517  006172  001404                          BEQ     SOPA
2518                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2519                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2520                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2521                                                                  ;           WHICH FOLLOWS W/ 774           <====
2522  006174  012742  000146                      MOV     #146,-(R2)  ;MOVE TO MAILBOX #  *******  146  *******
2523  006200  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2524  006202  000000                            HALT                ;CLR DID NOT SET Z-BIT
2525  006204  005237  006170          SOPA:   INC     @#SOPX          ;INC SOPX W/MODE 37
2526  006210  005467  177754                  NEG     SOPX            ;NEGATE SOPX W/MODE 67
2527  006214  100003                          BPL     SOPB
2528  006216  005277  000012                  INC     @SOPXAD         ;INC SOPX W/MODE 77
2529  006222  001405                          BEQ     TS74
2530                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2531                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2532                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2533                                                                  ;           WHICH FOLLOWS W/ 760           <====
2534  006224                          SOPB:
2535  006224  012742  000147                      MOV     #147,-(R2)  ;MOVE TO MAILBOX #  *******  147  *******
2536  006230  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2537  006232  000000                            HALT                ;INC DID NOT SET Z-BIT
2538                                                                  ;  OR SEQUENCE ERROR
2539  006234  006170                  SOPXAD: SOPX                    ;INDIRECT ADDRESS OF SOPX
2540
2541                                  ;*************************************************************************************
2542                                  ;
```

```
                        ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
                        ;USING MODE 0.  R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
                        ;TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION.  A TST INSTRUCTION
                        ;IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION
                        ;CODES.
                        ;
                        ;*****************************************************************************
                        ;TEST 74        TEST MODE 0 SOP NON-MODIFYING
                        ;*****************************************************************************
006236  005212  TS74:    INC     (R2)            ;UPDATE TEST NUMBER
006240  022712  000074    CMP     #74,(R2)        ;SEQUENCE ERROR?
006244  001010            BNE     TS75-10         ;BR TO ERROR HALT ON SEQ ERROR
006246  005000            CLR     R0              ;INITIALIZE R0=0
006250  000277            SCC                     ;SET CC=1011
006252  000244            CLZ
006254  005700            TST     R0              ;TRY TST W/ MODE 0
006256  102403            BVS     SNM0A           ;CHECK THAT CC=0100
006260  100402            BMI     SNM0A
006262  103401            BCS     SNM0A
006264  001404            BEQ     TS75
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 767           <====
006266          SNM0A:
006266  012742  000150    MOV     #150,-(R2)      ;MOVE TO MAILBOX #  *******  150  *******
006272  005242            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006274  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
                        ;*****************************************************************************
006276  005212  TS75:    INC     (R2)            ;UPDATE TEST NUMBER
006300  022712  000075    CMP     #75,(R2)        ;SEQUENCE ERROR?
006304  001010            BNE     TS76-10         ;BR TO ERROR HALT ON SEQ ERROR
006306  005000            CLR     R0              ;INITIALIZE
006310  105100            COMB    R0              ;R0=377
006312  000277            SCC                     ;SET CC=0111
006314  000250            CLN
006316  105700            TSTB    R0              ;TRY TST EVEN BYTE
006320  102402            BVS     SNMB0A          ;CHECK CC=1000
006322  101401            BLOS    SNMB0A
006324  100404            BMI     TS76
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 767           <====
```

```
2599  006326
2600  006326  012742  000151        SNMB0A:
2601  006332  005242                        MOV     #151,-(R2)      ;MOVE TO MAILBOX #  *******  151  *******
2602  006334  000000                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2603                                        HALT                    ;CONDITION CODES NOT SET PROPERLY
2604                                                                ;  OR SEQUENCE ERROR
2605                                ;*****************************************************************************
2606                                ;
2607                                ;       THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
2608                                ;R0 IS USED TO POINT TO AND CLEAR LOC. 0.  THE COMPLEMENT OF THE
2609                                ;EXPECTED CONDITION CODES ARE LOADED IN THE PSW.  A TST INSTRUCTION
2610                                ;IS THEN EXECUTED ON LOC. 0 USING R0 AND CONDITIONAL BRANCHES TEST
2611                                ;THE RESULTS.
2612                                ;
2613                                ;*****************************************************************************
2614                                ;TEST 76        TEST MODE 1 SOP NON-MODIFYING
2615                                ;*****************************************************************************
2616  006336  005212                TS76:   INC     (R2)            ;UPDATE TEST NUMBER
2617  006340  022712  000076                CMP     #76,(R2)        ;SEQUENCE ERROR?
2618  006344  001011                        BNE     TS77-10         ;BR TO ERROR HALT ON SEQ ERROR
2619  006346  005000                        CLR     R0              ;POINT TO LOC 0
2620  006350  005010                        CLR     (R0)            ;CLEAR LOC 0
2621  006352  000277                        SCC                     ;INITIALIZE
2622  006354  000244                        CLZ                     ;CC=1011
2623  006356  005710                        TST     (R0)            ;TRY TST W/ MODE 1
2624  006360  102403                        BVS     SNM1A           ;CHECK CC=0100
2625  006362  103402                        BCS     SNM1A
2626  006364  100401                        BMI     SNM1A
2627  006366  001404                        BEQ     TS77
2628                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2629                                                                ;           CONDITIONAL BRANCH INST. AND   <====
2630                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
2631                                                                ;           WHICH FOLLOWS W/ 766           <====
2632  006370                        SNM1A:
2633  006370  012742  000152                MOV     #152,-(R2)      ;MOVE TO MAILBOX #  *******  152  *******
2634  006374  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2635  006376  000000                        HALT                    ;CC'S NOT SET PROPERLY
2636                                                                ;  OR SEQUENCE ERROR
2637
2638                                ;*****************************************************************************
2639                                ;
2640                                ;       THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
2641                                ;THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
2642                                ;AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
2643                                ;PROPER CONDITION CODE BITS.
2644                                ;
2645                                ;*****************************************************************************
2646                                ;TEST 77        TEST MODE 1 BYTE INST. NON-MODIFYING
2647                                ;*****************************************************************************
2648  006400  005212                TS77:   INC     (R2)            ;UPDATE TEST NUMBER
2649  006402  022712  000077                CMP     #77,(R2)        ;SEQUENCE ERROR?
2650  006406  001026                        BNE     TS100-10        ;BR TO ERROR HALT ON SEQ ERROR
2651  006410  005000                        CLR     R0              ;POINT TO LOC 0
2652  006412  005010                        CLR     (R0)            ;CLEAR LOC 0
2653  006414  105110                        COMB    (R0)            ;COMPLEMENT BYTE 0
2654  006416  000277                        SCC                     ;SET CC=0111
```

```
                                        CLN
                                        TSTB    (R0)            ;TRY TST ON EVEN BYTE
                                        BVS     SNMB1A
                                        BLOS    SNMB1A
                                        BMI     SNMB1B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
006432                                  SNMB1A:
006432  012742  000153                        MOV     #153,-(R2)      ;MOVE TO MAILBOX #  *******  153  *******
006436  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006440  000000                                HALT                    ;CC'S NOT CORRECT
006442  005000                          SNMB1B: CLR     R0
006444  005200                                INC     R0
006446  000277                                SCC                     ;SET CC=1011
006450  000244                                CLZ
006452  105710                                TSTB    (R0)            ;TRY TO TST AN ODD BYTE
006454  102403                                BVS     SNMB1C          ;CHECK CC=0100
006456  103402                                BCS     SNMB1C
006460  100401                                BMI     SNMB1C
006462  001404                                BEQ     TS100
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 751           <====
006464                                  SNMB1C:
006464  012742  000154                        MOV     #154,-(R2)      ;MOVE TO MAILBOX #  *******  154  *******
006470  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006472  000000                                HALT                    ;CC'S NOT CORRECT
                                                                ;  OR SEQUENCE ERROR

;*****************************************************************************************
;
;       THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
;USING MODE 2.  IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
;MODE 1 TESTS.  ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
;IT IS INCREMENTED PROPERLY.
;
;*****************************************************************************************
;TEST 100       TEST MODE 2 WITH SOP NON-MODIFYING
;*****************************************************************************************
006474  005212                          TS100:  INC     (R2)            ;UPDATE TEST NUMBER
006476  022712  000100                        CMP     #100,(R2)       ;SEQUENCE ERROR?
006502  001020                                BNE     TS101-10        ;BR TO ERROR HALT ON SEQ ERROR
006504  005000                                CLR     R0              ;INITIALIZE R0=0
006506  005010                                CLR     (R0)            ;CLEAR LOC 0
006510  000277                                SCC                     ;SET CC=1011
006512  000244                                CLZ
006514  005720                                TST     (R0)+           ;TRY TST W/ MODE 2
006516  102403                                BVS     SNM2A           ;CHECK CC=0100
006520  103402                                BCS     SNM2A
006522  100401                                BMI     SNM2A
006524  001404                                BEQ     SNM2B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
006526                                          SNM2A:
006526  012742  000155                                MOV     #155,-(R2)      ;MOVE TO MAILBOX #  *******  155  *******
006532  005242                                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006534  000000                                      HALT                    ;CC'S NOT CORRECT
006536  005300                                  SNM2B: DEC     R0              ;RESET R0
006540  005300                                      DEC     R0
006542  001404                                      BEQ     TS101
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
006544  012742  000156                                MOV     #156,-(R2)      ;MOVE TO MAILBOX #  *******  156  *******
006550  005242                                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006552  000000                                      HALT                    ;MODE 2 DID NOT INC REQ CORRECTLY
                                                                ;  OR SEQUENCE ERROR

                                                ;*******************************************************************************
                                                ;
                                                ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
                                                ;INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0.  WITH LOCATION 0
                                                ;SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
                                                ;TO VERIFY THE CORRECT CC ARE SET.  THE REGISTER IS CHECKED FOR
                                                ;PROPER INCREMENTING.
                                                ;
                                                ;*******************************************************************************
                                                ;TEST 101       TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
                                                ;*******************************************************************************
006554  005212                                  TS101:  INC     (R2)            ;UPDATE TEST NUMBER
006556  022712  000101                                CMP     #101,(R2)       ;SEQUENCE ERROR?
006562  001042                                      BNE     TS102-10        ;BR TO ERROR HALT ON SEQ ERROR
006564  005000                                      CLR     R0              ;CLEAR R0
006566  005010                                      CLR     (R0)            ;CLEAR LOC 0
006570  105110                                      COMB    (R0)            ;SET LOC 0=377
006572  000277                                      SCC                     ;SET CC=0111
006574  000250                                      CLN
006576  105720                                      TSTB    (R0)+           ;TRY TST OF EVEN BYTE
006600  102402                                      BVS     SNMB2A
006602  101401                                      BLOS    SNMB2A
006604  100404                                      BMI     SNMB2B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
006606                                          SNMB2A:
006606  012742  000157                                MOV     #157,-(R2)      ;MOVE TO MAILBOX #  *******  157  *******
006612  005242                                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006614  000000                                      HALT                    ;CC'S NOT SET CORRECTLY
006616  005300                                  SNMB2B: DEC     R0              ;DECREMENT R0
006620  001404                                      BEQ     SNMB2C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
006622  012742  000160                                MOV     #160,-(R2)      ;MOVE TO MAILBOX #  *******  160  *******
```

```
2767  006626  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2768  006630  000000                    HALT                    ;MODE 2 DID NOT INC REG CORRECTLY
2769  006632  005200            SNMB2C: INC     R0              ;POINT TO ODD BYTE
2770  006634  000277                    SCC                     ;SET CC=1011
2771  006636  000244                    CLZ
2772  006640  105720                    TSTB    (R0)+           ;TRY TST OF ODD BYTE
2773  006642  102403                    BVS     SNMB2D          ;CHECK CC'S=0100
2774  006644  103402                    BCS     SNMB2D
2775  006646  100401                    BMI     SNMB2D
2776  006650  001404                    BEQ     SNMB2E
2777                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2778                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2779                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2780                                                            ;           WHICH FOLLOWS W/ 744           <====
2781  006652                    SNMB2D:
2782  006652  012742  000161          MOV     #161,-(R2)      ;MOVE TO MAILBOX #  *******  161  *******
2783  006656  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2784  006660  000000                    HALT                    ;CC'S NOT CORRECT
2785  006662  005300            SNMB2E: DEC     R0
2786  006664  005300                    DEC     R0
2787  006666  001404                    BEQ     TS102
2788                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2789                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2790                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2791                                                            ;           WHICH FOLLOWS W/ 735           <====
2792  006670  012742  000162          MOV     #162,-(R2)      ;MOVE TO MAILBOX #  *******  162  *******
2793  006674  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2794  006676  000000                    HALT                    ;R0 DID NOT INCREMENT PROPERLY
2795                                                            ;  OR SEQUENCE ERROR
2796
2797                              ;*****************************************************************************************
2798                              ;
2799                              ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
2800                              ;A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
2801                              ;THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
2802                              ;TST MODE 3 INSTRUCTION.
2803                              ;
2804                              ;*****************************************************************************************
2805                              ;TEST 102       TEST MODE 3 W/ SOP NON-MODIFYING INSTS
2806                              ;*****************************************************************************************
2807  006700  005212            TS102:  INC     (R2)            ;UPDATE TEST NUMBER
2808  006702  022712  000102            CMP     #102,(R2)       ;SEQUENCE ERROR?
2809  006706  001022                    BNE     TS103-10        ;BR TO ERROR HALT ON SEQ ERROR
2810  006710  005000                    CLR     R0              ;R0=0
2811  006712  005010                    CLR     (R0)            ;CLEAR LOC 0
2812  006714  105100                    COMB    R0              ;R0=376
2813  006716  005300                    DEC     R0
2814  006720  000277                    SCC                     ;SET CC=1011
2815  006722  000244                    CLZ
2816  006724  005730                    TST     @(R0)+          ;TRY TST W/ MODE 3
2817  006726  102403                    BVS     SNM3A           ;CHECK CC=0100
2818  006730  103402                    BCS     SNM3A
2819  006732  100401                    BMI     SNM3A
2820  006734  001404                    BEQ     SNM3B
2821                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2822                                                            ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
                                                                    ;          WHICH FOLLOWS W/ 764           <====
006736
006736  012742  000163          SNM3A:  MOV     #163,-(R2)          ;MOVE TO MAILBOX #  *******  163  *******
006742  005242                          INC     -(R2)               ;SET MSGTYP TO FATAL ERROR
006744  000000                          HALT                        ;CC'S NOT CORRECT
006746  005300                  SNM3B:  DEC     R0                  ;R0=377
006750  105100                          COMB    R0                  ;R0=0
006752  001404                          BEQ     TS103
                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                    ;          CONDITIONAL BRANCH INST. AND   <====
                                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
                                                                    ;          WHICH FOLLOWS W/ 755           <====
006754  012742  000164                  MOV     #164,-(R2)          ;MOVE TO MAILBOX #  *******  164  *******
006760  005242                          INC     -(R2)               ;SET MSGTYP TO FATAL ERROR
006762  000000                          HALT                        ;MODE 3 DID NOT INC REG CORRECTLY
                                                                    ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3
                                ;LOC. 0 IS SET TO 377.  TABLE AT LOC. 402-404 IS USED TO TEST
                                ;BYTE 0 AND BYTE 1.  THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND
                                ;THE CC'S ARE VERIFIED.
                                ;       THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND
                                ;AFTER THE TEST IS RUN.
                                ;
                                ;*******************************************************************************
                                ;TEST 103       TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
                                ;*******************************************************************************
006764  005212                  TS103:  INC     (R2)                ;UPDATE TEST NUMBER
006766  022712  000103                  CMP     #103,(R2)           ;SEQUENCE ERROR?
006772  001036                          BNE     TS104-10            ;BR TO ERROR HALT ON SEQ ERROR
006774  005000                          CLR     R0                  ;R0=0
006776  005010                          CLR     (R0)                ;CLEAR LOC 0
007000  105110                          COMB    (R0)                ;LOC. 0 =377
007002  105100                          COMB    R0
007004  005200                          INC     R0
007006  005720                          TST     (R0)+               ;R0=402
007010  000277                          SCC                         ;CC=0111
007012  000250                          CLN
007014  105730                          TSTB    @(R0)+              ;TRY TST OF EVEN BYTE
007016  102402                          BVS     SNMB3A              ;CHECK CC=1000
007020  101401                          BLOS    SNMB3A
007022  100404                          BMI     SNMB3B
                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                    ;          CONDITIONAL BRANCH INST. AND   <====
                                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
                                                                    ;          WHICH FOLLOWS W/ 763           <====
007024
007024  012742  000165          SNMB3A: MOV     #165,-(R2)          ;MOVE TO MAILBOX #  *******  165  *******
007030  005242                          INC     -(R2)               ;SET MSGTYP TO FATAL ERROR
007032  000000                          HALT                        ;CC'S NOT CORRECT
007034  000277                  SNMB3B: SCC                         ;SET CC=1011
007036  000244                          CLZ
007040  105730                          TSTB    @(R0)+              ;TRY TST OF ODD BYTE
```

```
2879  007042  102403                  BVS     SNMB3C          ;CHECK CC=0100
2880  007044  103402                  BCS     SNMB3C
2881  007046  100401                  BMI     SNMB3C
2882  007050  001404                  BEQ     SNMB3D
2883                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2884                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2885                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2886                                                            ;           WHICH FOLLOWS W/ 750           <====
2887  007052                  SNMB3C:
2888  007052  012742  000166          MOV     #166,-(R2)      ;MOVE TO MAILBOX #  *******  166  *******
2889  007056  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2890  007060  000000                  HALT                    ;CC'S NOT CORRECT
2891  007062  005720          SNMB3D: TST     (R0)+           ;R0=410
2892  007064  005710                  TST     (R0)
2893  007066  100404                  BMI     TS104
2894                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2895                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2896                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2897                                                            ;           WHICH FOLLOWS W/ 741           <====
2898  007070  012742  000167          MOV     #167,-(R2)      ;MOVE TO MAILBOX #  *******  167  *******
2899  007074  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2900  007076  000000                  HALT                    ;TSTB DID NOT INCREMENT R0 CORRETCLY
2901                                                            ;  OR SEQUENCE ERROR
2902                          ;*****************************************************************************
2903                          ;
2904                          ;       THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
2905                          ;LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
2906                          ;EXPECTED RESULTS.  R0 AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
2907                          ;THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
2908                          ;IS CHECKED FOR PROPER DECREMENTING.
2909                          ;
2910                          ;*****************************************************************************
2911                          ;TEST 104       TEST MODE 4 W/ SOP NON-MODIFYING INSTS
2912                          ;*****************************************************************************
2913  007100  005212          TS104:  INC     (R2)            ;UPDATE TEST NUMBER
2914  007102  022712  000104          CMP     #104,(R2)       ;SEQUENCE ERROR?
2915  007106  001017                  BNE     TS105-10        ;BR TO ERROR HALT ON SEQ ERROR
2916  007110  005000                  CLR     R0              ;R0=0
2917  007112  005010                  CLR     (R0)            ;LOC 0=0
2918  007114  005120                  COM     (R0)+           ;LOC 0=-1
2919  007116  000277                  SCC                     ;SET CC=1011
2920  007120  000244                  CLZ
2921  007122  005740                  TST     -(R0)           ;TRY TST W/ MODE 4
2922  007124  102402                  BVS     SNM4A           ;CHECK CC=0100
2923  007126  101401                  BLOS    SNM4A
2924  007130  100404                  BMI     SNM4B
2925                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2926                                                            ;           CONDITIONAL BRANCH INST. AND   <====
2927                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
2928                                                            ;           WHICH FOLLOWS W/ 766           <====
2929  007132                  SNM4A:
2930  007132  012742  000170          MOV     #170,-(R2)      ;MOVE TO MAILBOX #  *******  170  *******
2931  007136  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2932  007140  000000                  HALT                    ;CC'S NOT CORRECT
2933  007142  005700          SNM4B:  TST     R0
2934  007144  001404                  BEQ     TS105
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 760           <====
007146  012742  000171           MOV     #171,-(R2)      ;MOVE TO MAILBOX #  *******  171  *******
007152  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007154  000000                   HALT                    ;TST MODE 4 DID NOT DEC R0 CORRECTLY
                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
                        ;IT USES A POINTER AT LOC. 376 TO TEST LOC. 0.  R0 IS SET
                        ;TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
                        ;R0 IS CHECKED TO INSURE PROPER DECREMENTING.
                        ;
                        ;*****************************************************************************
                        ;TEST 105       TEST MODE 5 W/ SOP NON-MODIFYING INSTS
                        ;*****************************************************************************
007156  005212           TS105:  INC     (R2)            ;UPDATE TEST NUMBER
007160  022712  000105           CMP     #105,(R2)       ;SEQUENCE ERROR?
007164  001022                   BNE     TS106-10        ;BR TO ERROR HALT ON SEQ ERROR
007166  005000                   CLR     R0              ;R0=0
007170  005010                   CLR     (R0)            ;LOC 0=0
007172  005110                   COM     (R0)            ;LOC 0=-1
007174  105100                   COMB    R0              ;R0=377
007176  005200                   INC     R0              ;R0=400
007200  000277                   SCC                     ;SET CC=0111
007202  000250                   CLN
007204  005750                   TST     @-(R0)          ;TRY TST W/ MODE 5
007206  102402                   BVS     SNM5A           ;CHECK CC=1000
007210  101401                   BLOS    SNM5A
007212  100404                   BMI     SNM5B
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 764           <====
007214          SNM5A:
007214  012742  000172           MOV     #172,-(R2)      ;MOVE TO MAILBOX #  *******  172  *******
007220  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007222  000000                   HALT                    ;CC'S NOT SET PROPERLY
007224  005200           SNM5B:  INC     R0              ;R0=377
007226  105100                   COMB    R0              ;R0=0
007230  001404                   BEQ     TS106
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 755           <====
007232  012742  000173           MOV     #173,-(R2)      ;MOVE TO MAILBOX #  *******  173  *******
007236  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007240  000000                   HALT                    ;MODE 5 DID NOT DEC R0 CORRECTLY
                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.
```

```
                                        ;R0 IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED
                                        ;USING R0 AND AN OFFSET OF -377.  THE CC'S ARE CHECKED AS WELL
                                        ;AS R0 TO INSURE IT WAS NOT ALTERED.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 106       TEST MODE 6 W/ SOP NON-MODIFYING INSTS
                                        ;*****************************************************************************
007242  005212                TS106:  INC     (R2)            ;UPDATE TEST NUMBER
007244  022712  000106                CMP     #106,(R2)       ;SEQUENCE ERROR?
007250  001021                        BNE     TS107-10        ;BR TO ERROR HALT ON SEQ ERROR
007252  005000                        CLR     R0              ;R0=0
007254  005010                        CLR     (R0)            ;LOC 0=0
007256  005110                        COM     (R0)            ;LOC 0=-1
007260  105100                        COMB    R0              ;R0=377
007262  000277                        SCC                     ;SET CC=0111
007264  000250                        CLN
007266  005760  177401                TST     -377(R0)        ;TRY TST W/ MODE 6
007272  102402                        BVS     SNM6A           ;CHECK CC=1000
007274  101401                        BLOS    SNM6A
007276  100404                        BMI     SNM6B
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 764           <====
007300                  SNM6A:
007300  012742  000174                MOV     #174,-(R2)      ;MOVE TO MAILBOX #  *******  174  *******
007304  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007306  000000                        HALT                    ;CC'S INCORRECT
007310  105100                SNM6B:  COMB    R0              ;R0=0
007312  001404                        BEQ     TS107
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 756           <====
007314  012742  000175                MOV     #175,-(R2)      ;MOVE TO MAILBOX #  *******  175  *******
007320  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007322  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.
                                ;*****************************************************************************
007324  005212                  TS107:  INC     (R2)            ;UPDATE TEST NUMBER
007326  022712  000107                  CMP     #107,(R2)       ;SEQUENCE ERROR?
007332  001021                          BNE     TS110-10        ;BR TO ERROR HALT ON SEQ ERROR
007334  005000                          CLR     R0              ;R0=0
007336  005010                          CLR     (R0)            ;LOC 0=0
007340  005110                          COM     (R0)            ;LOC 0=-1
007342  105100                          COMB    R0              ;R0=377
007344  000277                          SCC                     ;CC=0111
007346  000250                          CLN
007350  005770  000001                  TST     @1(R0)          ;TRY TST W/ MODE 7
007354  102402                          BVS     SNM7A           ;CHECK CC=1000
007356  101401                          BLOS    SNM7A
007360  100404                          BMI     SNM7B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
007362                          SNM7A:
007362  012742  000176                  MOV     #176,-(R2)      ;MOVE TO MAILBOX #  *******  176  *******
007366  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007370  000000                          HALT                    ;CC'S NOT CORRECT
007372  105100                  SNM7B:  COMB    R0              ;R0=0
007374  001404                          BEQ     TS110
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 756           <====
007376  012742  000177                  MOV     #177,-(R2)      ;MOVE TO MAILBOX #  *******  177  *******
007402  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007404  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.
                                ;*****************************************************************************
007406  005212                  TS110:  INC     (R2)            ;UPDATE TEST NUMBER
007410  022712  000110                  CMP     #110,(R2)       ;SEQUENCE ERROR?
007414  001006                          BNE     TS111-10        ;BR TO ERROR HALT ON SEQ ERROR
007416  005000                          CLR     R0              ;R0=0
007420  005100                          COM     R0              ;R0=-1
007422  005004                          CLR     R4              ;R4=0
007424  060004                          ADD     R0,R4           ;TRY ADD: R4=-1
007426  005204                          INC     R4              ;R4=0
007430  001404                          BEQ     TS111
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
007432  012742  000200                      MOV     #200,-(R2)      ;MOVE TO MAILBOX #  *******  200  *******
007436  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007440  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
                                ;*****************************************************************************
007442  005212                  TS111:  INC     (R2)            ;UPDATE TEST NUMBER
007444  022712  000111                  CMP     #111,(R2)       ;SEQUENCE ERROR?
007450  001006                          BNE     TS112-10        ;BR TO ERROR HALT ON SEQ ERROR
007452  005000                          CLR     R0              ;R0=0
007454  005004                          CLR     R4              ;R4=0
007456  005100                          COM     R0              ;R0=-1
007460  010004                          MOV     R0,R4           ;TRY MOVE -1 TO R4
007462  005204                          INC     R4              ;INC R4
007464  001404                          BEQ     TS112
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
007466  012742  000201                      MOV     #201,-(R2)      ;MOVE TO MAILBOX #  *******  201  *******
007472  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007474  000000                          HALT                    ;MOVE FAILED MODE 0 TO MODE 0
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
```

```
                                        H 5
3127                                    ;
3128                                    ;*****************************************************************************
3129                                    ;TEST 112       TEST SUB MODE 0,0
3130                                    ;*****************************************************************************
3131   007476  005212                   TS112:  INC     (R2)            ;UPDATE TEST NUMBER
3132   007500  022712  000112                   CMP     #112,(R2)       ;SEQUENCE ERROR?
3133   007504  001016                           BNE     TS113-10        ;BR TO ERROR HALT ON SEQ ERROR
3134   007506  005000                           CLR     R0              ;R0=0
3135   007510  005004                           CLR     R4              ;R4=0
3136   007512  005204                           INC     R4              ;R4=1
3137   007514  160400                           SUB     R4,R0           ;TRY SUB 0,0   R0=-1
3138   007516  100003                           BPL     SUB0            ;CC=1001
3139   007520  001402                           BEQ     SUB0
3140   007522  102401                           BVS     SUB0
3141   007524  103404                           BCS     SUB0A
3142                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3143                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3144                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3145                                                                    ;           WHICH FOLLOWS W/ 767           <====
3146   007526                           SUB0:
3147   007526  012742  000202                   MOV     #202,-(R2)      ;MOVE TO MAILBOX #  *******  202  *******
3148   007532  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3149   007534  000000                           HALT                    ;CONDITION CODE FAILED ON SUB
3150   007536  005200                   SUB0A:  INC     R0
3151   007540  001404                           BEQ     TS113
3152                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3153                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3154                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3155                                                                    ;           WHICH FOLLOWS W/ 761           <====
3156   007542  012742  000203                   MOV     #203,-(R2)      ;MOVE TO MAILBOX #  *******  203  *******
3157   007546  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3158   007550  000000                           HALT                    ;DATA RESULT OF SUB FAILED
3159                                                                    ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS.
                                ;WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
                                ;SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
                                ;BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
                                ;VERIFIED.
                                ;
                                ;*****************************************************************************
                                ;TEST 113       TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
                                ;*****************************************************************************
007552  005212                  TS113:  INC     (R2)            ;UPDATE TEST NUMBER
007554  022712  000113                  CMP     #113,(R2)       ;SEQUENCE ERROR?
007560  001051                          BNE     TS114-10        ;BR TO ERROR HALT ON SEQ ERROR
007562  005000                          CLR     R0              ;R0=0
007564  010004                          MOV     R0,R4           ;TRY MOVE MODE 0,0
007566  001404                          BEQ     DOP0A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 774           <====
007570  012742  000204                  MOV     #204,-(R2)      ;MOVE TO MAILBOX #  *******  204  *******
007574  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007576  000000                          HALT                    ;Z-BIT NOT SET
007600  005200                  DOP0A:  INC     R0              ;R0=1
007602  005100                          COM     R0              ;R0=177776
007604  005104                          COM     R4              ;R4=177777
007606  040004                          BIC     R0,R4           ;TRY BIC: R4=1
007610  005304                          DEC     R4              ;R4=0
007612  001404                          BEQ     DOP0B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
007614  012742  000205                  MOV     #205,-(R2)      ;MOVE TO MAILBOX #  *******  205  *******
007620  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007622  000000                          HALT                    ;BIC CLEAR RESULT INCORRECT
007624  050004                  DOP0B:  BIS     R0,R4           ;TRY BIS: R4=177777
007626  005204                          INC     R4
007630  005204                          INC     R4              ;R4=0
007632  001404                          BEQ     DOP0C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 752           <====
007634  012742  000206                  MOV     #206,-(R2)      ;MOVE TO MAILBOX #  *******  206  *******
007640  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007642  000000                          HALT                    ;RESULT OF BIS INCORRECT
007644  005000                  DOP0C:  CLR     R0              ;R0=0
007646  105100                          COMB    R0              ;R0=377
007650  005004                          CLR     R4              ;R4=0
007652  005104                          COM     R4              ;R4=177777
007654  040004                          BIC     R0,R4           ;R4=177400
007656  060004                          ADD     R0,R4           ;TRY ADD: R4=177777
007660  005204                          INC     R4              ;R4=0
```

```
3216  007662  001404                    BEQ     DOPOD
3217                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3218                                                ;           CONDITIONAL BRANCH INST. AND   <====
3219                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3220                                                ;           WHICH FOLLOWS W/ 736           <====
3221  007664  012742  000207              MOV     #207,-(R2)  ;MOVE TO MAILBOX #  *******  207  *******
3222  007670  005242                      INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
3223  007672  000000                      HALT                ;RESULT OF ADD INCORRECT
3224  007674  160004              DOPOD:  SUB     R0,R4       ;177401=R4
3225  007676  105404                      NEGB    R4          ;R4=177777
3226  007700  005204                      INC     R4          ;RD=0
3227  007702  001404                      BEQ     TS114
3228                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3229                                                ;           CONDITIONAL BRANCH INST. AND   <====
3230                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3231                                                ;           WHICH FOLLOWS W/ 726           <====
3232  007704  012742  000210              MOV     #210,-(R2)  ;MOVE TO MAILBOX #  *******  210  *******
3233  007710  005242                      INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
3234  007712  000000                      HALT                ;RESULT OF SUB INCORRECT
3235                                                ;  OR SEQUENCE ERROR
3236
```

```
3237                                ;*****************************************************************************
3238                                ;
3239                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
3240                                ;DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
3241                                ;
3242                                ;*****************************************************************************
3243                                ;TEST 114       TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
3244                                ;*****************************************************************************
3245  007714  005212               TS114:  INC     (R2)            ;UPDATE TEST NUMBER
3246  007716  022712  000114               CMP     #114,(R2)       ;SEQUENCE ERROR?
3247  007722  001024                       BNE     TS115-10        ;BR TO ERROR HALT ON SEQ ERROR
3248  007724  005000                       CLR     R0              ;R0=0
3249  007726  005010                       CLR     (R0)            ;LOC. 0=0
3250  007730  105110                       COMB    (R0)            ;LOC. 0=377
3251  007732  005220                       INC     (R0)+           ;LOC. 0=400   R0=2
3252  007734  005400                       NEG     R0              ;R0=-2
3253  007736  060037  000000               ADD     R0,@#0          ;TRY ADD 0,3;  LOC. 0=376
3254  007742  100403                       BMI     DOP03A          ;CC=0001?
3255  007744  001402                       BEQ     DOP03A
3256  007746  102401                       BVS     DOP03A
3257  007750  103404                       BCS     DOP03B
3258                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3259                                                               ;           CONDITIONAL BRANCH INST. AND   <====
3260                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
3261                                                               ;           WHICH FOLLOWS W/ 764           <====
3262  007752                        DOP03A:
3263  007752  012742  000211               MOV     #211,-(R2)      ;MOVE TO MAILBOX #  *******  211  *******
3264  007756  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3265  007760  000000                       HALT                    ;CC'S NOT SET CORRECTLY
3266  007762  105137  000000       DOP03B: COMB    @#0             ;LOC. 0=1
3267  007766  005337  000000               DEC     @#0             ;LOC. 0=0
3268  007772  001404                       BEQ     TS115
3269                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3270                                                               ;           CONDITIONAL BRANCH INST. AND   <====
3271                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
3272                                                               ;           WHICH FOLLOWS W/ 753           <====
3273  007774  012742  000212               MOV     #212,-(R2)      ;MOVE TO MAILBOX #  *******  212  *******
3274  010000  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3275  010002  000000                       HALT                    ;DATA RESULT INCORRECT
3276                                                               ;  OR SEQUENCE ERROR
```

```
3277                                    ;*****************************************************************************
3278                                    ;
3279                                    ;       THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
3280                                    ;R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY.  COMPARE INSTRUCTIONS ARE
3281                                    ;THEN EXECUTED AND CHECKED.  FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
3282                                    ;
3283                                    ;*****************************************************************************
3284                                    ;TEST 115       TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
3285                                    ;*****************************************************************************
3286  010004  005212                TS115:  INC     (R2)            ;UPDATE TEST NUMBER
3287  010006  022712  000115                CMP     #115,(R2)       ;SEQUENCE ERROR?
3288  010012  001042                        BNE     TS116-10        ;BR TO ERROR HALT ON SEQ ERROR
3289  010014  005000                        CLR     R0              ;R0=0
3290  010016  005004                        CLR     R4              ;R4=0
3291  010020  005204                        INC     R4              ;R4=1
3292  010022  020400                        CMP     R4,R0           ;TRY COMPARE R4 TO R0
3293  010024  003004                        BGT     DNM1
3294                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3295                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3296                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3297                                                                ;           WHICH FOLLOWS W/ 772           <====
3298  010026  012742  000213                MOV     #213,-(R2)      ;MOVE TO MAILBOX #  *******  213  *******
3299  010032  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3300  010034  000000                        HALT                    ;CC'S NOT CORRECT FOR CMP
3301  010036  020004                DNM1:   CMP     R0,R4           ;TRY COMPARE R0 TO R4
3302  010040  002404                        BLT     DNM2
3303                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3304                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3305                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3306                                                                ;           WHICH FOLLOWS W/ 764           <====
3307  010042  012742  000214                MOV     #214,-(R2)      ;MOVE TO MAILBOX #  *******  214  *******
3308  010046  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3309  010050  000000                        HALT                    ;CC'S NOT CORRECT FOR CMP
3310  010052  005200                DNM2:   INC     R0              ;R0=1
3311  010054  020400                        CMP     R4,R0           ;TRY COMPARE R4=1 TO R0=1
3312  010056  001404                        BEQ     DNM3
3313                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3314                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3315                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3316                                                                ;           WHICH FOLLOWS W/ 755           <====
3317  010060  012742  000215                MOV     #215,-(R2)      ;MOVE TO MAILBOX #  *******  215  *******
3318  010064  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3319  010066  000000                        HALT                    ;CC'S NOT CORRECT (Z=1) FOR CMP
3320  010070  005000                DNM3:   CLR     R0              ;R0=0
3321  010072  005100                        COM     R0              ;R0=177777
3322  010074  005004                        CLR     R4              ;R4=0
3323  010076  030004                        BIT     R0,R4           ;TRY BIT R0 TO R4
3324  010100  001404                        BEQ     DNM4
3325                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3326                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3327                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3328                                                                ;           WHICH FOLLOWS W/ 744           <====
3329  010102  012742  000216                MOV     #216,-(R2)      ;MOVE TO MAILBOX #  *******  216  *******
3330  010106  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3331  010110  000000                        HALT                    ;CC'S NOT CORRECT FOR BIT
3332  010112  005304                DNM4:   DEC     R4              ;R4=177777
```

```
3333  010114  030004                    BIT     R0,R4           ;TRY BIT AGAIN
3334  010116  100404                    BMI     TS116
3335                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3336                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3337                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3338                                                            ;           WHICH FOLLOWS W/ 735           <====
3339  010120  012742  000217              MOV     #217,-(R2)      ;MOVE TO MAILBOX #  *******  217  *******
3340  010124  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3341  010126  000000                      HALT                    ;CC'S NOT CORRECT FOR BIT
3342                                                            ;  OR SEQUENCE ERROR
3343                                ;*****************************************************************************
3344                                ;
3345                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
3346                                ;IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
3347                                ;
3348                                ;*****************************************************************************
3349                                ;TEST 116       TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.
3350                                ;*****************************************************************************
3351  010130  005212                TS116:  INC     (R2)            ;UPDATE TEST NUMBER
3352  010132  022712  000116               CMP     #116,(R2)       ;SEQUENCE ERROR?
3353  010136  001022                       BNE     TS117-10        ;BR TO ERROR HALT ON SEQ ERROR
3354  010140  005000                       CLR     R0              ;R0=0
3355  010142  005010                       CLR     (R0)            ;LOC. 0=0
3356  010144  005110                       COM     (R0)            ;LOC. 0=177777
3357  010146  005200                       INC     R0              ;R0=1
3358  010150  020037  000000               CMP     R0,@#0          ;TRY CMP MODE 0,3
3359  010154  100403                       BMI     DNM03A          ;CC=0001
3360  010156  001402                       BEQ     DNM03A
3361  010160  102401                       BVS     DNM03A
3362  010162  103404                       BCS     DNM03B
3363                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3364                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3365                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3366                                                            ;           WHICH FOLLOWS W/ 765           <====
3367  010164                        DNM03A:
3368  010164  012742  000220               MOV     #220,-(R2)      ;MOVE TO MAILBOX #  *******  220  *******
3369  010170  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3370  010172  000000                       HALT                    ;CC'S NOT SET CORRECTLY
3371  010174  005300                DNM03B: DEC     R0
3372  010176  001002                       BNE     DNM03C
3373  010200  005210                       INC     (R0)
3374  010202  001404                       BEQ     TS117
3375                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3376                                                            ;           CONDITIONAL BRANCH INST. AND   <====
3377                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
3378                                                            ;           WHICH FOLLOWS W/ 755           <====
3379  010204                        DNM03C:
3380  010204  012742  000221               MOV     #221,-(R2)      ;MOVE TO MAILBOX #  *******  221  *******
3381  010210  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3382  010212  000000                       HALT                    ;DATA INCORRECTLY MODIFIED BY CMP
3383                                                            ;  OR SEQUENCE ERROR
```

```
3384                                    ;*****************************************************************************
3385                                    ;
3386                                    ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS.  R0 IS SET TO -1
3387                                    ;AND LOC 0 TO 1.  R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.
3388                                    ;IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE
3389                                    ;RESULTS VERIFIED.
3390                                    ;
3391                                    ;*****************************************************************************
3392                                    ;TEST 117       TEST MODE 1 W/ DOP INST.
3393                                    ;*****************************************************************************
3394  010214  005212                    TS117:  INC     (R2)            ;UPDATE TEST NUMBER
3395  010216  022712  000117                    CMP     #117,(R2)       ;SEQUENCE ERROR?
3396  010222  001007                            BNE     TS120-10        ;BR TO ERROR HALT ON SEQ ERROR
3397  010224  005000                            CLR     R0              ;R0=0
3398  010226  005100                            COM     R0              ;R0=177777
3399  010230  005004                            CLR     R4              ;R4=0
3400  010232  005014                            CLR     (R4)            ;LOC 0=0
3401  010234  005214                            INC     (R4)            ;LOC 0=1
3402  010236  061400                            ADD     (R4),R0         ;TRY ADD SOURCE MODE 1
3403  010240  001404                            BEQ     TS120
3404                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3405                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3406                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3407                                                                    ;           WHICH FOLLOWS W/ 770           <====
3408  010242  012742  000222                    MOV     #222,-(R2)      ;MOVE TO MAILBOX #  *******  222  *******
3409  010246  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3410  010250  000000                            HALT                    ;RESULT OF ADD INCORRECT
3411                                                                    ;  OR SEQUENCE ERROR
```

```
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422  010252  005212
3423  010254  022712  000120
3424  010260  001007
3425  010262  005000
3426  010264  005010
3427  010266  005110
3428  010270  005004
3429  010272  151004
3430  010274  105104
3431  010276  001404
3432
3433
3434
3435
3436  010300  012742  000223
3437  010304  005242
3438  010306  000000
3439
```

```
;*****************************************************************************************
;
;       THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS
;EVEN BYTES.  LOC. 0 IS SET TO -1 AND R4 IS CLEARED.  THEN R4 IS
;SET TO -1 USING A BISB THRU R0 WITH MODE 1.
;
;*****************************************************************************************
;TEST 120       TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
;*****************************************************************************************
;S120:  INC     (R2)            ;UPDATE TEST NUMBER
        CMP     #120,(R2)       ;SEQUENCE ERROR?
        BNE     TS121-10        ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0              ;R0=0
        CLR     (R0)            ;LOC. 0=0
        COM     (R0)            ;LOC. 0=177777
        CLR     R4              ;R4=0
        BISB    (R0),R4         ;TRY MODE 1- EVEN BYTE W/ DOP
        COMB    R4              ;R4=0
        BEQ     TS121
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 770           <====
        MOV     #223,-(R2)      ;MOVE TO MAILBOX #  *******  223  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;RESULT OF BISB IS INCORRECT
                                ;  OR SEQUENCE ERROR
```

```
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451  010310  005212
3452  010312  022712  000121
3453  010316  001007
3454  010320  005000
3455  010322  005010
3456  010324  005110
3457  010326  005004
3458  010330  105104
3459  010332  121004
3460  010334  001404
3461
3462
3463
3464
3465  010336  012742  000224
3466  010342  005242
3467  010344  000000
3468
```

```
;*****************************************************************************************
;
;       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.
;*****************************************************************************************
TS121:  INC     (R2)            ;UPDATE TEST NUMBER
        CMP     #121,(R2)       ;SEQUENCE ERROR?
        BNE     TS122-10        ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0              ;R0=0
        CLR     (R0)            ;LOC 0=0
        COM     (R0)            ;LOC 0=177777
        CLR     R4              ;R4=0
        COMB    R4              ;R4=377
        CMPB    (R0),R4         ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
        BEQ     TS122
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 770           <====
        MOV     #224,-(R2)      ;MOVE TO MAILBOX #  *******  224  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;RESULT OF CMPB INCORRECT
                                ;  OR SEQUENCE ERROR
```

```
3469
3470                    ;*****************************************************************************************
3471                    ;
3472                    ;       THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
3473                    ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
3474                    ;R4 IS SET TO -1.  MOVB ARE USED TO MOVE BYTE 0 TO R4.  THIS
3475                    ;VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
3476                    ;FUNCTION WITH MODE 0.
3477                    ;       THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
3478                    ;THE LOGIC FOR COMPLEMENTARY DATA.
3479                    ;       THIS TEST EXERCISES UNIQUE MICROCODE.
3480                    ;
3481                    ;*****************************************************************************************
3482                    ;TEST 122       TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
3483                    ;*****************************************************************************************
3484  010346  005212    TS122:  INC     (R2)            ;UPDATE TEST NUMBER
3485  010350  022712    000122          CMP     #122,(R2)       ;SEQUENCE ERROR?
3486  010354  001020                    BNE     TS123-10        ;BR TO ERROR HALT ON SEQ ERROR
3487  010356  005000                    CLR     R0              ;R0=0
3488  010360  005010                    CLR     (R0)            ;LOC 0=0
3489  010362  105110                    COMB    (R0)            ;LOC 0=177400
3490  010364  005110                    COM     (R0)
3491  010366  005004                    CLR     R4              ;R4=0
3492  010370  005104                    COM     R4              ;R4=177777
3493  010372  111004                    MOVB    (R0),R4         ;R4=0
3494  010374  005704                    TST     R4              ;CHECK SIGN OF WORD
3495  010376  001404                    BEQ     DOP1
3496                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3497                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3498                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3499                                                    ;           WHICH FOLLOWS W/ 766           <====
3500  010400  012742    000225                  MOV     #225,-(R2)      ;MOVE TO MAILBOX #  *******  225  *******
3501  010404  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3502  010406  000000                    HALT                    ;MOVB SHOULD SIGN X-TEND
3503  010410  005110            DOP1:   COM     (R0)            ;LOC 0=177777
3504  010412  111004                    MOVB    (R0),R4         ;DO MOVB W/ EVEN BYTE
3505  010414  100404                    BMI     TS123
3506
3507                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3508                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3509                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3510  010416  012742    000226                  MOV     #226,-(R2)      ;MOVE TO MAILBOX #  *******  226  *******
3511  010422  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3512  010424  000000                    HALT                    ;MOVB SHOULD SIGN X-TEND
3513                                                    ;  OR SEQUENCE ERROR
```

```
3514
3515                                    ;*****************************************************************************
3516                                    ;
3517                                    ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE
3518                                    ;ODD BYTES.  LOC. 0 IS SET TO 177400.  R0 IS SET TO 0 AND R4 IS
3519                                    ;SET TO 1.  THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.
3520                                    ;THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.
3521                                    ;
3522                                    ;*****************************************************************************
3523                                    ;TEST 123       TEST MODE 1-ODD BYTE W/ DOP INSTS.
3524                                    ;*****************************************************************************
3525  010426  005212                   TS123:  INC     (R2)            ;UPDATE TEST NUMBER
3526  010430  022712  000123                   CMP     #123,(R2)       ;SEQUENCE ERROR?
3527  010434  001010                           BNE     TS124-10        ;BR TO ERROR HALT ON SEQ ERROR
3528  010436  005000                           CLR     R0              ;R0=0
3529  010440  005010                           CLR     (R0)            ;LOC. 0=0
3530  010442  005004                           CLR     R4              ;R4=0
3531  010444  005204                           INC     R4              ;R4=1
3532  010446  105114                           COMB    (R4)            ;LOC. 0=177400
3533  010450  151410                           BISB    (R4),(R0)       ;TRY TO BIS LOW ORDER BITS W/ MODE 1
3534  010452  005210                           INC     (R0)            ;CHECK RESULT
3535  010454  001404                           BEQ     TS124
3536                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3537                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
3538                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
3539                                                                   ;           WHICH FOLLOWS W/ 767           <====
3540  010456  012742  000227                   MOV     #227,-(R2)      ;MOVE TO MAILBOX #  *******  227  *******
3541  010462  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3542  010464  000000                           HALT                    ;RESULT OF BISB INCORRECT
3543                                                                   ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS.  LOC. 0 IS SET TO -1.
                                ;R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0
                                ;TO R7.  THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER
                                ;IS CHECKED.
                                ;
                                ;*****************************************************************************
                                ;TEST 124        TEST MODE 2 W/ DOP INSTS.
                                ;*****************************************************************************
010466  005212                TS124:  INC     (R2)            ;UPDATE TEST NUMBER
010470  022712  000124                CMP     #124,(R2)       ;SEQUENCE ERROR?
010474  001015                        BNE     TS125-10        ;BR TO ERROR HALT ON SEQ ERROR
010476  005000                        CLR     R0              ;R0=0
010500  005010                        CLR     (R0)            ;LOC. 0=0
010502  005110                        COM     (R0)            ;LOC. 0=177777
010504  012004                        MOV     (R0)+,R4        ;TRY MOVE MODE 2,0
010506  005204                        INC     R4              ;CHECK R4
010510  001404                        BEQ     DOP2
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 771           <====
010512  012742  000230                MOV     #230,-(R2)      ;MOVE TO MAILBOX #  *******  230  *******
010516  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010520  000000                        HALT                    ;RESULT OF MOV INST INCORRECT
010522  005300                DOP2:   DEC     R0              ;TEST R0 AFTER MODE 2
010524  005300                        DEC     R0
010526  001404                        BEQ     TS125
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 762           <====
010530  012742  000231                MOV     #231,-(R2)      ;MOVE TO MAILBOX #  *******  231  *******
010534  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010536  000000                        HALT                    ;REGISTER NOT INCREMENTED IN MODE 2
                                                            ;  OR SEQUENCE ERROR
```

```
3582
3583                                    ;*****************************************************************************
3584                                    ;
3585                                    ;       THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS
3586                                    ;EVEN BYTES.  LOC. 0 IS SET TO -1.  R0 IS CLEARED AND USED AS THE
3587                                    ;ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING
3588                                    ;BYTE 0 DATA AND A BICB.  UNIQUE IN THIS TEST IS USE OF THE
3589                                    ;SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION.  THE SOURCE AND
3590                                    ;DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.
3591                                    ;
3592                                    ;*****************************************************************************
3593                                    ;TEST 125       TEST MODE 2 - EVEN BYTE W/ DOP INST.
3594                                    ;*****************************************************************************
3595  010540  005212                   TS125:  INC     (R2)            ;UPDATE TEST NUMBER
3596  010542  022712  000125                   CMP     #125,(R2)       ;SEQUENCE ERROR?
3597  010546  001016                           BNE     TS126-10        ;BR TO ERROR HALT ON SEQ ERROR
3598  010550  005000                           CLR     R0              ;R0=0
3599  010552  010010                           MOV     R0,(R0)         ;LOC. 0=0
3600  010554  005110                           COM     (R0)            ;LOC. 0=177777
3601  010556  142010                           BICB    (R0)+,(R0)      ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
3602  010560  105737  000001                   TSTB    @#1             ;CHECK RESULT
3603  010564  001404                           BEQ     DOPB2A
3604                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3605                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3606                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3607                                                                    ;           WHICH FOLLOWS W/ 770           <====
3608  010566  012742  000232                   MOV     #232,-(R2)      ;MOVE TO MAILBOX #  *******  232  *******
3609  010572  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3610  010574  000000                           HALT                    ;BICB DESTINATION INCORRECT
3611  010576  105137  000000           DOPB2A: COMB    @#0             ;CHECK BICB SOURCE
3612  010602  001404                           BEQ     TS126
3613                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3614                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3615                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3616                                                                    ;           WHICH FOLLOWS W/ 761           <====
3617  010604  012742  000233                   MOV     #233,-(R2)      ;MOVE TO MAILBOX #  *******  233  *******
3618  010610  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3619  010612  000000                           HALT                    ;BICB SOURCE INCORRECTLY CHANGED
3620                                                                    ;  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 PROPER BYTE WAS MOVED AND SIGN X-TENDED.
                                ;
                                ;*****************************************************************************
                                ;TEST 126       TEST MODE 2 - ODD BYTE W/ DOP INST.
                                ;*****************************************************************************
010614  005212                  TS126:  INC     (R2)            ;UPDATE TEST NUMBER
010616  022712  000126                  CMP     #126,(R2)       ;SEQUENCE ERROR?
010622  001017                          BNE     TS127-10        ;BR TO ERROR HALT ON SEQ ERROR
010624  005000                          CLR     R0              ;R0=0
010626  005004                          CLR     R4              ;R4=0
010630  005010                          CLR     (R0)            ;LOC. 0=0
010632  005110                          COM     (R0)            ;LOC. 0=177777
010634  105120                          COMB    (R0)+           ;LOC 0=177400; R0=1
010636  112004                          MOVB    (R0)+,R4        ;TRY DOP MODE 2 W/ ODD BYTE
010640  005204                          INC     R4              ;CHECK RESULT OF MOVB
010642  001404                          BEQ     DOPB2B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
010644  012742  000234                  MOV     #234,-(R2)      ;MOVE TO MAILBOX #  *******  234  *******
010650  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010652  000000                          HALT                    ;RESULT OF MOVB INCORRECT
010654  005740                  DOPB2B: TST     -(R0)           ;BUMP R0 DOWN BY 2
010656  005700                          TST     R0              ;CHECK R0
010660  001404                          BEQ     TS127
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
010662  012742  000235                  MOV     #235,-(R2)      ;MOVE TO MAILBOX #  *******  235  *******
010666  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010670  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.
                                        ;*************************************************************************
010672  005212                          TS127:  INC     (R2)            ;UPDATE TEST NUMBER
010674  022712  000127                          CMP     #127,(R2)       ;SEQUENCE ERROR?
010700  001011                                  BNE     TS130-10        ;BR TO ERROR HALT ON SEQ ERROR
010702  012737  052525  000000                  MOV     #052525,@#0     ;MOVE 52525 TO LOC. 0
010710  012700  125252                          MOV     #125252,R0      ;SET ALT. ONE AND ZERO IN R0
010714  053700  000000                          BIS     @#0,R0          ;TRY TO SET ALL OTHER BITS W/ MODE 3
010720  005200                                  INC     R0              ;TEST RESULT
010722  001404                                  BEQ     TS130
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                        ;           WHICH FOLLOWS W/ 766            <====
010724  012742  000236                          MOV     #236,-(R2)      ;MOVE TO MAILBOX #  *******  236  *******
010730  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010732  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.
                                        ;*************************************************************************
010734  005212                          TS130:  INC     (R2)            ;UPDATE TEST NUMBER
010736  022712  000130                          CMP     #130,(R2)       ;SEQUENCE ERROR?
010742  001011                                  BNE     TS131-10        ;BR TO ERROR HALT ON SEQ ERROR
010744  012737  052652  000000                  MOV     #52652,@#0      ;NOVE 1'S AND 0' PATTERN TO LOC. 0
010752  005000                                  CLR     R0              ;R0=0
010754  153700  000000                          BISB    @#0,R0          ;TRY R0=252 W/ MODE 3 - EVEN BYTE
010760  022700  000252                          CMP     #252,R0         ;BISB W/ EVEN BYTE SUCCESSFUL?
010764  001404                                  BEQ     TS131
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                        ;           WHICH FOLLOWS W/ 766            <====
010766  012742  000237                          MOV     #237,-(R2)      ;MOVE TO MAILBOX #  *******  237  *******
010772  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010774  000000                                  HALT                    ;BISB W/ MODE 3 - EVEN BYTE FAILED
                                                                        ;  OR SEQUENCE ERROR
```

```
3713                                    ;*****************************************************************************************
3714                                    ;
3715                                    ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
3716                                    ;WHICH ADDRESS ODD BYTES.  THE SAME PROCEDURE USED IN PREVIOUS
3717                                    ;TEST IS USED HERE.  THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
3718                                    ;THE EXPECTED RESULT IS:  R0 = 125.
3719                                    ;
3720                                    ;*****************************************************************************************
3721                                    ;TEST 131       TEST MODE 3 - ODD BYTE W/ DOP INSTS.
3722                                    ;*****************************************************************************************
3723  010776  005212                    TS131:  INC     (R2)            ;UPDATE TEST NUMBER
3724  011000  022712  000131                    CMP     #131,(R2)       ;SEQUENCE ERROR?
3725  011004  001011                            BNE     TS132-10        ;BR TO ERROR HALT ON SEQ ERROR
3726  011006  012737  052652  000000           MOV     #52652,@#0      ;MOVE 1'S AND 0'S PATTERN TO LOC 0
3727  011014  005000                            CLR     R0              ;R0=0
3728  011016  153700  000001                   BISB    @#1,R0          ;TRY R0=152 W/ MODE 3 - ODD BYTE
3729  011022  022700  000125                   CMP     #125,R0         ;R0=125?
3730  011026  001404                            BEQ     TS132
3731                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3732                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3733                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3734                                                                    ;           WHICH FOLLOWS W/ 766           <====
3735  011030  012742  000240                   MOV     #240,-(R2)      ;MOVE TO MAILBOX #  *******  240  *******
3736  011034  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3737  011036  000000                            HALT                    ;BISB W/ MODE 3 - ODD BYTE FAILED
3738                                                                    ;  OR SEQUENCE ERROR
3739
3740                                    ;*****************************************************************************************
3741                                    ;TEST 132       TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
3742                                    ;*****************************************************************************************
3743  011040  005212                    TS132:  INC     (R2)            ;UPDATE TEST NUMBER
3744  011042  022712  000132                    CMP     #132,(R2)       ;SEQUENCE ERROR?
3745  011046  001017                            BNE     TS133-10        ;BR TO ERROR HALT ON SEQ ERROR
3746  011050  005000                            CLR     R0              ;R0=0
3747  011052  105100                            COMB    R0              ;R0=377
3748  011054  000263                            +SEC!SEV                ;SET C AND V BITS
3749  011056  132700  000200                   BITB    #200,R0         ;TRY DOPNM DEST. MODE 0-BYTE
3750  011062  001403                            BEQ     DNMB0A          ;BR TO ERROR IF Z BIT SET
3751  011064  102402                            BVS     DNMB0A          ;BR TO ERROR IF V BIT SET
3752  011066  103001                            BCC     DNMB0A          ;BR TO ERROR IF C BIT CLEAR.
3753  011070  100404                            BMI     DNMB0B
3754                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3755                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3756                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3757                                                                    ;           WHICH FOLLOWS W/ 766           <====
3758  011072                            DNMB0A:
3759  011072  012742  000241                   MOV     #241,-(R2)      ;MOVE TO MAILBOX #  *******  241  *******
3760  011076  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3761  011100  000000                            HALT                    ;CC'S INCORRECT
3762  011102  105100                    DNMB0B: COMB    R0              ;CHECK DESTINATION DATA
3763  011104  001404                            BEQ     TS133
3764                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3765                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
3766                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
3767                                                                    ;           WHICH FOLLOWS W/ 760           <====
3768  011106  012742  000242                   MOV     #242,-(R2)      ;MOVE TO MAILBOX #  *******  242  *******
```

```
                                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                                        HALT                    ;DEST. DATA MODIFIED
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 133       TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
                                ;*******************************************************************************
                                TS133:  INC     (R2)            ;UPDATE TEST NUMBER
                                        CMP     #133,(R2)       ;SEQUENCE ERROR?
                                        BNE     TS134-10        ;BR TO ERROR HALT ON SEQ ERROR
                                        CLR     R0              ;R0=0
                                        CLR     (R0)            ;LOC. 0=0
                                        CLC                     ;CLEAR C BIT
                                        BIT     #177777,(R0)    ;TRY DOPNM DEST. MODE 1
                                        BMI     DNM1A           ;BR TO ERROR IF N BIT SET
                                        BVS     DNM1A           ;BR TO ERROR IF V BIT SET
                                        BCS     DNM1A           ;BR TO ERROR IF C BIT SET
                                        BEQ     DNM1B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
011150                          DNM1A:
011150  012742  000243                MOV     #243,-(R2)      ;MOVE TO MAILBOX #  *******  243  *******
011154  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011156  000000                        HALT                    ;COND. CODES INCORRECT
011160  005710                DNM1B:  TST     (R0)            ;CHECK TEST DATA
011162  001404                        BEQ     TS134
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
011164  012742  000244                MOV     #244,-(R2)      ;MOVE TO MAILBOX #  *******  244  *******
011170  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011172  000000                        HALT                    ;DESTINATION DATA MODIFIED
                                                                ;  OR SEQUENCE ERROR

                                ;*******************************************************************************
                                ;TEST 134       TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
                                ;*******************************************************************************
011174  005212                  TS134:  INC     (R2)            ;UPDATE TEST NUMBER
011176  022712  000134                CMP     #134,(R2)       ;SEQUENCE ERROR?
011202  001027                        BNE     TS135-10        ;BR TO ERROR HALT ON SEQ ERROR
011204  005000                        CLR     R0              ;R0=0
011206  005010                        CLR     (R0)            ;LOC. 0=0
011210  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125252
011214  032720  077777                BIT     #77777,(R0)+    ;TRY DOPNM INST W/ MODE 2
011220  102402                        BVS     DNM2A           ;BR TO ERROR IF V BIT SET
011222  001401                        BEQ     DNM2A           ;BR TO ERROR IF Z-BIT SET
011224  100004                        BPL     DNM2B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
011226                          DNM2A:
011226  012742  000245                MOV     #245,-(R2)      ;MOVE TO MAILBOX #  *******  245  *******
```

```
                                        L 6
3825  011232  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3826  011234  000000                        HALT                    ;COND. CODES INCORRECT
3827  011236  005300              DNM2B:    DEC     R0              ;DECREMENT R0 TO CHECK IT.
3828  011240  005300                        DEC     R0
3829  011242  001404                        BEQ     DNM2D
3830                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3831                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3832                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3833                                                                ;           WHICH FOLLOWS W/ 757           <====
3834  011244                        DNM2C:
3835  011244  012742  000246                MOV     #246,-(R2)      ;MOVE TO MAILBOX #  *******  246  *******
3836  011250  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3837  011252  000000                        HALT                    ;MODE 2 REGISTER NOT INCREMENTED BY 2
3838  011254  022710  125252      DNM2D:    CMP     #125252,(R0)    ;CHECK DEST. DATA
3839  011260  001404                        BEQ     TS135
3840                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3841                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3842                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3843                                                                ;           WHICH FOLLOWS W/ 750           <====
3844  011262  012742  000247                MOV     #247,-(R2)      ;MOVE TO MAILBOX #  *******  247  *******
3845  011266  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3846  011270  000000                        HALT                    ;DEST. DATA MODIFIED
3847                                                                ;  OR SEQUENCE ERROR
3848
3849                                    ;***************************************************************************************
3850                                    ;TEST 135       TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
3851                                    ;***************************************************************************************
3852  011272  005212              TS135:    INC     (R2)            ;UPDATE TEST NUMBER
3853  011274  022712  000135                CMP     #135,(R2)       ;SEQUENCE ERROR?
3854  011300  001051                        BNE     TS136-10        ;BR TO ERROR HALT ON SEQ ERROR
3855  011302  005000                        CLR     R0              ;R0=0
3856  011304  005010                        CLR     (R0)            ;LOC. 0=0
3857  011306  052710  052652              BIS     #52652,(R0)     ;LOC. 0=52652
3858  011312  000263                        +SEC!SEV                ;SET C AND V BITS
3859  011314  132720  000201              BITB    #201,(R0)+      ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
3860  011320  001403                        BEQ     DNMB2A          ;BR TO ERROR IF Z-BIT SET
3861  011322  103002                        BCC     DNMB2A          ;BR TO ERROR IF C-BIT CLEAR
3862  011324  102401                        BVS     DNMB2A          ;BR TO ERROR IF V-BIT SET
3863  011326  100404                        BMI     DNMB2B
3864                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3865                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3866                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3867                                                                ;           WHICH FOLLOWS W/ 764           <====
3868  011330                        DNMB2A:
3869  011330  012742  000250                MOV     #250,-(R2)      ;MOVE TO MAILBOX #  *******  250  *******
3870  011334  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3871  011336  000000                        HALT                    ;COND. CODES INCORRECT
3872  011340  005300              DNMB2B:   DEC     R0              ;CHECK DEST. REGISTER.
3873  011342  001404                        BEQ     DNMB2C
3874                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3875                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3876                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3877                                                                ;           WHICH FOLLOWS W/ 756           <====
3878  011344  012742  000251                MOV     #251,-(R2)      ;MOVE TO MAILBOX #  *******  251  *******
3879  011350  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3880  011352  000000                        HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
```

```
3881  011354  005200                DNMB2C: INC     R0              ;R0=1
3882  011356  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/MODE 2-ODD BYTE
3883  011362  001402                        BEQ     DNMB2D  ;BR TO ERROR IF Z-BIT SET
3884  011364  102401                        BVS     DNMB2D          ;BR TO ERROR IF V-BIT SET
3885  011366  100004                        BPL     DNMB2E
3886                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3887                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3888                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3889                                                                ;           WHICH FOLLOWS W/ 744           <====
3890  011370                        DNMB2D:
3891  011370  012742  000252                MOV     #252,-(R2)      ;MOVE TO MAILBOX #  *******  252  *******
3892  011374  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3893  011376  000000                        HALT                    ;COND. CODES INCORRECT
3894  011400  005300                DNMB2E: DEC     R0              ;DEC R0 TO CHECK IT.
3895  011402  005300                        DEC     R0
3896  011404  001404                        BEQ     DNMB2F
3897                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3898                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3899                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3900                                                                ;           WHICH FOLLOWS W/ 735           <====
3901  011406  012742  000253                MOV     #253,-(R2)      ;MOVE TO MAILBOX #  *******  253  *******
3902  011412  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3903  011414  000000                        HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
3904  011416  022710  052652        DNMB2F: CMP     #52652,(R0)     ;CHECK DEST. DATA IS UNMODIFIED
3905  011422  001404                        BEQ     TS136
3906                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3907                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3908                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
3909                                                                ;           WHICH FOLLOWS W/ 726           <====
3910  011424  012742  000254                MOV     #254,-(R2)      ;MOVE TO MAILBOX #  *******  254  *******
3911  011430  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3912  011432  000000                        HALT                    ;DEST. DATA WAS MODIFIED.
3913                                                                ;  OR SEQUENCE ERROR
3914
3915
3916                                ;*****************************************************************************************
3917                                ;TEST 136       TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
3918                                ;*****************************************************************************************
3919  011434  005212                TS136:  INC     (R2)            ;UPDATE TEST NUMBER
3920  011436  022712  000136                CMP     #136,(R2)       ;SEQUENCE ERROR?
3921  011442  001050                        BNE     TS137-10        ;BR TO ERROR HALT ON SEQ ERROR
3922  011444  005000                        CLR     R0              ;R0=0
3923  011446  005010                        CLR     (R0)            ;LOC. 0=0
3924  011450  052710  125125                BIS     #125125,(R0)    ;LOC. 0=125125
3925  011454  105100                        COMB    R0              ;R0=377
3926  011456  005200                        INC     R0              ;R0=400
3927  011460  005010                        CLR     (R0)            ;LOC. 400=0
3928  011462  000263                        +SEC!SEV                ;C-BIT=V-BIT=1
3929  011464  132730  000201                BITB    #201,@(R0)+     ;TRY DOPNM W/MODE 3-EVEN BYTE
3930  011470  001403                        BEQ     DNMB3A          ;BR TO ERROR IF Z BIT SET
3931  011472  102402                        BVS     DNMB3A          ;BR TO ERROR IF V BIT SET
3932  011474  103001                        BCC     DNMB3A          ;BR TO ERROR IF C BIT CLEAR
3933  011476  100004                        BPL     DNMB3B
3934                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3935                                                                ;           CONDITIONAL BRANCH INST. AND   <====
3936                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                ;           WHICH FOLLOWS W/ 761          <====
011500                          DNMB3A:
011500  012742  000255                MOV     #255,-(R2)      ;MOVE TO MAILBOX #  *******  255  *******
011504  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011506  000000                        HALT                    ;COND. CODES INCORRECT
011510  022700  000402        DNMB3B: CMP     #402,R0         ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
011514  001404                        BEQ     DNMB3C
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 752           <====
011516  012742  000256                MOV     #256,-(R2)      ;MOVE TO MAILBOX #  *******  256  *******
011522  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011524  000000                        HALT                    ;DEST. REGISTER NOT INCREMENTED BY 2
011526  005200        DNMB3C: INC     R0                      ;R0=404
011530  005200                INC     R0
011532  132730  000201        BITB    #201,@(R0)+     ;TRY DOPNM DEST MODE 3-BYTE(ODD)
011536  001402                BEQ     DNMB3D          ;BR TO ERROR IF Z BIT SET
011540  102401                BVS     DNMB3D          ;BR TO ERROR IF V BIT SET
011542  100404                BMI     DNMB3E
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 737           <====
011544                          DNMB3D:
011544  012742  000257                MOV     #257,-(R2)      ;MOVE TO MAILBOX #  *******  257  *******
011550  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011552  000000                        HALT                    ;COND. CODES INCORRECT
011554  005004        DNMB3E: CLR     R4                      ;R4=0
011556  022714  125125                CMP     #125125,(R4)    ;CHECK DEST. DATA
011562  001404                        BEQ     TS137
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 727           <====
011564  012742  000260                MOV     #260,-(R2)      ;MOVE TO MAILBOX #  *******  260  *******
011570  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011572  000000                        HALT                    ;DEST. DATA MODIFIED
                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 137       TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
                        ;*****************************************************************************
011574  005212          TS137:  INC     (R2)            ;UPDATE TEST NUMBER
011576  022712  000137                CMP     #137,(R2)       ;SEQUENCE ERROR?
011602  001033                        BNE     TS140-10        ;BR TO ERROR HALT ON SEQ ERROR
011604  005000                        CLR     R0              ;R0=0
011606  005010                        CLR     (R0)            ;LOC. 0=0
011610  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125125
011614  052700  000002                BIS     #2,R0           ;R0=2
011620  000277                        SCC                     ;SET ALL COND. CODE BITS
011622  032740  020000                BIT     #20000,-(R0)    ;TRY DOPNM W/ MODE 4
011626  100403                        BMI     DNM4A           ;BR TO ERROR IF N-BIT SET
011630  102402                        BVS     DNM4A           ;BR TO ERROR IF V-BIT SET
011632  103001                        BCC     DNM4A           ;BR TO ERROR IF C-BIT CHAR
011634  001004                        BNE     DNM4B
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
011636                          DNM4A:
011636  012742  000261                MOV     #261,-(R2)      ;MOVE TO MAILBOX #  *******  261  *******
011642  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011644  000000                        HALT                    ;COND. CODES INCORRECT
011646  005700                DNM4B:  TST     R0              ;CHECK DEST. REGISTER
011650  001404                        BEQ     DNM4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
011652  012742  000262                MOV     #262,-(R2)      ;MOVE TO MAILBOX #  *******  262  *******
011656  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011660  000000                        HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
011662  022737  125252  000000  DNM4C: CMP     #125252,@#0     ;CHECK DEST. DATA
011670  001404                        BEQ     TS140
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 744           <====
011672  012742  000263                MOV     #263,-(R2)      ;MOVE TO MAILBOX #  *******  263  *******
011676  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011700  000000                        HALT                    ;DEST. DATA MODIFIED
                                                        ;  OR SEQUENCE ERROR

                        ;*******************************************************************************************
                        ;TEST 140       TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
                        ;*******************************************************************************************
011702  005212          TS140:  INC     (R2)            ;UPDATE TEST NUMBER
011704  022712  000140          CMP     #140,(R2)       ;SEQUENCE ERROR?
011710  001051                  BNE     TS141-10        ;BR TO ERROR HALT ON SEQ ERROR
011712  005000                  CLR     R0              ;R0=0
011714  005010                  CLR     (R0)            ;LOC. 0=0
011716  052710  052652          BIS     #52652,(R0)     ;LOC. 0=52652
011722  052700  000002          BIS     #2,R0           ;R0=2
011726  000257                  CCC                     ;COND. CODES=0
011730  132740  000201          BITB    #201,-(R0)      ;TRY DOPNM INST W/MODE 4 ODD BYTE
011734  102403                  BVS     DNMB4A          ;BR TO ERROR IF V BIT SET
011736  001402                  BEQ     DNMB4A  ;BR TO ERROR IF Z BIT SET
011740  103401                  BCS     DNMB4A          ;BR TO ERROR IF C BIT SET
011742  001004                  BNE     DNMB4B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
011744          DNMB4A:
011744  012742  000264                MOV     #264,-(R2)      ;MOVE TO MAILBOX #  *******  264  *******
011750  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011752  000000                        HALT                    ;COND. CODES INCORRECT
011754  022700  000001  DNMB4B: CMP     #1,R0           ;CHECK DEST. REGISTER
011760  001404                        BEQ     DNMB4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
```

```
4049                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4050                                                    ;           WHICH FOLLOWS W/ 753           <====
4051  011762  012742  000265          MOV     #265,-(R2)      ;MOVE TO MAILBOX #  *******  265  *******
4052  011766  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4053  011770  000000                  HALT                    ;DEST REG. NOT DECREMENTED BY 1
4054  011772  132740  000201  DNMB4C: BITB    #201,-(R0)      ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
4055  011776  001401                  BEQ     DNMB4D          ;BR TO ERROR IF Z-BIT SET
4056  012000  100404                  BMI     DNMB4E
4057                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4058                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4059                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4060                                                    ;           WHICH FOLLOWS W/ 743           <====
4061  012002                  DNMB4D:
4062  012002  012742  000266          MOV     #266,-(R2)      ;MOVE TO MAILBOX #  *******  266  *******
4063  012006  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4064  012010  000000                  HALT                    ;COND. CODES INCORRECT
4065  012012  005700          DNMB4E: TST     R0              ;CHECK DEST. REGISTER
4066  012014  001404                  BEQ     DNMB4F
4067                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4068                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4069                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4070                                                    ;           WHICH FOLLOWS W/ 735           <====
4071  012016  012742  000267          MOV     #267,-(R2)      ;MOVE TO MAILBOX #  *******  267  *******
4072  012022  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4073  012024  000000                  HALT                    ;DEST. REG. NOT DECREMENTED BY 1
4074  012026  022710  052652  DNMB4F: CMP     #52652,(R0)     ;CHECK DESTINATION DATA
4075  012032  001404                  BEQ     TS141
4076                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4077                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4078                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4079                                                    ;           WHICH FOLLOWS W/ 726           <====
4080  012034  012742  000270          MOV     #270,-(R2)      ;MOVE TO MAILBOX #  *******  270  *******
4081  012040  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4082  012042  000000                  HALT                    ;DEST. DATA MODIFIED
4083                                                    ;  OR SEQUENCE ERROR
4084
4085                    ;*****************************************************************************
4086                    ;TEST 141       TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
4087                    ;*****************************************************************************
4088  012044  005212  TS141:  INC     (R2)            ;UPDATE TEST NUMBER
4089  012046  022712  000141          CMP     #141,(R2)       ;SEQUENCE ERROR?
4090  012052  001034                  BNE     TS142-10        ;BR TO ERROR HALT ON SEQ ERROR
4091  012054  005000                  CLR     R0              ;R0=0
4092  012056  005010                  CLR     (R0)            ;LOC 0=0
4093  012060  052710  100000          BIS     #100000,(R0)    ;LOC. 0=100000
4094  012064  052700  000402          BIS     #402,R0         ;R0=2
4095  012070  000277                  SCC                     ;SET ALL COND. CODE BITS
4096  012072  032750  100000          BIT     #100000,@-(R0)  ;TRY DOPNM W/MODE 5
4097  012076  102403                  BVS     DNM5A           ;BR TO ERROR IF V-BIT SET
4098  012100  103002                  BCC     DNM5A           ;BR TO ERROR IF C-BIT CLEAR
4099  012102  001401                  BEQ     DNM5A           ;BR TO ERROR IF Z-BIT SET
4100  012104  100404                  BMI     DNM5B
4101                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4102                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4103                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4104                                                    ;           WHICH FOLLOWS W/ 762           <====
```

```
4105  012106                                DNM5A:
4106  012106  012742  000271                      MOV     #271,-(R2)      ;MOVE TO MAILBOX #  *******  271  *******
4107  012112  005242                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4108  012114  000000                              HALT                    ;COND. CODES INCORRECT
4109  012116  022700  000400              DNM5B:  CMP #400,R0             ;CHECK DEST. REGISTER
4110  012122  001404                              BEQ     DNM5C
4111                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4112                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
4113                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
4114                                                                      ;           WHICH FOLLOWS W/ 753           <====
4115  012124  012742  000272                      MOV     #272,-(R2)      ;MOVE TO MAILBOX #  *******  272  *******
4116  012130  005242                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4117  012132  000000                              HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
4118  012134  022737  100000  000000  DNM5C:  CMP     #100000,@#0     ;CHECK DESTINATION DATA
4119  012142  001404                              BEQ     TS142
4120                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4121                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
4122                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
4123                                                                      ;           WHICH FOLLOWS W/ 743           <====
4124  012144  012742  000273                      MOV     #273,-(R2)      ;MOVE TO MAILBOX #  *******  273  *******
4125  012150  005242                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4126  012152  000000                              HALT                    ;DEST. DATA INCORRECTLY MODIFIED
4127                                                                      ;  OR SEQUENCE ERROR
4128
4129                                  ;*************************************************************************************
4130                                  ;TEST 142       TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
4131                                  ;*************************************************************************************
4132  012154  005212                  TS142:  INC     (R2)            ;UPDATE TEST NUMBER
4133  012156  022712  000142                  CMP     #142,(R2)       ;SEQUENCE ERROR?
4134  012162  001033                          BNE     TS143-10        ;BR TO ERROR HALT ON SEQ ERROR
4135  012164  005000                          CLR     R0              ;R0=0
4136  012166  005010                          CLR     (R0)            ;LOC> 0=0
4137  012170  052710  000001                BIS     #1,(R0)         ;LOC. 0=1
4138  012174  005100                          COM     R0              ;R0=-1 C-BIT=1
4139  012176  032760  000001  000001        BIT     #1,1(R0)        ;TRY DOPNM W/MODE 6
4140  012204  001403                          BEQ     DNM6A           ;BR TO ERROR IF Z-BIT SET
4141  012206  102402                          BVS     DNM6A           ;BR TO ERROR IF V-BIT SET
4142  012210  103001                          BCC     DNM6A           ;BR TO ERROR IF C-BIT CLEAR
4143  012212  100004                          BPL     DNM6B
4144                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4145                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
4146                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
4147                                                                      ;           WHICH FOLLOWS W/ 763           <====
4148  012214                          DNM6A:
4149  012214  012742  000274                  MOV     #274,-(R2)      ;MOVE TO MAILBOX #  *******  274  *******
4150  012220  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4151  012222  000000                          HALT                    ;COND CODES INCORRECT
4152  012224  022700  177777          DNM6B:  CMP     #-1,R0          ;CHECK DEST. REGISTER
4153  012230  001404                          BEQ     DNM6C
4154                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4155                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
4156                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
4157                                                                      ;           WHICH FOLLOWS W/ 754           <====
4158  012232  012742  000275                  MOV     #275,-(R2)      ;MOVE TO MAILBOX #  *******  275  *******
4159  012236  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4160  012240  000000                          HALT                    ;DEST. REGISTER MODIFIED
```

```
4161  012242  022737  000001  000000  DNM6C:  CMP     #1,@#0          ;CHECK DEST. DATA
4162  012250  001404                  BEQ     TS143
4163                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4164                                            ;           CONDITIONAL BRANCH INST. AND   <====
4165                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4166                                            ;           WHICH FOLLOWS W/ 744           <====
4167  012252  012742  000276                  MOV     #276,-(R2)      ;MOVE TO MAILBOX #  *******  276  *******
4168  012256  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4169  012260  000000                  HALT                    ;DEST. DATA MODIFIED
4170                                                          ;  OR SEQUENCE ERROR
4171
4172                          ;*****************************************************************************
4173                          ;TEST 143       TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
4174                          ;*****************************************************************************
4175  012262  005212          TS143:  INC     (R2)            ;UPDATE TEST NUMBER
4176  012264  022712  000143          CMP     #143,(R2)       ;SEQUENCE ERROR?
4177  012270  001034                  BNE     TS144-10        ;BR TO ERROR HALT ON SEQ ERROR
4178  012272  005000                  CLR     R0              ;R0=0
4179  012274  005010                  CLR     (R0)            ;LOC. 0=0 C-BIT=0
4180  012276  052710  125125          BIS     #125125,(R0)    ;LOC. 0=125125
4181  012302  052700  000001          BIS     #1,R0           ;R0=1
4182  012306  132770  000125  000403  BITB    #125,@403(R0)   ;TRY DOPNM W/MODE 7
4183  012314  102403                  BVS     DNM7A           ;BR TO ERROR IF V-BIT SET
4184  012316  100402                  BMI     DNM7A           ;BR TO ERROR IF N-BIT SET
4185  012320  103401                  BCS     DNM7A           ;BR TO ERROR IF C-BIT SET
4186  012322  001404                  BEQ     DNM7B
4187                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4188                                            ;           CONDITIONAL BRANCH INST. AND   <====
4189                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4190                                            ;           WHICH FOLLOWS W/ 762           <====
4191  012324          DNM7A:
4192  012324  012742  000277                  MOV     #277,-(R2)      ;MOVE TO MAILBOX #  *******  277  *******
4193  012330  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4194  012332  000000                  HALT                    ;COND. CODES INCORRECT
4195  012334  022700  000001  DNM7B:  CMP     #1,R0           ;CHECK DEST. REGISTER
4196  012340  001404                  BEQ     DNM7C
4197                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4198                                            ;           CONDITIONAL BRANCH INST. AND   <====
4199                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4200                                            ;           WHICH FOLLOWS W/ 753           <====
4201  012342  012742  000300                  MOV     #300,-(R2)      ;MOVE TO MAILBOX #  *******  300  *******
4202  012346  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4203  012350  000000                  HALT                    ;DESTINATION REGISTER MODIFIED
4204  012352  022737  125125  000000  DNM7C:  CMP     #125125,@#0     ;CHECK DEST. DATA
4205  012360  001404                  BEQ     TS144
4206                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4207                                            ;           CONDITIONAL BRANCH INST. AND   <====
4208                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4209                                            ;           WHICH FOLLOWS W/ 743           <====
4210  012362  012742  000301                  MOV     #301,-(R2)      ;MOVE TO MAILBOX #  *******  301  *******
4211  012366  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4212  012370  000000                  HALT                    ;DEST. DATA INCORRECT
4213                                                          ;  OR SEQUENCE ERROR
4214
4215                          ;*****************************************************************************
4216                          ;
```

```
                                ;       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
                                ;*****************************************************************************
012372  005212                  TS144:  INC     (R2)            ;UPDATE TEST NUMBER
012374  022712  000144                  CMP     #144,(R2)       ;SEQUENCE ERROR?
012400  001016                          BNE     TS145-10        ;BR TO ERROR HALT ON SEQ ERROR
012402  005000                          CLR     R0              ;R0=0
012404  005010                          CLR     (R0)            ;LOC. 0=0
012406  005100                          COM     R0              ;R0=-1
012410  005004                          CLR     R4              ;R4 POINTS TO LOC. 0
012412  010014                          MOV     R0,(R4)         ;TRY MOVE MODE 0,1
012414  102402                          BVS     MDM1A           ;BR TO ERROR IF V SET
012416  001401                          BEQ     MDM1A           ;BR TO ERROR IF Z SET
012420  100404                          BMI     MDM1B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
012422                          MDM1A:
012422  012742  000302                  MOV     #302,-(R2)      ;MOVE TO MAILBOX #  *******  302  *******
012426  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012430  000000                          HALT                    ;CONDITION CODE NOT CORRECT
012432  005704                  MDM1B:  TST     R4
012434  001404                          BEQ     TS145
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 761           <====
012436  012742  000303                  MOV     #303,-(R2)      ;MOVE TO MAILBOX #  *******  303  *******
012442  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012444  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
                                ;*****************************************************************************
012446  005212                  TS145:  INC     (R2)            ;UPDATE TEST NUMBER
012450  022712  000145                  CMP     #145,(R2)       ;SEQUENCE ERROR?
012454  001026                          BNE     TS146-10        ;BR TO ERROR HALT ON SEQ ERROR
012456  005000                          CLR     R0              ;R0=0
012460  005001                          CLR     R1              ;R1=0
012462  005010                          CLR     (R0)            ;LOC.0=0
012464  005110                          COM     (R0)            ;LOC. 0= 1
012466  010120                          MOV     R1,(R0)+        ;TRY MOVE MODE 0,2
012470  100402                          BMI     MDM2A           ;BR TO ERROR IF N SET
012472  102401                          BVS     MDM2A           ;BR TO ERROR IF V SET
```

```
4273  012474  001404                    BEQ     MDM2B
4274                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4275                                                ;           CONDITIONAL BRANCH INST. AND   <====
4276                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4277                                                ;           WHICH FOLLOWS W/ 767           <====
4278  012476                    MDM2A:
4279  012476  012742  000304          MOV     #304,-(R2)      ;MOVE TO MAILBOX #  *******  304  *******
4280  012502  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4281  012504  000000                    HALT                    ;CC'S INCORRECT
4282  012506  005300            MDM2B:  DEC     R0
4283  012510  005300                    DEC     R0
4284  012512  001404                    BEQ     MDM2D
4285                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4286                                                ;           CONDITIONAL BRANCH INST. AND   <====
4287                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4288                                                ;           WHICH FOLLOWS W/ 760           <====
4289  012514                    MDM2C:
4290  012514  012742  000305          MOV     #305,-(R2)      ;MOVE TO MAILBOX #  *******  305  *******
4291  012520  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4292  012522  000000                    HALT                    ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
4293  012524  005737  000000    MDM2D:  TST     @#0
4294  012530  001404                    BEQ     TS146
4295                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4296                                                ;           CONDITIONAL BRANCH INST. AND   <====
4297                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4298                                                ;           WHICH FOLLOWS W/ 751           <====
4299  012532  012742  000306          MOV     #306,-(R2)      ;MOVE TO MAILBOX #  *******  306  *******
4300  012536  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4301  012540  000000                    HALT                    ;DESTINATION DATA INCORRECT
4302                                                ;  OR SEQUENCE ERROR
4303
4304            ;*****************************************************************************************
4305            ;
4306            ;       THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
4307            ;INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
4308            ;
4309            ;*****************************************************************************************
4310            ;TEST 146       TEST MOV-BYTE DESTINATION MODE 2
4311            ;*****************************************************************************************
4312  012542  005212            TS146:  INC     (R2)            ;UPDATE TEST NUMBER
4313  012544  022712  000146           CMP     #146,(R2)       ;SEQUENCE ERROR?
4314  012550  001046                    BNE     TS147-10        ;BR TO ERROR HALT ON SEQ ERROR
4315  012552  005000                    CLR     R0              ;R0=0
4316  012554  005010                    CLR     (R0)            ;LOC. 0=0
4317  012556  112720  000125          MOVB    #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
4318  012562  102402                    BVS     MBDM2A          ;BR TO ERROR IF V SET
4319  012564  001401                    BEQ     MBDM2A          ;BR TO ERROR IF Z SET
4320  012566  100004                    BPL     MBDM2B
4321                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4322                                                ;           CONDITIONAL BRANCH INST. AND   <====
4323                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4324                                                ;           WHICH FOLLOWS W/ 770           <====
4325  012570                    MBDM2A:
4326  012570  012742  000307          MOV     #307,-(R2)      ;MOVE TO MAILBOX #  *******  307  *******
4327  012574  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4328  012576  000000                    HALT                    ;CC'S INCORRECT
```

```
 4329  012600  022700  000001          MBDM2B: CMP     #1,R0
 4330  012604  001404                          BEQ     MBDM2C
 4331                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4332                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 4333                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 4334                                                    ;           WHICH FOLLOWS W/ 761           <====
 4335  012606  012742  000310                MOV     #310,-(R2)      ;MOVE TO MAILBOX #  *******  310  *******
 4336  012612  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4337  012614  000000                          HALT                    ;REGISTER NOT INCREMENTED BY ONE
 4338  012616  112720  000252          MBDM2C: MOVB    #252,(R0)+      ;TRY DESTINATION MODE 2 W/ODD BYTE
 4339  012622  102402                          BVS     MBDM2D
 4340  012624  001401                          BEQ     MBDM2D
 4341  012626  100404                          BMI     MBDM2E
 4342                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4343                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 4344                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 4345                                                    ;           WHICH FOLLOWS W/ 750           <====
 4346  012630                          MBDM2D:
 4347  012630  012742  000311                MOV     #311,-(R2)      ;MOVE TO MAILBOX #  *******  311  *******
 4348  012634  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4349  012636  000000                          HALT                    ;CC'S NOT SET CORRECT
 4350  012640  022700  000002          MBDM2E: CMP     #2,R0
 4351  012644  001404                          BEQ     MBDM2F
 4352                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4353                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 4354                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 4355                                                    ;           WHICH FOLLOWS W/ 741           <====
 4356  012646  012742  000312                MOV     #312,-(R2)      ;MOVE TO MAILBOX #  *******  312  *******
 4357  012652  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4358  012654  000000                          HALT                    ;REGISTER NOT INCREMENTED BY ONE
 4359  012656  022737  125125  000000  MBDM2F: CMP     #125125,@#0     ;CHECK DATA
 4360  012664  001404                          BEQ     TS147
 4361                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4362                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 4363                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 4364                                                    ;           WHICH FOLLOWS W/ 731           <====
 4365  012666  012742  000313                MOV     #313,-(R2)      ;MOVE TO MAILBOX #  *******  313  *******
 4366  012672  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4367  012674  000000                          HALT                    ;DESTINATION DATA INCORRECT
 4368                                                                  ;  OR SEQUENCE ERROR
 4369
 4370                          ;*****************************************************************************
 4371                          ;
 4372                          ;       THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
 4373                          ;AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB
 4374                          ;INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
 4375                          ;*****************************************************************************
 4376                          ;TEST 147       TEST MOV(B) DESTINATION MODE 3
 4377                          ;*****************************************************************************
 4378  012676  005212          TS147:  INC     (R2)            ;UPDATE TEST NUMBER
 4379  012700  022712  000147          CMP     #147,(R2)       ;SEQUENCE ERROR?
 4380  012704  001057                  BNE     TS150-10        ;BR TO ERROR HALT ON SEQ ERROR
 4381  012706  012700  000400          MOV     #400,R0         ;R0=400
 4382  012712  005010                  CLR     (R0)            ;LOC. 400 POINTS TO LOC. 0
 4383  012714  005037  000000          CLR     @#0             ;LOC. 0=0
 4384  012720  012730  125252          MOV     #125252,@(R0)+  ;TRY MOV DESTINATION MODE 2
```

```
4385  012724  102402                        BVS     MDM3A           ;BR TO ERROR IF V SET
4386  012726  001401                        BEQ     MDM3A           ;BR TO ERROR IF Z SET
4387  012730  100404                        BMI     MDM3B
4388                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4389                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4390                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4391                                                                  ;           WHICH FOLLOWS W/ 765           <====
4392  012732                          MDM3A:
4393  012732  012742  000314                MOV     #314,-(R2)      ;MOVE TO MAILBOX #  *******  314  *******
4394  012736  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4395  012740  000000                        HALT                    ;CC'S INCORRECT
4396  012742  022700  000402        MDM3B:  CMP     #402,R0         ;CHECK DEST. MODE REGISTER
4397  012746  001404                        BEQ     MDM3C
4398                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4399                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4400                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4401                                                                  ;           WHICH FOLLOWS W/ 756           <====
4402  012750  012742  000315                MOV     #315,-(R2)      ;MOVE TO MAILBOX #  *******  315  *******
4403  012754  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4404  012756  000000                        HALT                    ;REGISTER NOT INCREMENTED BY 2
4405  012760  022737  125252  000000  MDM3C:  CMP     #125252,@#0     ;CHECK DESTINATION DATA
4406  012766  001404                        BEQ     MDM3D
4407                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4408                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4409                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4410                                                                  ;           WHICH FOLLOWS W/ 746           <====
4411  012770  012742  000316                MOV     #316,-(R2)      ;MOVE TO MAILBOX #  *******  316  *******
4412  012774  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4413  012776  000000                        HALT                    ;DESTINATION DATA INCORRECT
4414  013000  112737  000125  000000  MDM3D:  MOVB    #125,@#0        ;TRY MOVB DESTINATION MODE 2 EVEN BYTE
4415  013006  022737  125125  000000        CMP     #125125,@#0     ;CHECK DATA
4416  013014  001404                        BEQ     MDM3E
4417                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4418                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4419                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4420                                                                  ;           WHICH FOLLOWS W/ 733           <====
4421  013016  012742  000317                MOV     #317,-(R2)      ;MOVE TO MAILBOX #  *******  317  *******
4422  013022  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4423  013024  000000                        HALT                    ;DESTINATION DATA INCORRECT
4424  013026  112737  000525  000001  MDM3E:  MOVB    #525,@#1        ;TRY MOVB DESTINATION MODE 2 ODD BYTE
4425  013034  022737  052525  000000        CMP     #52525,@#0      ;CHECK DATA
4426  013042  001404                        BEQ     TS150
4427                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4428                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
4429                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
4430                                                                  ;           WHICH FOLLOWS W/ 720           <====
4431  013044  012742  000320                MOV     #320,-(R2)      ;MOVE TO MAILBOX #  *******  320  *******
4432  013050  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4433  013052  000000                        HALT                    ;
4434
4435                          ;*************************************************************************************
4436                          ;
4437                          ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4438                          ;SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4439                          ;R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4440                          ;CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
```

```
4441                                    ;
4442                                    ;*************************************************************************************
4443                                    ;TEST 150       TEST MOV DESTINATION MODE 4
4444                                    ;*************************************************************************************
4445  013054  005212                TS150:  INC     (R2)            ;UPDATE TEST NUMBER
4446  013056  022712  000150               CMP     #150,(R2)       ;SEQUENCE ERROR?
4447  013062  001026                       BNE     TS151-10        ;BR TO ERROR HALT ON SEQ ERROR
4448  013064  005000                       CLR     R0              ;R0=0
4449  013066  005010                       CLR     (R0)            ;LOC 0=0
4450  013070  012704  000002               MOV     #2,R4           ;R4=2
4451  013074  012744  012345               MOV     #12345,-(R4)    ;TRY MOV DEST. MODE 4
4452  013100  102402                       BVS     MDM4A           ;BR TO ERROR IF V-BIT SET
4453  013102  001401                       BEQ     MDM4A           ;BR TO ERROR IF Z-BIT SET
4454  013104  100004                       BPL     MDM4B
4455                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4456                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4457                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4458                                                                ;           WHICH FOLLOWS W/ 766           <====
4459  013106                        MDM4A:
4460  013106  012742  000321               MOV     #321,-(R2)      ;MOVE TO MAILBOX #  *******  321  *******
4461  013112  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4462  013114  000000                       HALT                    ;CC'S NOT CORRECT
4463  013116  005704                MDM4B:  TST     R4              ;CHECK DECREMENTING OF MODE 4 REG.
4464  013120  001404                       BEQ     MDM4C
4465                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4466                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4467                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4468                                                                ;           WHICH FOLLOWS W/ 760           <====
4469  013122  012742  000322               MOV     #322,-(R2)      ;MOVE TO MAILBOX #  *******  322  *******
4470  013126  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4471  013130  000000                       HALT                    ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
4472  013132  022710  012345        MDM4C:  CMP     #12345,(R0)     ;CHECK DESTINATION DATA
4473  013136  001404                       BEQ     TS151
4474                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4475                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4476                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4477                                                                ;           WHICH FOLLOWS W/ 751           <====
4478  013140  012742  000323               MOV     #323,-(R2)      ;MOVE TO MAILBOX #  *******  323  *******
4479  013144  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4480  013146  000000                       HALT                    ;DESTINATION DATA INCORRECT
4481                                                                ;  OR SEQUENCE ERROR
4482
4483                                    ;*************************************************************************************
4484                                    ;
4485                                    ;       THIS TEST VERIFIES THE MOVB DESTINATION MODE 4 INSTRUCTION
4486                                    ;       ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
4487                                    ;       USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
4488                                    ;       ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH
4489                                    ;       INSTRUCTIONS ARE USED TO VERIFY THE DATA.
4490                                    ;
4491                                    ;*************************************************************************************
4492                                    ;TEST 151       TEST MOVB DESTINATION MODE 4
4493                                    ;*************************************************************************************
4494  013150  005212                TS151:  INC     (R2)            ;UPDATE TEST NUMBER
4495  013152  022712  000151               CMP     #151,(R2)       ;SEQUENCE ERROR?
4496  013156  001046                       BNE     TS152-10        ;BR TO ERROR HALT ON SEQ ERROR
```

```
4497  013160  005004                    CLR     R4              ;R4=0
4498  013162  005014                    CLR     (R4)            ;LOC. 0=0
4499  013164  012700  000002            MOV     #2,R0           ;R0 = 2
4500  013170  112740  125125            MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4-ODD BYTE
4501  013174  020027  000001            CMP     R0,#1           ;CHECK THAT DEST. REG. WAS DECREMENTED
4502  013200  001404                    BEQ     MBDM4A
4503                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4504                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4505                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4506                                                            ;           WHICH FOLLOWS W/ 766           <====
4507  013202  012742  000324            MOV     #324,-(R2)      ;MOVE TO MAILBOX #  *******  324  *******
4508  013206  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4509  013210  000000                    HALT                    ;DESTINATION REG. NOT DECREMENTED BY 1
4510  013212  021427  052400  MBDM4A:   CMP     (R4),#52400     ;CHECK DEST. DATA
4511  013216  001404                    BEQ     MBDM4B
4512                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4513                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4514                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4515                                                            ;           WHICH FOLLOWS W/ 757           <====
4516  013220  012742  000325            MOV     #325,-(R2)      ;MOVE TO MAILBOX #  *******  325  *******
4517  013224  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4518  013226  000000                    HALT                    ;DEST. DATA NOT CORRECT
4519  013230  112740  125125  MBDM4B:   MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4--EVEN BYTE
4520  013234  102402                    BVS     MBDM4C          ;BR. TO ERROR IF V-BIT SET
4521  013236  001401                    BEQ     MBDM4C          ;BR TO ERROR IF Z-BIT SET
4522  013240  100004                    BPL     MBDM4D
4523                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4524                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4525                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4526                                                            ;           WHICH FOLLOWS W/ 746           <====
4527  013242                    MBDM4C:
4528  013242  012742  000326            MOV     #326,-(R2)      ;MOVE TO MAILBOX #  *******  326  *******
4529  013246  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4530  013250  000000                    HALT                    ;COND. CODES INCORRECT
4531  013252  005700  MBDM4D:   TST     R0                      ;CHECK MODE 4 DEST. REGISTER
4532  013254  001404                    BEQ     MBDM4E
4533                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4534                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4535                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4536                                                            ;           WHICH FOLLOWS W/ 740           <====
4537  013256  012742  000327            MOV     #327,-(R2)      ;MOVE TO MAILBOX #  *******  327  *******
4538  013262  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4539  013264  000000                    HALT                    ;DESTINATION REG NOT DECREMENTED BY 1
4540  013266  021427  052525  MBDM4E:   CMP     (R4),#52525     ;CHECK DEST. DAT^A
4541  013272  001404                    BEQ     TS152
4542                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4543                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4544                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4545                                                            ;           WHICH FOLLOWS W/ 731           <====
4546  013274  012742  000330            MOV     #330,-(R2)      ;MOVE TO MAILBOX #  *******  330  *******
4547  013300  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4548  013302  000000                    HALT                    ;DESTINATION DATA INCORRECT
4549                                                            ;  OR SEQUENCE ERROR
4550
4551                              ;*************************************************************************************
4552                              ;
```

```
4553                                    ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOVB
4554                                    ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
4555                                    ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
4556                                    ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
4557                                    ;THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
4558                                    ;PROPER ADDRESSING AND DATA.
4559                                    ;
4560                                    ;*****************************************************************************
4561                                    ;TEST 152       TEST MOV DESTINATION MODE 5
4562                                    ;*****************************************************************************
4563  013304  005212                    TS152:  INC     (R2)            ;UPDATE TEST NUMBER
4564  013306  022712  000152                    CMP     #152,(R2)       ;SEQUENCE ERROR?
4565  013312  001051                            BNE     TS153-10        ;BR TO ERROR HALT ON SEQ ERROR
4566  013314  005004                            CLR     R4              ;R4=0
4567  013316  005014                            CLR     (R4)            ;LOC. 0 = 0
4568  013320  012700  000400                    MOV     #400,R0         ;R0=400
4569  013324  012750  004321                    MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
4570  013330  102402                            BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
4571  013332  001401                            BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
4572  013334  100004                            BPL     MDM5B
4573                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4574                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4575                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4576                                                                    ;           WHICH FOLLOWS W/ 766           <====
4577  013336                            MDM5A:
4578  013336  012742  000331                    MOV     #331,-(R2)      ;MOVE TO MAILBOX #  *******  331  *******
4579  013342  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4580  013344  000000                            HALT                    ;COND. CODES INCORRECT
4581  013346  022700  000376            MDM5B:  CMP     #376,R0         ;CHECK MODE 5 REG. WAS DECREMENTED
4582  013352  001404                            BEQ     MDM5C
4583                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4584                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4585                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4586                                                                    ;           WHICH FOLLOWS W/ 757           <====
4587  013354  012742  000332                    MOV     #332,-(R2)      ;MOVE TO MAILBOX #  *******  332  *******
4588  013360  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4589  013362  000000                            HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
4590  013364  022714  004321            MDM5C:  CMP     #4321,(R4)      ;CHECK DEST. DATA
4591  013370  001404                            BEQ     MDM5D
4592                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4593                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4594                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4595                                                                    ;           WHICH FOLLOWS W/ 750           <====
4596  013372  012742  000333                    MOV     #333,-(R2)      ;MOVE TO MAILBOX #  *******  333  *******
4597  013376  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4598  013400  000000                            HALT                    ;DEST. DATA INCORRECT
4599  013402  012700  000406            MDM5D:  MOV     #406,R0         ;R0=406
4600  013406  112750  000377                    MOVB    #377,@-(R0)     ;TRY MOV DEST. MODE 5 --EVEN BYTE
4601  013412  022700  000404                    CMP     #404,R0         ;CHECK MODE 5 REG.
4602  013416  001404                            BEQ     MDM5E
4603                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4604                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4605                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4606                                                                    ;           WHICH FOLLOWS W/ 735           <====
4607  013420  012742  000334                    MOV     #334,-(R2)      ;MOVE TO MAILBOX #  *******  334  *******
4608  013424  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
4609  013426  000000                        HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
4610  013430  022714  177721    MDM5E:      CMP     #177721,(R4)    ;CHECK DEST. DATA
4611  013434  001404                        BEQ     TS153
4612                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4613                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4614                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4615                                                                ;           WHICH FOLLOWS W/ 726           <====
4616  013436  012742  000335                  MOV     #335,-(R2)      ;MOVE TO MAILBOX #  *******  335  *******
4617  013442  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4618  013444  000000                        HALT                    ;DEST. DATA INCORRECT
4619                                                                ;  OR SEQUENCE ERROR
4620
4621                                    ;*****************************************************************************
4622                                    ;
4623                                    ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE
4624                                    ;DESTINATION MODE 6 INDTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0
4625                                    ;FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0
4626                                    ;BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY
4627                                    ;PROPER ADDRESSING AND DATA.
4628                                    ;
4629                                    ;*****************************************************************************
4630                                    ;TEST 153       TEST MOV DESTINATION MODE 6
4631                                    ;*****************************************************************************
4632  013446  005212            TS153:  INC     (R2)            ;UPDATE TEST NUMBER
4633  013450  022712  000153           CMP     #153,(R2)       ;SEQUENCE ERROR?
4634  013454  001054                   BNE     TS154-10        ;BR TO ERROR HALT ON SEQ ERROR
4635  013456  005000                   CLR     R0              ;R0=0
4636  013460  005010                   CLR     (R0)            ;LOC. 0=0
4637  013462  005200                   INC     R0              ;R0=1
4638  013464  012760  052525  177777     MOV     #052525,-1(R0)  ;TRY MOV DEST. MODE 6
4639  013472  102402                   BVS     MDM6A           ;BR TO ERROR IF V-BIT SET
4640  013474  001401                   BEQ     MDM6A           ;BR TO ERROR IF Z-BIT SET
4641  013476  100004                   BPL     MDM6B
4642                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4643                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4644                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4645                                                            ;           WHICH FOLLOWS W/ 766           <====
4646  013500                    MDM6A:
4647  013500  012742  000336           MOV     #336,-(R2)      ;MOVE TO MAILBOX #  *******  336  *******
4648  013504  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4649  013506  000000                   HALT                    ;COND. CODES INCORRECT
4650  013510  022700  000001    MDM6B:  CMP     #1,R0           ;CHECK DEST. REGISTER UNALTERED
4651  013514  001404                   BEQ     MDM6C
4652                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4653                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4654                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4655                                                            ;           WHICH FOLLOWS W/ 757           <====
4656  013516  012742  000337           MOV     #337,-(R2)      ;MOVE TO MAILBOX #  *******  337  *******
4657  013522  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4658  013524  000000                   HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4659  013526  022737  052525  000000 MDM6C:  CMP     #52525,@#0      ;CHECK DEST. DATA
4660  013534  001404                   BEQ     MDM6D
4661                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4662                                                            ;           CONDITIONAL BRANCH INST. AND   <====
4663                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
4664                                                            ;           WHICH FOLLOWS W/ 747           <====
```

```
4665  013536  012742  000340              MOV     #340,-(R2)      ;MOVE TO MAILBOX #  *******  340  *******
4666  013542  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4667  013544  000000                      HALT                    ;DEST. DATA INCORRECT
4668  013546  012700  000002      MDM6D:  MOV     #2,R0           ;R0=2
4669  013552  112760  000377  177777      MOVB    #377,-1(R0)     ;TRY MOVB DEST. MODE 6
4670  013560  022700  000002              CMP     #2,R0           ;CHECK DEST. REGISTER UNALTERED
4671  013564  001404                      BEQ     MDM6E
4672                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4673                                                              ;           CONDITIONAL BRANCH INST. AND   <====
4674                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
4675                                                              ;           WHICH FOLLOWS W/ 733           <====
4676  013566  012742  000341              MOV     #341,-(R2)      ;MOVE TO MAILBOX #  *******  341  *******
4677  013572  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4678  013574  000000                      HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4679  013576  022737  177525  000000  MDM6E:  CMP     #177525,@#0     ;CHECK DEST. DATA
4680  013604  001404                      BEQ     TS154
4681                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4682                                                              ;           CONDITIONAL BRANCH INST. AND   <====
4683                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
4684                                                              ;           WHICH FOLLOWS W/ 723           <====
4685  013606  012742  000342              MOV     #342,-(R2)      ;MOVE TO MAILBOX #  *******  342  *******
4686  013612  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4687  013614  000000                      HALT                    ;DEST. DATA INCORRECT
4688                                                              ;  OR SEQUENCE ERROR
4689
4690                          ;*************************************************************************************
4691                          ;
4692                          ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
4693                          ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
4694                          ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
4695                          ;USED TO VERIFY PROPER ADDRESSING AND DATA.
4696                          ;
4697                          ;*************************************************************************************
4698                          ;TEST 154       TEST MOV DESTINATION MODE 7
4699                          ;*************************************************************************************
4700  013616  005212          TS154:  INC     (R2)            ;UPDATE TEST NUMBER
4701  013620  022712  000154          CMP     #154,(R2)       ;SEQUENCE ERROR?
4702  013624  001053                  BNE     TS155-10        ;BR TO ERROR HALT ON SEQ ERROR
4703  013626  005004                  CLR     R4              ;R4=0
4704  013630  005014                  CLR     (R4)            ;LOC.0=0
4705  013632  012700  000403          MOV     #403,R0         ;R0=403
4706  013636  012770  070707  177777  MOV     #70707,@-1(R0)  ;TRY MOV W/DEST MODE 7
4707  013644  102402                  BVS     MDM7A           ;BR. TO ERROR IF V-BIT SET
4708  013646  001401                  BEQ     MDM7A           ;BR TO ERROR IF Z-BIT SET
4709  013650  100004                  BPL     MDM7B
4710                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4711                                                          ;           CONDITIONAL BRANCH INST. AND   <====
4712                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
4713                                                          ;           WHICH FOLLOWS W/ 765           <====
4714  013652                  MDM7A:
4715  013652  012742  000343          MOV     #343,-(R2)      ;MOVE TO MAILBOX #  *******  343  *******
4716  013656  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4717  013660  000000                  HALT                    ;COND. CODES INCORRECT
4718  013662  022700  000403  MDM7B:  CMP     #403,R0         ;CHECK DEST. REGISTER
4719  013666  001404                  BEQ     MDM7C
4720                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
4721                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4722                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4723                                                    ;           WHICH FOLLOWS W/ 756           <====
4724  013670  012742  000344          MOV     #344,-(R2)      ;MOVE TO MAILBOX #  *******  344  *******
4725  013674  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4726  013676  000000                  HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4727  013700  022737  070707  000000  MDM7C:  CMP     #70707,@#0      ;CHECK DEST. DATA
4728  013706  001404                  BEQ     MDM7D
4729                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4730                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4731                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4732                                                    ;           WHICH FOLLOWS W/ 746           <====
4733  013710  012742  000345          MOV     #345,-(R2)      ;MOVE TO MAILBOX #  *******  345  *******
4734  013714  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4735  013716  000000                  HALT                    ;DEST. DATA INCORRECT
4736  013720  112770  107070  000001  MDM7D:  MOVB    #107070,@1(R0)  ;TRY MOVB W/DEST MODE 7--ODD BYTE
4737  013726  022700  000403          CMP     #403,R0         ;CHECK MODE 7 DEST. REG.
4738  013732  001404                  BEQ     MDM7E
4739                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4740                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4741                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4742                                                    ;           WHICH FOLLOWS W/ 734           <====
4743  013734  012742  000346          MOV     #346,-(R2)      ;MOVE TO MAILBOX #  *******  346  *******
4744  013740  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4745  013742  000000                  HALT                    ;DEST. DATA INCORRECT
4746  013744  022737  034307  000000  MDM7E:  CMP     #34307,@#0      ;CHECK DEST. DATA
4747  013752  001404                  BEQ     TS155
4748                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4749                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4750                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4751                                                    ;           WHICH FOLLOWS W/ 724           <====
4752  013754  012742  000347          MOV     #347,-(R2)      ;MOVE TO MAILBOX #  *******  347  *******
4753  013760  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4754  013762  000000                  HALT                    ;DESTINATION DATA INCORRECT
4755                                                    ;  OR SEQUENCE ERROR
4756
4757            ;*****************************************************************************************
4758            ;
4759            ;       THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
4760            ;THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
4761            ;TABLE OF OPERANDS.  THE TABLE OF OPERANDS AND THE WORK LOCATION IS
4762            ;STORED FOLLOWING THE TEST CODE.  A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
4763            ;THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
4764            ;VALUE.  THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
4765            ;GO UNDETECTED.  WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
4766            ;ODD ADDRESSES ARE USED IN THE TEST.  THE LISTING SHOWS THE
4767            ;EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
4768            ;
4769            ;*****************************************************************************************
4770            ;TEST 155       TEST MODE 4 W/ DOP INSTS.
4771            ;*****************************************************************************************
4772  013764  005212          TS155:  INC     (R2)            ;UPDATE TEST NUMBER
4773  013766  022712  000155          CMP     #155,(R2)       ;SEQUENCE ERROR?
4774  013772  001015                  BNE     DOP4            ;BR TO ERROR HALT ON SEQ ERROR
4775  013774  012700  014046          MOV     #TBL1,R0        ;INITIALIZE R0
4776  014000  014037  014046          MOV     -(R0),@#TBL1    ;TBL1=125252
```

```
                        ADD     -(R0),@#TBL1    ;TBL1=000377
                        BICB    -(R0),@#TBL1    ;TBL1=000252
                        BISB    -(R0),@#TBL1+1  ;TBL1=125252
                        CMP     -(R0),@#TBL1    ;CHECK RESULT
                        BEQ     TS156
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 762           <====
                DOP4:
                        MOV     #350,-(R2)      ;MOVE TO MAILBOX #  *******  350  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        HALT                    ;RESULT OF MODE 4 INSTS. INCORRECT
                                                ;  OR SEQUENCE ERROR

                        125252
                        52652
                        53125
                        125252
                TBL1:   0

                ;*****************************************************************************************
                ;
                ;       THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
                ;THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
                ;THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
                ;THE DATA TABLE USED IN THE PREVIOUS TEST.  THE TEST IS IDENTICAL TO
                ;THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
                ;TABLE AND MODE 5 ADDRESSING.  (SEE PREVIOUS TEST).
                ;
                ;*****************************************************************************************
                ;TEST 156       TEST MODE 5 W/ DOP INSTS.
                ;*****************************************************************************************
                TS156:  INC     (R2)            ;UPDATE TEST NUMBER
                        CMP     #156,(R2)       ;SEQUENCE ERROR?
                        BNE     DOP5            ;BR TO ERROR HALT ON SEQ ERROR
                        MOV     #TBL2+2,R0      ;INITIALIZE R0
                        MOV     @-(R0),@#TBL1   ;TBL1=125252
                        ADD     @-(R0),@#TBL1   ;TBL1=000377
                        BICB    @-(R0),@#TBL1   ;TBL1=000252
                        BISB    @-(R0),@#TBL1+1 ;TBL1=125252
                        CMP     @-(R0),@#TBL1   ;CHECK RESULT
                        BEQ     TS157
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 762           <====
                DOP5:
                        MOV     #351,-(R2)      ;MOVE TO MAILBOX #  *******  351  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        HALT                    ;RESULT OF MODE 5 INSTS. INCORRECT
                                                ;  OR SEQUENCE ERROR

                        TBL1-10
                        TBL1-6
                        TBL1-5
                        TBL1-4
```

```
                                                    D 8
  4833  014132  014044                TBL2:   TBL1-2
```

```
4834                                ;*****************************************************************************
4835                                ;
4836                                ;       THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
4837                                ;IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
4838                                ;THIS TIME THE DATA IS ACCESSED USING MODE 6.  R0 IS SET
4839                                ;TO POINT TO THE MIDDLE OF THE TABLE.  THE TABLE IS ACCESSED FROM
4840                                ;BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
4841                                ;THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
4842                                ;TESTS.
4843                                ;
4844                                ;*****************************************************************************
4845                                ;TEST 157       TEST MODE 6 W/ DOP INSTS.
4846                                ;*****************************************************************************
4847  014134  005212                TS157:  INC     (R2)            ;UPDATE TEST NUMBER
4848  014136  022712  000157                CMP     #157,(R2)       ;SEQUENCE ERROR?
4849  014142  001022                        BNE     TS160-10        ;BR TO ERROR HALT ON SEQ ERROR
4850  014144  012700  014042                MOV     #TBL1-4,R0      ;INITIALIZE R0
4851  014150  016037  000002  014046        MOV     2(R0),@#TBL1    ;TBL1=125252
4852  014156  066037  000000  014046        ADD     0(R0),@#TBL1    ;TBL1=000377
4853  014164  146037  177777  014046        BICB    -1(R0),@#TBL1   ;TBL1=000252
4854  014172  156037  177776  014047        BISB    -2(R0),@#TBL1+1 ;TBL1=125252
4855  014200  026037  177774  014046        CMP     -4(R0),@#TBL1   ;CHECK RESULT
4856  014206  001404                        BEQ     TS160
4857                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4858                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4859                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4860                                                                ;           WHICH FOLLOWS W/ 755           <====
4861  014210  012742  000352                MOV     #352,-(R2)      ;MOVE TO MAILBOX #  *******  352  *******
4862  014214  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4863  014216  000000                        HALT                    ;RESULT OF MODE 6 INSTS. INCORRECT
4864                                                                ;  OR SEQUENCE ERROR
4865                                ;*****************************************************************************
4866                                ;
4867                                ;       THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
4868                                ;THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
4869                                ;THE MODE 5 TESTS.  THIS TIME THE DATA IS ACCESSED USING MODE 7.
4870                                ;R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
4871                                ;TEST.  THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
4872                                ;IN THE MODE 7 INSTRUCTIONS.  THE DATA RESULTS ARE IDENTICAL TO
4873                                ;THOSE EXPECTED IN THE MODE 5 TESTS.
4874                                ;
4875                                ;*****************************************************************************
4876                                ;TEST 160       TEST MODE 7 W/ DOP INSTS.
4877                                ;*****************************************************************************
4878  014220  005212                TS160:  INC     (R2)            ;UPDATE TEST NUMBER
4879  014222  022712  000160                CMP     #160,(R2)       ;SEQUENCE ERROR?
4880  014226  001022                        BNE     TS161-10        ;BR TO ERROR HALT ON SEQ ERROR
4881  014230  012700  014126                MOV     #TBL2-4,R0      ;INITIALIZE R0
4882  014234  017037  000004  014046        MOV     @4(R0),@#TBL1   ;TBL1=125252
4883  014242  067037  000002  014046        ADD     @2(R0),@#TBL1   ;TBL1=000377
4884  014250  147037  000000  014046        BICB    @0(R0),@#TBL1   ;TBL1=000252
4885  014256  157037  177776  014047        BISB    @-2(R0),@#TBL1+1                ;TBL1=125252
4886  014264  027037  177774  014046        CMP     @-4(R0),@#TBL1          ;CHECK RESULT
4887  014272  001404                        BEQ     TS161
4888                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4889                                                                ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 755           <====
                        MOV     #353,-(R2)      ;MOVE TO MAILBOX #  *******  353  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        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
                ;*****************************************************************************
014304  005212  TS161:  INC     (R2)            ;UPDATE TEST NUMBER
014306  022712  000161          CMP     #161,(R2)       ;SEQUENCE ERROR?
014312  001026                  BNE     TS162-10        ;BR TO ERROR HALT ON SEQ ERROR
014314  012700  125252          MOV     #125252,R0      ;INITIALIZE DATA
014320  000261                  SEC                     ;SET C-BIT
014322  006100                  ROL     R0              ;TRY ROL W/ MODE 0
014324  102004                  BVC     ROT0A           ;CC=0011
014326  103003                  BCC     ROT0A
014330  022700  052525          CMP     #052525,R0      ;CHECK DATA
014334  001404                  BEQ     ROT0B
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 766           <====
014336  ROT0A:
014336  012742  000354                  MOV     #354,-(R2)      ;MOVE TO MAILBOX #  *******  354  *******
014342  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014344  000000                  HALT                    ;ROL MODE 0 FAILED
014346  012700  125252  ROT0B:  MOV     #125252,R0      ;INITIALIZE DATA
014352  000261                  SEC                     ;SET C-BIT
014354  106100                  ROLB    R0              ;TRY ROL W/ MODE 0 EVEN BYTE
014356  102004                  BVC     ROT0C           ;CC=0011
014360  103003                  BCC     ROT0C
014362  022700  125125          CMP     #125125,R0      ;CHECK DATA
014366  001404                  BEQ     TS162
                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 751           <====
014370  ROT0C:
014370  012742  000355                  MOV     #355,-(R2)      ;MOVE TO MAILBOX #  *******  355  *******
014374  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014376  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
                                ;*****************************************************************************
014400  005212                  TS162:  INC     (R2)            ;UPDATE TEST NUMBER
014402  022712  000162                  CMP     #162,(R2)       ;SEQUENCE ERROR?
014406  001051                          BNE     TS163-10        ;BR TO ERROR HALT ON SEQ ERROR
014410  005000                          CLR     R0              ;POINT TO LOC. 0
014412  012710  052525                  MOV     #52525,(R0)     ;INITIALIZE DATA
014416  000241                          CLC                     ;CLEAR C-BIT
014420  006110                          ROL     (R0)            ;TRY ROL W/ MODE 1
014422  102005                          BVC     ROT1A           ;CC=1010
014424  103404                          BCS     ROT1A
014426  023727  000000  125252          CMP     @#0,#125252     ;CHECK RESULT
014434  001404                          BEQ     ROT1B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
014436                          ROT1A:
014436  012742  000356                  MOV     #356,-(R2)      ;MOVE TO MAILBOX #  *******  356  *******
014442  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014444  000000                          HALT                    ;ROL MODE 1 FAILED
014446  000261                  ROT1B:  SEC
014450  012710  125252                  MOV     #125252,(R0)    ;INITIALIZE DATA
014454  106110                          ROLB    (R0)            ;TRY ROLB W/ MODE 1 EVEN BYTE
014456  102005                          BVC     ROT1C           ;CC=1011
014460  103004                          BCC     ROT1C
014462  022737  125125  000000          CMP     #125125,@#0     ;TEST RESULT
014470  001404                          BEQ     ROT1D
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 746           <====
014472                          ROT1C:
014472  012742  000357                  MOV     #357,-(R2)      ;MOVE TO MAILBOX #  *******  357  *******
014476  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014500  000000                          HALT                    ;ROLB W/ MODE 1 EVEN BYTE FAILED
014502  012710  125252          ROT1D:  MOV     #125252,(R0)
014506  005000                          CLR     R0              ;POINT TO ODD BYTE
014510  005200                          INC     R0
014512  000261                          SEC                     ;SET C-BIT
014514  106110                          ROLB    (R0)            ;TRY ROLB W/ MODE 1 ODD BYTE
014516  102005                          BVC     ROT1E           ;CC=0011
014520  103004                          BCC     ROT1E
014522  022737  052652  000000          CMP     #052652,@#0     ;CHECK DATA
014530  001404                          BEQ     TS163
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 726           <====
014532                          ROT1E:
014532  012742  000360                MOV     #360,-(R2)      ;MOVE TO MAILBOX #  *******  360  *******
014536  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014540  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
                                ;*************************************************************************
014542  005212                  TS163:  INC     (R2)            ;UPDATE TEST NUMBER
014544  022712  000163                  CMP     #163,(R2)       ;SEQUENCE ERROR?
014550  001057                        BNE     TS164-10        ;BR TO ERROR HALT ON SEQ ERROR
014552  005000                        CLR     R0              ;POINT TO LOC 0
014554  012710  173737                MOV     #173737,(R0)    ;INITIALIZE DATA
014560  000241                        CLC                     ;CLEAR C-BIT
014562  006120                        ROL     (R0)+           ;TRY ROL W/ MODE 2
014564  103007                        BCC     ROT2A           ;CHECK C-BIT
014566  022737  167676  000000        CMP     #167676,@#0     ;CHECK DATA
014574  001003                        BNE     ROT2A           ;BRANCH IF RESULT INCORRECT
014576  005300                        DEC     R0              ;TEST R0
014600  005300                        DEC     R0
014602  001404                        BEQ     ROT2B
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 762           <====
014604                          ROT2A:
014604  012742  000361                MOV     #361,-(R2)      ;MOVE TO MAILBOX #  *******  361  *******
014610  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014612  000000                      HALT                    ;ROL W/ MODE 2 FAILED
014614  005000                  ROT2B:  CLR     R0              ;POINT TO LOC 0
014616  012710  004040                MOV     #4040,(R0)      ;INITIALIZE DATA
014622  000241                        CLC                     ;CLEAR C-BIT
014624  106120                        ROLB    (R0)+           ;TRY ROLB W/ MODE 2 EVEN BYTE
014626  103406                        BCS     ROT2C           ;CHECK C-BIT
014630  022737  004100  000000        CMP     #4100,@#0       ;CHECK DATA
014636  001002                        BNE     ROT2C           ;BRANCH IF DATA INCORRECT
014640  005300                        DEC     R0              ;CHECK R0
014642  001404                        BEQ     ROT2D
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 742           <====
014644                          ROT2C:
014644  012742  000362                MOV     #362,-(R2)      ;MOVE TO MAILBOX #  *******  362  *******
```

```
5053  014650  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5054  014652  000000                        HALT                    ;ROLB W/ MODE 2 EVEN BYTE FAILED
5055  014654  005000                ROT2D:  CLR     R0              ;POINT TO LOC 0
5056  014656  012710  004040              MOV     #4040,(R0)      ;INITIALIZE DATA
5057  014662  005200                        INC     R0              ;POINT TO ODD BYTE OF DATA
5058  014664  000261                        SEC                     ;SET C-BIT
5059  014666  106120                        ROLB    (R0)+           ;TRY ROL W/ MODE 2 ODD BYTE
5060  014670  103407                        BCS     ROT2E           ;CHECK C-BIT
5061  014672  022737  010440  000000        CMP     #10440,@#0      ;CHECK DATA
5062  014700  001003                        BNE     ROT2E           ;BRANCH IF DATA INCORRECT
5063  014702  005300                        DEC     R0              ;CHECK R0
5064  014704  005300                        DEC     R0
5065  014706  001404                        BEQ     TS164
5066                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5067                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5068                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5069                                                                ;           WHICH FOLLOWS W/ 720           <====
5070  014710                        ROT2E:
5071  014710  012742  000363              MOV     #363,-(R2)      ;MOVE TO MAILBOX #  *******  363  *******
5072  014714  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5073  014716  000000                        HALT                    ;ROLB W/ MODE 2 ODD BYTE FAILED
5074                                                                ;  OR SEQUENCE ERROR
```

```
5075
5076                            ;*****************************************************************************************
5077                            ;
5078                            ;       THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
5079                            ;THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
5080                            ;TESTS.  THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
5081                            ;MODE 37.  BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
5082                            ;
5083                            ;*****************************************************************************************
5084                            ;TEST 164       TEST ROTATE INSTRUCTIONS /W MODE 3
5085                            ;*****************************************************************************************
5086  014720  005212            TS164:  INC     (R2)            ;UPDATE TEST NUMBER
5087  014722  022712  000164            CMP     #164,(R2)       ;SEQUENCE ERROR?
5088  014726  001051                    BNE     TS165-10        ;BR TO ERROR HALT ON SEQ ERROR
5089  014730  012737  052525  000000    MOV     #52525,@#0      ;INITIALIZE DATA IN LOC 0
5090  014736  000261                    SEC                     ;SET C-BIT
5091  014740  006137  000000            ROL     @#0             ;TRO ROL W/ MODE 3
5092  014744  103404                    BCS     ROT3A           ;CHECK C-BIT
5093  014746  022737  125253  000000    CMP     #125253,@#0     ;CHECK DATA
5094  014754  001404                    BEQ     ROT3B
5095                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5096                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5097                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5098                                                            ;           WHICH FOLLOWS W/ 764           <====
5099  014756                    ROT3A:
5100  014756  012742  000364            MOV     #364,-(R2)      ;MOVE TO MAILBOX #  *******  364  *******
5101  014762  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5102  014764  000000                    HALT                    ;ROL W/ MODE 3 FAILED
5103  014766  012737  125252  000000    ROT3B:  MOV     #125252,@#0     ;INITIALIZE DATA
5104  014774  000241                    CLC                     ;CLEAR C-BIT
5105  014776  106137  000000            ROLB    @#0             ;TRY ROL W/ MODE 3 EVEN BYTE
5106  015002  103004                    BCC     ROT3C           ;CHECK C-BIT
5107  015004  023727  000000  125124    4$:     CMP     @#0,#125124     ;CHECK DATA
5108  015012  001404                    BEQ     ROT3D
5109                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5110                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5111                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5112                                                            ;           WHICH FOLLOWS W/ 745           <====
5113  015014                    ROT3C:
5114  015014  012742  000365            MOV     #365,-(R2)      ;MOVE TO MAILBOX #  *******  365  *******
5115  015020  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5116  015022  000000                    HALT                    ;ROL W/ MODE 3 EVEN BYTE FAILED
5117  015024  012737  125252  000000    ROT3D:  MOV     #125252,@#0     ;INITIALIZE DATA IN LOC. 0
5118  015032  000261                    SEC                     ;SET C-BIT
5119  015034  106137  000001            ROLB    @#1             ;TRY ROL W/ MODE 3 ODD BYTE
5120  015040  103004                    BCC     ROT3E           ;CHECK C-BIT
5121  015042  022737  052652  000000    CMP     #052652,@#0     ;CHECK DATA
5122  015050  001404                    BEQ     TS165
5123                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5124                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5125                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5126                                                            ;           WHICH FOLLOWS W/ 726           <====
5127  015052                    ROT3E:
5128  015052  012742  000366            MOV     #366,-(R2)      ;MOVE TO MAILBOX #  *******  366  *******
5129  015056  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5130  015060  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
                                ;*******************************************************************************
015062  005212                  TS165:  INC     (R2)            ;UPDATE TEST NUMBER
015064  022712  000165                  CMP     #165,(R2)       ;SEQUENCE ERROR?
015070  001016                          BNE     TS166-10        ;BR TO ERROR HALT ON SEQ ERROR
015072  012737  070707  000000          MOV     #070707,@#0     ;INITIALIZE DATA IN LOC. 0
015100  012700  000002                  MOV     #2,R0           ;INITIALIZE R0 AS POINTER
015104  000261                          SEC                     ;SET C-BIT
015106  006140                          ROL     -(R0)           ;TRY ROL W/ MODE 4
015110  103406                          BCS     ROT4            ;CHECK C-BIT
015112  022737  161617  000000          CMP     #161617,@#0     ;CHECK DATA
015120  001002                          BNE     ROT4            ;BRANCH IF DATA INCORRECT
015122  005700                          TST     R0              ;CHECK MODE 4 REGISTER
015124  001404                          BEQ     TS166
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 761           <====
015126                          ROT4:
015126  012742  000367                  MOV     #367,-(R2)      ;MOVE TO MAILBOX #  *******  367  *******
015132  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015134  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
                                ;*******************************************************************************
015136  005212                  TS166:  INC     (R2)            ;UPDATE TEST NUMBER
015140  022712  000166                  CMP     #166,(R2)       ;SEQUENCE ERROR?
015144  001021                          BNE     ROT5            ;BR TO ERROR HALT ON SEQ ERROR
015146  012737  015220  000000          MOV     #ROTX,@#0       ;MOVE POINTER TO LOC. 0
015154  012700  000002                  MOV     #2,R0           ;SET MODE 5 REG. TO LOC. 0
015160  012767  107070  000032          MOV     #107070,ROTX    ;INITIALIZE DATA
015166  000241                          CLC                     ;CLEAR C-BIT
015170  006150                          ROL     @-(R0)          ;TRY ROL W/ MODE 5
```

```
5187  015172  103006                          BCC     ROT5                    ;CHECK C-BIT
5188  015174  022737  016160  015220          CMP     #016160,@#ROTX          ;CHECK DATA
5189  015202  001002                          BNE     ROT5                    ;BRANCH IF DATA INCORRECT
5190  015204  005700                          TST     R0                      ;CHECK MODE 5 REGISTER
5191  015206  001405                          BEQ     TS167
5192                                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5193                                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5194                                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5195                                                                          ;           WHICH FOLLOWS W/ 756           <====
5196  015210                          ROT5:
5197  015210  012742  000370                          MOV     #370,-(R2)      ;MOVE TO MAILBOX #  *******  370  *******
5198  015214  005242                          INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
5199  015216  000000                          HALT                            ;ROL MODE 5 FAILED
5200                                                                          ;  OR SEQUENCE ERROR
5201  015220  000000                  ROTX:   0
5202
5203                                  ;*****************************************************************************
5204                                  ;
5205                                  ;       THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.
5206                                  ;IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE
5207                                  ;ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.
5208                                  ;THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).
5209                                  ;
5210                                  ;*****************************************************************************
5211                                  ;TEST 167       TEST MODE 6 W/ ROTATE INSTRUCTIONS
5212                                  ;*****************************************************************************
5213  015222  005212                  TS167:  INC     (R2)                    ;UPDATE TEST NUMBER
5214  015224  022712  000167                  CMP     #167,(R2)               ;SEQUENCE ERROR?
5215  015230  001013                          BNE     TS170-10                ;BR TO ERROR HALT ON SEQ ERROR
5216  015232  012737  125252  015220          MOV     #125252,@#ROTX          ;INITIALIZE DATA
5217  015240  000261                          SEC                             ;SET C-BIT
5218  015242  006167  177752                  ROL     ROTX                    ;TRY ROL W/ MODE 6
5219  015246  103004                          BCC     ROT6                    ;CHECK C-BIT
5220  015250  022737  052525  015220          CMP     #52525,@#ROTX           ;CHECK DATA
5221  015256  001404                          BEQ     TS170
5222                                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5223                                                                          ;           CONDITIONAL BRANCH INST. AND   <====
5224                                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
5225                                                                          ;           WHICH FOLLOWS W/ 764           <====
5226  015260                          ROT6:
5227  015260  012742  000371                          MOV     #371,-(R2)      ;MOVE TO MAILBOX #  *******  371  *******
5228  015264  005242                          INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
5229  015266  000000                          HALT                            ;ROL W/ MODE 6 FAILED
5230                                                                          ;  OR SEQUENCE ERROR
```

```
5231
5232                                    ;*****************************************************************************
5233                                    ;
5234                                    ;THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
5235                                    ;THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST).  THE ROL INSTRUCTION
5236                                    ;ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
5237                                    ;(ROTXAD) FOLLOWING THE TEST CODE.
5238                                    ;
5239                                    ;*****************************************************************************
5240                                    ;TEST 170       TEST MODE 7 W/ ROTATE INSTRUCTIONS
5241                                    ;*****************************************************************************
5242  015270  005212                    TS170:  INC     (R2)            ;UPDATE TEST NUMBER
5243  015272  022712  000170                    CMP     #170,(R2)       ;SEQUENCE ERROR?
5244  015276  001016                            BNE     ROT7            ;BR TO ERROR HALT ON SEQ ERROR
5245  015300  012737  052525  015220            MOV     #52525,@#ROTX   ;INITIALIZE DATA
5246  015306  012737  015220  015344            MOV     #ROTX,@#ROTXAD  ;INITIALIZE ADDRESS POINTER
5247  015314  000241                            CLC                     ;CLEAR C-BIT
5248  015316  006177  000022                    ROL     @ROTXAD         ;TRY ROL W/ MODE 7
5249  015322  103404                            BCS     ROT7            ;CHECK C-BIT
5250  015324  023727  015220  125252            CMP     @#ROTX,#125252  ;CHECK DATA
5251  015332  001405                            BEQ     TS171
5252                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5253                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5254                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5255                                                                    ;           WHICH FOLLOWS W/ 761           <====
5256  015334                            ROT7:
5257  015334  012742  000372                    MOV     #372,-(R2)      ;MOVE TO MAILBOX #  *******  372  *******
5258  015340  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5259  015342  000000                            HALT                    ;ROL W/ MODE 7 FAILED
5260                                                                    ;  OR SEQUENCE ERROR
5261  015344  000000                    ROTXAD: 0
5262
5263
5264                                    ;*****************************************************************************
5265                                    ;
5266                                    ;       THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION.  R0 IS SET TO
5267                                    ;177400.  A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH
5268                                    ;IS USED TO CHECK THE SIGN OF THE RESULT.  ALSO, A COMPARISON
5269                                    ;IS MADE TO CHECK THE DATA RESULTS.
5270                                    ;
5271                                    ;*****************************************************************************
5272                                    ;TEST 171       TEST MODE 0 W/ SWAB INST.
5273                                    ;*****************************************************************************
5274  015346  005212                    TS171:  INC     (R2)            ;UPDATE TEST NUMBER
5275  015350  022712  000171                    CMP     #171,(R2)       ;SEQUENCE ERROR?
5276  015354  001013                            BNE     TS172-10        ;BR TO ERROR HALT ON SEQ ERROR
5277  015356  012700  177400                    MOV     #177400,R0      ;MOVE TEST PATTERN TO R0
5278  015362  000300                            SWAB    R0              ;TRY SWAB MODE 0
5279  015364  100404                            BMI     SB0
5280                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5281                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5282                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5283                                                                    ;           WHICH FOLLOWS W/ 773           <====
5284  015366  012742  000373                    MOV     #373,-(R2)      ;MOVE TO MAILBOX #  *******  373  *******
5285  015372  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5286  015374  000000                            HALT                    ;SWAB DID NOT SET CC'S CORRECT
```

```
5287  015376  022700  000377          SB0:    CMP     #377,R0         ;CHECK RESULT
5288  015402  001404                          BEQ     TS172
5289                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5290                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5291                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5292                                                                  ;           WHICH FOLLOWS W/ 764           <====
5293  015404  012742  000374                      MOV     #374,-(R2)      ;MOVE TO MAILBOX #  *******  374  *******
5294  015410  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5295  015412  000000                            HALT                    ;RESULT OF SWAB MODE 0 FAILED
5296                                                                    ;  OR SEQUENCE ERROR
5297
5298                                          ;*****************************************************************************
5299                                          ;
5300                                          ;       THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION.  THE TEST
5301                                          ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE ADDRESSING
5302                                          ;REGISTER IN THE MODE 1 SWAB.  THE DATA RESULTS ARE CHECKED WITH
5303                                          ;A COMPARE.
5304                                          ;
5305                                          ;*****************************************************************************
5306                                          ;TEST 172       TEST MODE 1 W/ SWAB INST
5307                                          ;*****************************************************************************
5308  015414  005212                          TS172:  INC     (R2)            ;UPDATE TEST NUMBER
5309  015416  022712  000172                          CMP     #172,(R2)       ;SEQUENCE ERROR?
5310  015422  001011                          BNE     TS173-10        ;BR TO ERROR HALT ON SEQ ERROR
5311  015424  012737  125652  000000                MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
5312  015432  005000                          CLR     R0              ;R0=0
5313  015434  000310                          SWAB    (R0)            ;TRY SWAB MODE 1
5314  015436  022737  125253  000000                CMP     #125253,@#0     ;CHECK RESULT
5315  015444  001404                          BEQ     TS173
5316                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5317                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5318                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5319                                                                  ;           WHICH FOLLOWS W/ 766           <====
5320  015446  012742  000375                      MOV     #375,-(R2)      ;MOVE TO MAILBOX #  *******  375  *******
5321  015452  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5322  015454  000000                            HALT                    ;RESULT OF SWAB MODE 1 FAILED
5323                                                                    ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION.  THE TEST
                                ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE MODE
                                ;2 ADDRESSING REGISTER.  THE RESULTS ARE CHECKED WITH A COMPARE.
                                ;R0 IS CHECKED FOR PROPER DECREMENTING.
                                ;
                                ;*****************************************************************************
                                ;TEST 173        TEST MODE 2 W/ SWAB INST
                                ;*****************************************************************************
015456  005212                  TS173:  INC     (R2)            ;UPDATE TEST NUMBER
015460  022712  000173                  CMP     #173,(R2)       ;SEQUENCE ERROR?
015464  001020                          BNE     TS174-10        ;BR TO ERROR HALT ON SEQ ERROR
015466  012737  125152  000000          MOV     #125152,@#0     ;MOVE TEST PATTERN TO LOC. 0
015474  005000                          CLR     R0              ;R0=0
015476  000320                          SWAB    (R0)+           ;TRY SWAB MODE 2
015500  022737  065252  000000          CMP     #65252,@#0      ;CHECK RESULT
015506  001404                          BEQ     SB2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
015510  012742  000376                  MOV     #376,-(R2)      ;MOVE TO MAILBOX #  *******  376  *******
015514  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015516  000000                          HALT                    ;RESULT OF SWAB MODE 0 FAILED
015520  162700  000002          SB2:    SUB     #2,R0           ;CHECK EFFECT OF REG.
015524  001404                          BEQ     TS174
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
015526  012742  000377                  MOV     #377,-(R2)      ;MOVE TO MAILBOX #  *******  377  *******
015532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015534  000000                          HALT                    ;REGISTER VALUE INCORRECT
                                                                ;  OR SEQUENCE ERROR


                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION.  THE TEST
                                ;PATTERN IS MOVED TO LOC 0.  A MODE 3 SWAB INSTRUCTION IS EXECUTED
                                ;USING R7 AS THE ADDRESSING REGISTER.  A COMPARE VERIFIES THE
                                ;DATA RESULTS.
                                ;
                                ;*****************************************************************************
                                ;TEST 174        TEST MODE 3 W/SWAB INST.
                                ;*****************************************************************************
015536  005212                  TS174:  INC     (R2)            ;UPDATE TEST NUMBER
015540  022712  000174                  CMP     #174,(R2)       ;SEQUENCE ERROR?
015544  001011                          BNE     TS175-10        ;BR TO ERROR HALT ON SEQ ERROR
015546  012737  000377  000000          MOV     #377,@#0        ;MOVE TEST PATTERN TO LOC. 0
015554  000337  000000                  SWAB    @#0             ;TRY SWAB W/ MODE 3
015560  022737  177400  000000          CMP     #177400,@#0     ;CHECK RESULT
015566  001404                          BEQ     TS175
```

```
5380                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5381                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5382                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5383                                                    ;           WHICH FOLLOWS W/ 766           <====
5384  015570  012742  000400          MOV     #400,-(R2)  ;MOVE TO MAILBOX #  *******  400  *******
5385  015574  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
5386  015576  000000                  HALT                ;RESULT OF SWAB INCORRECT
5387                                                    ;  OR SEQUENCE ERROR
```

```
5388
5389
5390                                    ;*****************************************************************************
5391                                    ;
5392                                    ;       THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
5393                                    ;IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
5394                                    ;REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
5395                                    ;FOR PROPER DECREMENTING.
5396                                    ;
5397                                    ;*****************************************************************************
5398                                    ;TEST 175       TEST MODE 4 W/ SWAB INST
5399                                    ;*****************************************************************************
5400  015600  005212                    TS175:  INC     (R2)            ;UPDATE TEST NUMBER
5401  015602  022712  000175                    CMP     #175,(R2)       ;SEQUENCE ERROR?
5402  015606  001020                            BNE     TS176-10        ;BR TO ERROR HALT ON SEQ ERROR
5403  015610  012737  125652  000000            MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
5404  015616  012700  000002                    MOV     #2,R0           ;SET UP REGISTER POINTER
5405  015622  000340                            SWAB    -(R0)           ;TRY SWAB MODE 4
5406  015624  022737  125253  000000            CMP     #125253,@#0     ;CHECK RESULT
5407  015632  001404                            BEQ     SB4
5408                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5409                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5410                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5411                                                                    ;           WHICH FOLLOWS W/ 765           <====
5412  015634  012742  000401                    MOV     #401,-(R2)      ;MOVE TO MAILBOX #  *******  401  *******
5413  015640  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5414  015642  000000                            HALT                    ;RESULT OF SWAB INCORRECT
5415  015644  005700                    SB4:    TST     R0              ;CHECK EFFECT ON REG.
5416  015646  001404                            BEQ     TS176
5417
5418                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5419                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5420                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5421  015650  012742  000402                            MOV     #402,-(R2)      ;MOVE TO MAILBOX #  *******  402  *******
5422  015654  005242                                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5423  015656  000000                                    HALT                    ;REGISTER VALUE INCORRECT
5424                                                                            ;  OR SEQUENCE ERROR
5425
```

```
5426
5427                                    ;*****************************************************************************************
5428                                    ;
5429                                    ;       THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION.  THE TEST USES
5430                                    ;TWO LOCATIONS FOLLOWING THE TEST CODE.  SB5X HOLDS THE DATA;
5431                                    ;SB5XAD IS A POINTER TO THE DATA LOCATION.  THE DATA IS MOVED TO
5432                                    ;SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD.  FOLLOWING
5433                                    ;THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA.  R0 IS
5434                                    ;CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
5435                                    ;
5436                                    ;*****************************************************************************************
5437                                    ;TEST 176       TEST MODE 5 W/ SWAB INST.
5438                                    ;*****************************************************************************************
5439  015660  005212                    TS176:  INC     (R2)            ;UPDATE TEST NUMBER
5440  015662  022712  000176                    CMP     #176,(R2)       ;SEQUENCE ERROR?
5441  015666  001021                            BNE     SB5             ;BR TO ERROR HALT ON SEQ ERROR
5442  015670  012700  015746                    MOV     #SB5XAD+2,R0    ;SET UP POINTER TO WORK LOCATION
5443  015674  012767  125125  000040            MOV     #125125,SB5X    ;MOVE PATTERN TO WORK LOCATION
5444  015702  000350                            SWAB    @-(R0)          ;TRY SWAB MODE 5
5445  015704  022767  052652  000030            CMP     #52652,SB5X     ;CHECK RESULT
5446  015712  001404                            BEQ     SB5A
5447                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5448                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5449                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5450                                                                    ;           WHICH FOLLOWS W/ 765           <====
5451  015714  012742  000403                    MOV     #403,-(R2)      ;MOVE TO MAILBOX #  *******  403  *******
5452  015720  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5453  015722  000000                            HALT                    ;RESULT OF SWAB INCORRECT
5454  015724  020027  015744            SB5A:   CMP     R0,#SB5XAD      ;CHECK RESULT OF REG.
5455  015730  001406                            BEQ     TS177
5456                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5457                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5458                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5459                                                                    ;           WHICH FOLLOWS W/ 756           <====
5460  015732                            SB5:
5461  015732  012742  000404                    MOV     #404,-(R2)      ;MOVE TO MAILBOX #  *******  404  *******
5462  015736  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5463  015740  000000                            HALT                    ;REGISTER VALUE INCORRECT
5464                                                                    ;  OR SEQUENCE ERROR
5465  015742  000000                    SB5X:   0                       ;WORK LOCATION
5466  015744  015742                    SB5XAD: SB5X
5467
```

```
5468
5469                                    ;*****************************************************************************
5470                                    ;
5471                                    ;       THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION.  THIS TEST
5472                                    ;USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE.  TEST DATA
5473                                    ;IS LOADED INTO THE WORK LOCATION.  R0, THE ADDRESSING REGISTER
5474                                    ;IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
5475                                    ;THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET.  THE DATA IS
5476                                    ;VERIFIED WITH A COMPARE.
5477                                    ;
5478                                    ;*****************************************************************************
5479                                    ;TEST 177       TEST MODE 6 W/ SWAB INST.
5480                                    ;*****************************************************************************
5481  015746  005212                    TS177:  INC     (R2)            ;UPDATE TEST NUMBER
5482  015750  022712  000177                   CMP     #177,(R2)       ;SEQUENCE ERROR?
5483  015754  001013                           BNE     SB6             ;BR TO ERROR HALT ON SEQ ERROR
5484  015756  012767  125125  000030          MOV     #125125,SB6X    ;MOVE PATTERN TO WORK LOCATION
5485  015764  012700  016006                   MOV     #SB6X-6,R0      ;MOVE OFFSET POINTER TO R0
5486  015770  000360  000006                   SWAB    6(R0)           ;TRY SWAB W/ MODE 6
5487  015774  022760  052652  000006          CMP     #52652,6(R0)    ;CHECK RESULT
5488  016002  001405                           BEQ     TS200
5489                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5490                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
5491                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
5492                                                                   ;           WHICH FOLLOWS W/ 764           <====
5493  016004                            SB6:
5494  016004  012742  000405                   MOV     #405,-(R2)      ;MOVE TO MAILBOX #  *******  405  *******
5495  016010  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5496  016012  000000                           HALT                    ;RESULT OF SWAB INCORRECT
5497                                                                   ;  OR SEQUENCE ERROR
5498  016014  000000                    SB6X:   0                       ;WORK LOCATION
5499
```

```
5500
5501                                    ;*****************************************************************************
5502                                    ;
5503                                    ;       THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION.  THIS TEST
5504                                    ;USES TWO LOCATIONS FOLLOWING THE TEST CODE:  A WORK LOCATION
5505                                    ;(SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD).  DATA IS MOVED
5506                                    ;TO THE WORK LOCATION.  R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
5507                                    ;OF THE ADDRESS POINTER.  THE DATA IS SWAB'ED USING A MODE 7
5508                                    ;INSTRUCTION WITH AN OFFSET OF +72.  THE DATA IS VERIFIED WITH A
5509                                    ;COMPARE.
5510                                    ;
5511                                    ;*****************************************************************************
5512                                    ;TEST 200       TEST MODE 7 W/ SWAB INST.
5513                                    ;*****************************************************************************
5514  016016  005212                    TS200:  INC     (R2)            ;UPDATE TEST NUMBER
5515  016020  022712  000200                   CMP     #200,(R2)       ;SEQUENCE ERROR?
5516  016024  001013                           BNE     SB7             ;BR TO ERROR HALT ON SEQ ERROR
5517  016026  012767  177400  000030          MOV     #177400,SB7X    ;MOVE PATTERN TO WORK LOCATION
5518  016034  012700  015774                   MOV     #SB7XAD-72,R0   ;MOVE OFFSET POINTER TO R0
5519  016040  000370  000072                   SWAB    @72(R0)         ;TRY SWAB MODE 7
5520  016044  027027  000072  000377          CMP     @72(R0),#377    ;CHECK RESULTS
5521  016052  001406                           BEQ     TS201
5522                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5523                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
5524                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
5525                                                                   ;           WHICH FOLLOWS W/ 764           <====
5526  016054                            SB7:
5527  016054  012742  000406                   MOV     #406,-(R2)      ;MOVE TO MAILBOX #  *******  406  *******
5528  016060  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5529  016062  000000                           HALT                    ;RESULT OF SWAB INCORRECT
5530                                                                   ;  OR SEQUENCE ERROR
5531  016064  000000                    SB7X:   0                       ;WORK LOCATION
5532  016066  016064                    SB7XAD: SB7X                    ;POINTER TO WORK LOCATION
5533
```

```
5534
5535                            ;*****************************************************************************************
5536                            ;
5537                            ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
5538                            ;BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
5539                            ;UTILIZES SEVERAL DIFFERENT TECHNIQUES.  THE CODE IS NOT EXECUTED
5540                            ;IN A LINEAR FASHION.  THE DIFFERENT MODES ARE EXECUTED IN ORDER
5541                            ;FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
5542                            ;FROGS THRU THE TEST CODE.  THE ORDER OF APPEARANCE OF THE CODE
5543                            ;IS:
5544                            ;       JMP MODE 1
5545                            ;       JMP MODE 3
5546                            ;       JMP MODE 2
5547                            ;       JMP MODE 4
5548                            ;       JMP MODE 6
5549                            ;       JMP MODE 5
5550                            ;       JMP MODE 7
5551                            ;AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
5552                            ;JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
5553                            ;       THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE.  EACH CODE
5554                            ;BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
5555                            ;THAT BLOCK.  A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK.  FOR
5556                            ;EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
5557                            ;OF THE PREVIOUS MODE 2 JUMP.  (ANY REGISTER CHANGES ARE VERIFIED
5558                            ;AND THE SEQUENCE CHECK IS MADE).  THEN THE REGISTERS ARE SETUP
5559                            ;FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
5560                            ;CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
5561                            ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
5562                            ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
5563                            ;THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
5564                            ;REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
5565                            ;
5566                            ;*****************************************************************************************
5567                            ;TEST 201       TEST THE JMP INSTRUCTION IN ALL MODES
5568                            ;*****************************************************************************************
5569  016070  005212            TS201:  INC     (R2)            ;UPDATE TEST NUMBER
5570  016072  022712  000201            CMP     #201,(R2)       ;SEQUENCE ERROR?
5571  016076  001150                    BNE     JMPCK+6         ;BR TO ERROR HALT ON SEQ ERROR
5572  016100  005067  000326            CLR     JMPSEQ          ;ESTABLISH A SEQUENCE CHECKER
5573  016104  012700  016164            MOV     #JMP2,R0        ;SET R0=JUMP TARGET
5574  016110  000110                    JMP     (R0)            ;TRY JMP MODE 1
5575  016112  022700  016114    JMP3:   CMP     #.+2,R0         ;CHECK RESULT OF MODE 2 JUMP
5576  016116  001404                    BEQ     JMP3A
5577                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5578                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5579                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5580                                                            ;           WHICH FOLLOWS W/ 767           <====
5581  016120  012742  000407            MOV     #407,-(R2)      ;MOVE TO MAILBOX #  *******  407  *******
5582  016124  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5583  016126  000000                    HALT                    ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
5584  016130  026727  000276  000001    JMP3A:  CMP     JMPSEQ,#1       ;MAKE SURE JMPS ARE IN SEQUENCE: JMPSEQ=1?
5585  016136  001404                    BEQ     JMP3B
5586                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5587                                                            ;           CONDITIONAL BRANCH INST. AND   <====
5588                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
5589                                                            ;           WHICH FOLLOWS W/ 757           <====
```

```
5590  016140  012742  000410              MOV     #410,-(R2)      ;MOVE TO MAILBOX #  *******  410  *******
5591  016144  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5592  016146  000000                      HALT                    ;SHOULD BE HERE FROM JMP MODE 2 ONLY
5593  016150  012700  016162      JMP3B:  MOV     #IJMP4,R0       ;POINT R0 TO INDIRECT JMP ADDR.
5594  016154  005267  000252              INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5595  016160  000130                      JMP     @(R0)+          ;TRY JMP MODE 3
5596  016162  016214      IJMP4:  JMP4                            ;ADDRESS INDIRECT JUMP
5597
5598  016164  005767  000242      JMP2:   TST     JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
5599  016170  001404                      BEQ     JMP2A
5600                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5601                                                              ;           CONDITIONAL BRANCH INST. AND   <====
5602                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
5603                                                              ;           WHICH FOLLOWS W/ 742           <====
5604  016172  012742  000411              MOV     #411,-(R2)      ;MOVE TO MAILBOX #  *******  411  *******
5605  016176  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5606  016200  000000                      HALT                    ;SHOULD BE HERE FROM JMP MODE 1 ONLY
5607  016202  005267  000224      JMP2A:  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5608  016206  012700  016112              MOV     #JMP3,R0        ;SET R0=JUMP TARGET
5609  016212  000120                      JMP     (R0)+           ;TRY A JUMP MODE 2 TO "JMP3"
5610  016214  022700  016164      JMP4:   CMP     #IJMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
5611  016220  001404                      BEQ     JMP4A
5612                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5613                                                              ;           CONDITIONAL BRANCH INST. AND   <====
5614                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
5615                                                              ;           WHICH FOLLOWS W/ 726           <====
5616  016222  012742  000412              MOV     #412,-(R2)      ;MOVE TO MAILBOX #  *******  412  *******
5617  016226  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5618  016230  000000                      HALT                    ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5619  016232  022767  000002  000172  JMP4A:  CMP     #2,JMPSEQ       ;CHECK JUMP SEQUENCE: JMPSEQ=2?
5620  016240  001404                      BEQ     JMP4B
5621                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5622                                                              ;           CONDITIONAL BRANCH INST. AND   <====
5623                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
5624                                                              ;           WHICH FOLLOWS W/ 716           <====
5625  016242  012742  000413              MOV     #413,-(R2)      ;MOVE TO MAILBOX #  *******  413  *******
5626  016246  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5627  016250  000000                      HALT                    ;SHOULD BE ONLY FROM MODE 3 JUMP
5628  016252  012700  016322      JMP4B:  MOV     #JMP5+2,R0      ;SET UP POINTER TO JUMP TARGET
5629  016256  005267  000150              INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
5630  016262  000140                      JMP     -(R0)           ;TRY JUMP MODE 4 TO "JMP4"
5631
5632  016264  022767  000004  000140  JMP6:   CMP     #4,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5633  016272  001404                      BEQ     JMP6A
5634                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5635                                                              ;           CONDITIONAL BRANCH INST. AND   <====
5636                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
5637                                                              ;           WHICH FOLLOWS W/ 701           <====
5638  016274  012742  000414              MOV     #414,-(R2)      ;MOVE TO MAILBOX #  *******  414  *******
5639  016300  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5640  016302  000000                      HALT                    ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
5641  016304  012700  016752      JMP6A:  MOV     #JMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
5642  016310  005267  000116              INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5643  016314  000160  177402              JMP     -376(R0)        ;TRY MODE 6 JUMP
5644
5645  016320  022767  000003  000104  JMP5:   CMP     #3,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
```

```
5646  016326  001404                          BEQ     JMP5A
5647                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5648                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5649                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5650                                                  ;           WHICH FOLLOWS W/ 663           <====
5651  016330  012742  000415                        MOV     #415,-(R2)      ;MOVE TO MAILBOX #  ******* 415 *******
5652  016334  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5653  016336  000000                                HALT                    ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5654  016340  012700  016354          JMP5A:        MOV     #1JMP5+2,R0     ;SET UP POINTER TO INDIRECT JUMP ADDR.
5655  016344  005267  000062                        INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5656  016350  000150                                JMP     @-(R0)          ;TRY JUMP MODE 5 TO "JMP6"
5657  016352  016264          1JMP5:        JMP6                            ;INDIRECT ADDRESS POINTER
5658
5659  016354  022767  000005  000050  JMP7:        CMP     #5,JMPSEQ       ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5660  016362  001404                          BEQ     JMP7A
5661                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5662                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5663                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5664                                                  ;           WHICH FOLLOWS W/ 645           <====
5665  016364  012742  000416                        MOV     #416,-(R2)      ;MOVE TO MAILBOX #  ******* 416 *******
5666  016370  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5667  016372  000000                                HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5668  016374  012700  016420          JMP7A:        MOV     #1JMP+10,R0     ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5669  016400  005267  000026                        INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
5670  016404  000170  177770                        JMP     @-10(R0)        ;TRY MODE 7 JUMP
5671  016410  016412          1JMP:         JMPCK                           ;INDIRECT ADDRESS
5672
5673  016412  026727  000014  000006  JMPCK:        CMP     JMPSEQ,#6       ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5674  016420  001405                          BEQ     TS202
5675                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5676                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5677                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5678                                                  ;           WHICH FOLLOWS W/ 626           <====
5679  016422  012742  000417                        MOV     #417,-(R2)      ;MOVE TO MAILBOX #  ******* 417 *******
5680  016426  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5681  016430  000000                                HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5682                                                                          ;  OR SEQUENCE ERROR
5683  016432  000000                  JMPSEQ:       0
```

```
5684
5685                                    ;*****************************************************************************
5686                                    ;
5687                                    ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
5688                                    ;THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
5689                                    ;IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST).  EACH
5690                                    ;BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
5691                                    ;CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
5692                                    ;THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
5693                                    ;SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
5694                                    ;REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
5695                                    ;SUCCESSFUL.  R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
5696                                    ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
5697                                    ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
5698                                    ;THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
5699                                    ;REGISTER SAVED).
5700                                    ;
5701                                    ;*****************************************************************************
5702                                    ;TEST 202       TEST JSR INSTRUCTION W/ ALL MODES
5703                                    ;*****************************************************************************
5704  016434  005212                    TS202:  INC     (R2)            ;UPDATE TEST NUMBER
5705  016436  022712  000202                   CMP     #202,(R2)       ;SEQUENCE ERROR?
5706  016442  001001                           BNE     JSR0            ;BR TO ERROR HALT ON SEQ ERROR
5707  016444  000402                           BR      JSR1
5708  016446  000137  017102            JSR0:   JMP     @#JSRCK1
5709
5710  016452  012706  001000            JSR1:   MOV     #STBOT,R6       ;SET STACK POINTER
5711  016456  012700  016564                    MOV     #JSR2,R0        ;SET TARGET ADDRESS
5712  016462  005037  017062                    CLR     @#JSRSEQ        ;INITIALIZE SEQUENCE CHECKER
5713  016466  005001                           CLR     R1              ;INITIALIZE R1
5714  016470  005101                           COM     R1
5715  016472  004110                           JSR     R1,(R0)         ;TRY JSR MODE 1
5716                                                                    ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
5717                                                                    ;           FOLLOWING W/ 774               <====
5718  016474                            JSR1A:
5719  016474  012742  000420                   MOV     #420,-(R2)      ;MOVE TO MAILBOX #  *******  420  *******
5720  016500  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5721  016502  000000                           HALT                    ;JSR MODE 1 FAILED
5722
5723  016504  022737  000001  017062  JSR3:   CMP     #1,@# JSRSEQ    ;CHECK SEQUENCE: JSRSEQ=1?
5724  016512  001014                           BNE     JSR3A           ;BRANCH IF OUT OF SEQUENCE
5725  016514  020127  016646                   CMP     R1,#JSR4        ;PROPER PC SAVED?
5726  016520  001011                           BNE     JSR3A           ;BRANCH IF PC WRONG
5727  016522  022706  000776                   CMP     #STBOT-2,R6     ;STACK POINTER DECREMENTED?
5728  016526  001006                           BNE     JSR3A           ;BRANCH IF SP WRONG
5729  016530  022716  125252                   CMP     #125252,(R6)    ;REG SAVED ON STACK?
5730  016534  001003                           BNE     JSR3A           ;BRANCH IF REG. NOT SAVED
5731  016536  022700  016506                   CMP     #JSR3+2,R0      ;MODE 2 INCREMENT CORRECT?
5732  016542  001404                           BEQ     JSR3B
5733                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5734                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5735                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5736                                                                    ;           WHICH FOLLOWS W/ 737           <====
5737  016544                            JSR3A:
5738  016544  012742  000421                   MOV     #421,-(R2)      ;MOVE TO MAILBOX #  *******  421  *******
5739  016550  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
5740  016552  000000                        HALT                    ;JSR MODE 3 MALFUNCTIONED
5741  016554  005237  017062        JSR3B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5742  016560  004137  016646                JSR     R1,@#JSR4       ;TRY JSR MODE 4
5743
5744  016564  005737  017062        JSR2:   TST     @#JSRSEQ        ;CHECK SEQUENCE: JSRSEQ=0?
5745  016570  001011                        BNE     JSR2A           ;BRANCH IF OUT OF SEQUENCE
5746  016572  020127  016474                CMP     R1,#JSR1A       ;PROPER PC SAVED?
5747  016576  001006                        BNE     JSR2A           ;BRANCH IF PC WRONG
5748  016600  022706  000776                CMP     #STBOT-2,R6     ;R6 DECREMENT?
5749  016604  001003                        BNE     JSR2A           ;BRANCH IF R6 IS INCORRECT
5750  016606  021627  177777                CMP     (R6),#-1        ;REGISTER SAVED?
5751  016612  001404                        BEQ     JSR2B
5752                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5753                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5754                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5755                                                                ;           WHICH FOLLOWS W/ 713           <====
5756  016614                          JSR2A:
5757  016614  012742  000422                MOV     #422,-(R2)      ;MOVE TO MAILBOX #  *******  422  *******
5758  016620  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5759  016622  000000                        HALT                    ;JSR MODE 1 MALFUNCTIONED
5760  016624  012706  001000        JSR2B:  MOV     #STBOT,R6       ;INITIALIZE R6
5761  016630  012701  125252                MOV     #125252,R1      ;INITIALIZE R1
5762  016634  005237  017062                INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5763  016640  012700  016504                MOV     #JSR3,R0        ;SET TARGET ADDRESS
5764  016644  004120                        JSR     R1,(R0)+        ;TRY JSR MODE 2
5765
5766  016646  022737  000002  017062 JSR4:   CMP     #2,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=2?
5767  016654  001003                        BNE     JSR4A           ;BRANCH IF OUT OF SEQUENCE
5768  016656  022701  016564                CMP     #JSR2,R1        ;PROPER PC SAVED?
5769  016662  001404                        BEQ     JSR4B
5770                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5771                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5772                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5773                                                                ;           WHICH FOLLOWS W/ 667           <====
5774  016664                          JSR4A:
5775  016664  012742  000423                MOV     #423,-(R2)      ;MOVE TO MAILBOX #  *******  423  *******
5776  016670  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5777  016672  000000                        HALT                    ;JSR MODE 3 MALFUNCTIONED
5778  016674  005237  017062        JSR4B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5779  016700  012700  016754                MOV     #JSR5+2,R0      ;SET TARGET ADDRESS
5780  016704  004140                        JSR     R1,-(R0)        ;TRY JSR MODE 4
5781
5782  016706  022767  000004  000146 JSR6:   CMP     #4,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=4?
5783  016714  001006                        BNE     JSR6A           ;BRANCH IF OUT OF SEQUENCE
5784  016716  022701  017020                CMP     #JSR7,R1        ;PROPER PC SAVED?
5785  016722  001003                        BNE     JSR6A           ;BRANCH IF PC WRONG
5786  016724  022700  017056                CMP     #JSR6AD,R0      ;MODE 5 REGISTER CORRECT?
5787  016730  001404                        BEQ     JSR6B
5788                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5789                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5790                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5791                                                                ;           WHICH FOLLOWS W/ 644           <====
5792  016732                          JSR6A:
5793  016732  012742  000424                MOV     #424,-(R2)      ;MOVE TO MAILBOX #  *******  424  *******
5794  016736  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5795  016740  000000                        HALT                    ;JSR MODE 5 FAILED
```

```
5796  016742  005237  017062          JSR6B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5797  016746  004167  000046                  JSR     R1,JSR7         ;TRY JSR MODE 6
5798  016752  022767  000003  000102  JSR5:   CMP     #3,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=3?
5799  016760  001006                          BNE     JSR5A           ;BRANCH IF OUT OF SEQUENCE
5800  016762  022701  016706                  CMP     #JSR6,R1        ;PROPER PC SAVED?
5801  016766  001003                          BNE     JSR5A           ;BRANCH IF PC WRONG
5802  016770  022700  016752                  CMP     #JSR5,R0        ;CHECK MODE 4 REGISTER
5803  016774  001404                          BEQ     JSR5B
5804                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5805                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5806                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5807                                                                  ;           WHICH FOLLOWS W/ 622           <====
5808  016776                          JSR5A:
5809  016776  012742  000425                  MOV     #425,-(R2)      ;MOVE TO MAILBOX #  *******  425  *******
5810  017002  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5811  017004  000000                          HALT                    ;JSR MODE 4 MALFUNCTIONED
5812  017006  005237  017062          JSR5B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5813  017012  012700  017060                  MOV     #JSR6AD+2,R0    ;POINT R0 TO TARGET ADDRESS
5814  017016  004150                          JSR     R1,@-(R0)       ;TRY JSR MODE 5
5815
5816  017020  022737  000005  017062  JSR7:   CMP     #5,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=5?
5817  017026  001003                          BNE     JSR7A           ;BRANCH IF OUT OF SEQUENCE
5818  017030  022701  016752                  CMP     #JSR5,R1        ;PROPER PC SAVED?
5819  017034  001404                          BEQ     JSR7B
5820                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5821                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5822                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5823                                                                  ;           WHICH FOLLOWS W/ 602           <====
5824  017036                          JSR7A:
5825  017036  012742  000426                  MOV     #426,-(R2)      ;MOVE TO MAILBOX #  *******  426  *******
5826  017042  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5827  017044  000000                          HALT                    ;JSR MODE 6 FAILED
5828  017046  005237  017062          JSR7B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5829  017052  004177  000002                  JSR     R1,@JSRCKAD     ;TRY JSR MODE 7
5830
5831  017056  016706                  JSR6AD: JSR6                    ;MODE 5 TARGET ADDRESS
5832  017060  017064                  JSRCKAD:JSRCK                   ;MODE 7 TARGET ADDRESS
5833  017062  000000                  JSRSEQ: 0                       ;SEQUENCE CHECKER
5834
5835  017064  022767  000006  177770  JSRCK:  CMP     #6,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=6?
5836  017072  001003                          BNE     JSRCK1          ;BRANCH IF OUT OF SEQUENCE
5837  017074  022701  017056                  CMP     #JSR6AD,R1      ;PROPER PC SAVED?
5838  017100  001404                          BEQ     TS203
5839                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5840                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
5841                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5842                                                                  ;           WHICH FOLLOWS W/ 560           <====
5843  017102                          JSRCK1:
5844  017102  012742  000427                  MOV     #427,-(R2)      ;MOVE TO MAILBOX #  *******  427  *******
5845  017106  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5846  017110  000000                          HALT                    ;JSR MODE 7 MALFUNCTIONED
5847                                                                  ;  OR SEQUENCE ERROR
5848
5849
```

```
5850                                    ;*****************************************************************************
5851                                    ;
5852                                    ;       THIS TEST VERIFIES THE RTS INSTRUCTION.  THE STACK POINTER
5853                                    ;IS INITIALIZED AND A TEST PATTERN STORED ON STACK.  R0 IS LOADED
5854                                    ;WITH RETURN ADDRESS.  AN RTS IS EXECUTED, AND, AT THE TARGET
5855                                    ;ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
5856                                    ;STACK.
5857                                    ;*****************************************************************************
5858                                    ;TEST 203       TEST RTS INSTRUCTION
5859                                    ;*****************************************************************************
5860  017112  005212                TS203:  INC     (R2)            ;UPDATE TEST NUMBER
5861  017114  022712  000203               CMP     #203,(R2)       ;SEQUENCE ERROR?
5862  017120  001016                       BNE     TS204-10        ;BR TO ERROR HALT ON SEQ ERROR
5863  017122  012706  001000               MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
5864  017126  012746  052525               MOV     #52525,-(R6)    ;INITIALIZE TOP OF STACK
5865  017132  012700  017150               MOV     #RTS1,R0        ;INITIALIZE RETURN REGISTER
5866  017136  000200                       RTS     R0              ;TRY RTS THROUGH R0
5867                                                                ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <====
5868                                                                ;           FOLLOWING W/ 770               <====
5869  017140  012742  000430               MOV     #430,-(R2)      ;MOVE TO MAILBOX #  *******  430  *******
5870  017144  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5871  017146  000000                       HALT                    ;RTS FAILED
5872  017150  022700  052525        RTS1:  CMP     #52525,R0       ;CHECK THAT R0 RESTORED FROM STACK
5873  017154  001404                       BEQ     TS204
5874                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5875                                                                ;           CONDITIONAL BRANCH INST. AND   <====
5876                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
5877                                                                ;           WHICH FOLLOWS W/ 761           <====
5878  017156  012742  000431               MOV     #431,-(R2)      ;MOVE TO MAILBOX #  *******  431  *******
5879  017162  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5880  017164  000000                       HALT                    ;RTS MALFUNCTIONED
5881                                                                ;  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
                                ;*****************************************************************************
017166  005212          TS204:  INC     (R2)            ;UPDATE TEST NUMBER
017170  022712  000204          CMP     #204,(R2)       ;SEQUENCE ERROR?
017174  001022                  BNE     TS205-10        ;BR TO ERROR HALT ON SEQ ERROR
017176  000277                  SCC                     ;CC=0110
017200  000251                  +CLN!CLC
017202  012700  100000          MOV     #100000,R0      ;CC=1000
017206  101402                  BLOS    MOV1
017210  102401                  BVS     MOV1
017212  100404                  BMI     MOV2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
017214                  MOV1:
017214  012742  000432          MOV     #432,-(R2)      ;MOVE TO MAILBOX #  *******  432  *******
017220  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017222  000000                  HALT                    ;MOV DID NOT SET CC'S CORRECTLY
017224  000277          MOV2:   SCC                     ;CC=1011
017226  000244                  CLZ
017230  012700  000000          MOV     #0,R0           ;CC=0101
017234  101002                  BHI     MOV3            ;C OR Z = 0?
017236  102401                  BVS     MOV3            ;V=1?
017240  100004                  BPL     TS205
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 755           <====
017242                  MOV3:
017242  012742  000433          MOV     #433,-(R2)      ;MOVE TO MAILBOX #  *******  433  *******
017246  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017250  000000                  HALT                    ;MOV DID NOT SET CC'S CORRECTLY
                                                        ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 205       TEST BIT INSTRUCTION
                                ;*****************************************************************************
017252  005212          TS205:  INC     (R2)            ;UPDATE TEST NUMBER
017254  022712  000205          CMP     #205,(R2)       ;SEQUENCE ERROR?
017260  001024                  BNE     TS206-10        ;BR TO ERROR HALT ON SEQ ERROR
```

```
5938  017262  012700  100001          MOV     #100001,R0
5939  017266  000277                  SCC
5940  017270  000251                  +CLN!CLC
5941  017272  032700  100000          BIT     #100000,R0      ;CC=1000
5942  017276  101402                  BLOS    BIT1
5943  017300  102401                  BVS     BIT1
5944  017302  100404                  BMI     BIT2
5945                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5946                                  ;           CONDITIONAL BRANCH INST. AND   <====
5947                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5948                                  ;           WHICH FOLLOWS W/ 766           <====
5949  017304                  BIT1:
5950  017304  012742  000434          MOV     #434,-(R2)      ;MOVE TO MAILBOX #  ******* 434 *******
5951  017310  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5952  017312  000000                  HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5953
5954  017314  000277          BIT2:   SCC                     ;CC=1011
5955  017316  000244                  CLZ
5956  017320  032700  077776          BIT     #77776,R0       ;CC=0101
5957  017324  101002                  BHI     BIT3
5958  017326  102401                  BVS     BIT3
5959  017330  100004                  BPL     TS206
5960                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5961                                  ;           CONDITIONAL BRANCH INST. AND   <====
5962                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5963                                  ;           WHICH FOLLOWS W/ 753           <====
5964  017332                  BIT3:
5965  017332  012742  000435          MOV     #435,-(R2)      ;MOVE TO MAILBOX #  ******* 435 *******
5966  017336  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5967  017340  000000                  HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5968                                                          ;  OR SEQUENCE ERROR
5969                  ;*****************************************************************************************
5970                  ;TEST 206       TEST BIC INSTRUCTION
5971                  ;*****************************************************************************************
5972  017342  005212          TS206:  INC     (R2)            ;UPDATE TEST NUMBER
5973  017344  022712  000206          CMP     #206,(R2)       ;SEQUENCE ERROR?
5974  017350  001024                  BNE     TS207-10        ;BR TO ERROR HALT ON SEQ ERROR
5975  017352  012700  177777          MOV     #177777,R0
5976  017356  000277                  SCC                     ;CC=0110
5977  017360  000251                  +CLN!CLC
5978  017362  042700  077777          BIC     #77777,R0       ;CC=1000
5979  017366  101402                  BLOS    BIC1
5980  017370  102401                  BVS     BIC1
5981  017372  100404                  BMI     BIC2
5982                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5983                                  ;           CONDITIONAL BRANCH INST. AND   <====
5984                                  ;           REPLACE THE MOVE INSTRUCTION   <====
5985                                  ;           WHICH FOLLOWS W/ 766           <====
5986  017374                  BIC1:
5987  017374  012742  000436          MOV     #436,-(R2)      ;MOVE TO MAILBOX #  ******* 436 *******
5988  017400  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5989  017402  000000                  HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5990  017404  000277          BIC2:   SCC                     ;CC=1011
5991  017406  000244                  CLZ
5992  017410  042700  100000          BIC     #100000,R0      ;CC=0101
5993  017414  101002                  BHI     BIC3
```

```
5994  017416  102401                BVS     BIC3
5995  017420  100004                BPL     TS207
5996                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5997                                ;           CONDITIONAL BRANCH INST. AND   <====
5998                                ;           REPLACE THE MOVE INSTRUCTION   <====
5999                                ;           WHICH FOLLOWS W/ 753           <====
6000  017422                BIC3:
6001  017422  012742  000437        MOV     #437,-(R2)      ;MOVE TO MAILBOX #  ******* 437 *******
6002  017426  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6003  017430  000000                HALT                    ;BIC DID NOT SET CC'S CORRECTLY
6004                                                        ;  OR SEQUENCE ERROR
6005  ;*************************************************************************************
6006  ;TEST 207       TEST BIS INSTRUCTION
6007  ;*************************************************************************************
6008  017432  005212        TS207:  INC     (R2)            ;UPDATE TEST NUMBER
6009  017434  022712  000207        CMP     #207,(R2)       ;SEQUENCE ERROR?
6010  017440  001025                BNE     TS210-10        ;BR TO ERROR HALT ON SEQ ERROR
6011  017442  005000                CLR     R0              ;R0=0
6012  017444  000277                SCC                     ;CC=1010
6013  017446  000251                +CLN!CLC
6014  017450  052700  000000        BIS     #0,R0           ;CC=0100  R0=0
6015  017454  103403                BCS     BIS1
6016  017456  102402                BVS     BIS1
6017  017460  100401                BMI     BIS1
6018  017462  001404                BEQ     BIS2
6019                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6020                                ;           CONDITIONAL BRANCH INST. AND   <====
6021                                ;           REPLACE THE MOVE INSTRUCTION   <====
6022                                ;           WHICH FOLLOWS W/ 766           <====
6023  017464                BIS1:
6024  017464  012742  000440        MOV     #440,-(R2)      ;MOVE TO MAILBOX #  ******* 440 *******
6025  017470  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6026  017472  000000                HALT                    ;BIS DID NOT SET CC'S CORRECTLY
6027  017474  000277        BIS2:   SCC                     ;CC=0111
6028  017476  000250                CLN
6029  017500  052700  177777        BIS     #177777,R0      ;CC=1001
6030  017504  103003                BCC     BIS3
6031  017506  102402                BVS     BIS3
6032  017510  001401                BEQ     BIS3
6033  017512  100404                BMI     TS210
6034                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6035                                ;           CONDITIONAL BRANCH INST. AND   <====
6036                                ;           REPLACE THE MOVE INSTRUCTION   <====
6037                                ;           WHICH FOLLOWS W/ 752           <====
6038  017514                BIS3:
6039  017514  012742  000441        MOV     #441,-(R2)      ;MOVE TO MAILBOX #  ******* 441 *******
6040  017520  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6041  017522  000000                HALT                    ;BIS DID NOT SET CC'S CORRECTLY
6042                                ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
                        ;DEC INSTRUCTIONS.  THESE INSTRUCTIONS BOTH EFFECT THE C AND V
                        ;BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
                        ;UPON THE DATA RESULTS.  THE SAME PROCEDURE IS USED.  THE CONDITION
                        ;CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
                        ;RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
                        ;THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
                        ;DIFFERENT COMBINATIONS OF THE C AND V BITS.
                        ;
                        ;*****************************************************************************
                        ;TEST 210       TEST INC INSTRUCTION
                        ;*****************************************************************************
017524  005212  TS210:  INC     (R2)            ;UPDATE TEST NUMBER
017526  022712  000210  CMP     #210,(R2)       ;SEQUENCE ERROR?
017532  001037          BNE     TS211-10        ;BR TO ERROR HALT ON SEQ ERROR
017534  012700  077777  MOV     #077777,R0      ;R0=077777
017540  000257          CCC                     ;CC=0100
017542  000264          SEZ
017544  005200          INC     R0              ;CC=1010   R0=10000
017546  101402          BLOS    INC1
017550  100001          BPL     INC1
017552  102404          BVS     INC2
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 767           <====
017554  INC1:
017554  012742  000442          MOV     #442,-(R2)      ;MOVE TO MAILBOX #  *******  442  *******
017560  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017562  000000                  HALT                    ;INC DID NOT SET CC'S CORRECTLY
017564  052700  077777  INC2:   BIS     #77777,R0       ;R0=177777
017570  000261                  SEC                     ;CC=1011
017572  000244                  CLZ
017574  005200                  INC     R0              ;CC=0101   R0=0
017576  100403                  BMI     INC3
017600  102402                  BVS     INC3
017602  103001                  BCC     INC3
017604  001404                  BEQ     INC4
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 752           <====
017606  INC3:
017606  012742  000443          MOV     #443,-(R2)      ;MOVE TO MAILBOX #  *******  443  *******
017612  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017614  000000                  HALT                    ;INC DID NOT SET CC'S CORRECTLY
017616  000277  INC4:   SCC                     ;CC=1110
017620  000241          CLC
017622  005200          INC     R0              ;CC=0000   R0=1
017624  101402          BLOS    INC5
017626  100401          BMI     INC5
017630  100004          BPL     TS211
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 740           <====
017632                          INC5:
017632  012742  000444                MOV     #444,-(R2)      ;MOVE TO MAILBOX #  *******  444  *******
017636  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017640  000000                      HALT                    ;INC DID NOT SET CC'S CORRECTLY
                                                            ;  OR SEQUENCE ERROR

                                ;*************************************************************************************
                                ;TEST 211       TEST DEC INSTRUCTION
                                ;*************************************************************************************
017642  005212                  TS211:  INC     (R2)            ;UPDATE TEST NUMBER
017644  022712  000211                  CMP     #211,(R2)       ;SEQUENCE ERROR?
017650  001051                        BNE     TS212-10        ;BR TO ERROR HALT ON SEQ ERROR
017652  012700  000002                MOV     #2,R0           ;R0=2
017656  000277                      SCC                     ;CC=1111
017660  005300                      DEC     R0              ;CC=0001   R0=1
017662  100403                      BMI     DEC1
017664  001402                      BEQ     DEC1
017666  102401                      BVS     DEC1
017670  103404                      BCS     DEC2
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 767           <====
017672                          DEC1:
017672  012742  000445                MOV     #445,-(R2)      ;MOVE TO MAILBOX #  *******  445  *******
017676  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017700  000000                      HALT                    ;DEC DID NOT SET CC'S CORRECTLY
017702  000261                  DEC2:   SEC                     ;CC=1011
017704  000244                      CLZ
017706  005300                      DEC     R0              ;CC=0101   R0=0
017710  101002                      BHI     DEC3
017712  100401                      BMI     DEC3
017714  102004                      BVC     DEC4
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 755           <====
017716                          DEC3:
017716  012742  000446                MOV     #446,-(R2)      ;MOVE TO MAILBOX #  *******  446  *******
017722  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017724  000000                      HALT                    ;DEC DID NOT SET CC'S CORRECTLY
017726  000277                  DEC4:   SCC                     ;CC=0110
017730  000251                      +CLN!CLC
017732  005300                      DEC     R0              ;CC=1000   R0=177777
017734  101402                      BLOS    DEC5
017736  102401                      BVS     DEC5
017740  100404                      BMI     DEC6
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 743           <====
017742                          DEC5:
```

```
6155  017742  012742  000447          MOV     #447,-(R2)      ;MOVE TO MAILBOX #  *******  447  *******
6156  017746  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6157  017750  000000                  HALT                    ;DEC DID NOT SET CC'S CORRECTLY
6158  017752  042700  077777  DEC6:   BIC     #77777,R0       ;R0=100000
6159  017756  000277                  SCC                     ;CC=0101
6160  017760  000252                  +CLN!CLV
6161  017762  005300                  DEC     R0              ;CC=1011   R0=77777
6162  017764  100403                  BMI     DEC7            ;CC=0011
6163  017766  001402                  BEQ     DEC7
6164  017770  102001                  BVC     DEC7
6165  017772  103404                  BCS     TS212
6166                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6167                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6168                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6169                                                          ;           WHICH FOLLOWS W/ 726           <====
6170  017774                  DEC7:
6171  017774  012742  000450          MOV     #450,-(R2)      ;MOVE TO MAILBOX #  *******  450  *******
6172  020000  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6173  020002  000000                  HALT                    ;DEC DID NOT SET CC'S CORRECTLY
6174                                                          ;  OR SEQUENCE ERROR
6175
```

```
                                        H 10
6176
6177                            ;*****************************************************************************************
6178                            ;
6179                            ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
6180                            ;TST, AND SWAB INSTRUCTIONS.  THESE THREE INSTRUCTIONS ALL LEAVE
6181                            ;THE C AND V BITS CLEARED.  AGAIN, THE CONDITION CODES ARE PRESET,
6182                            ;THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
6183                            ;BRANCH INSTRUCTIONS.  THE PROCEDURE IS REPEATED TO PRODUCE OTHER
6184                            ;COMBINATIONS OF CONDITION CODES.
6185                            ;
6186                            ;*****************************************************************************************
6187                            ;TEST 212       TEST CLR INSTRUCTION
6188                            ;*****************************************************************************************
6189  020004  005212            TS212:  INC     (R2)            ;UPDATE TEST NUMBER
6190  020006  022712  000212            CMP     #212,(R2)       ;SEQUENCE ERROR?
6191  020012  001007                    BNE     TS213-10        ;BR TO ERROR HALT ON SEQ ERROR
6192  020014  000277                    SCC                     ;CC=1011
6193  020016  000244                    CLZ
6194  020020  005000                    CLR     R0              ;CC=0100   R0=0
6195  020022  100403                    BMI     CLR1
6196  020024  102402                    BVS     CLR1
6197  020026  103401                    BCS     CLR1
6198  020030  001404                    BEQ     TS213
6199                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6200                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6201                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6202                                                            ;           WHICH FOLLOWS W/ 770           <====
6203  020032                    CLR1:
6204  020032  012742  000451            MOV     #451,-(R2)      ;MOVE TO MAILBOX #  *******  451  *******
6205  020036  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6206  020040  000000                    HALT                    ;CLR DID NOT SET CC'S CORRECTLY
6207                                                            ;  OR SEQUENCE ERROR
6208
6209                            ;*****************************************************************************************
6210                            ;TEST 213       TEST TST INSTRUCTION
6211                            ;*****************************************************************************************
6212  020042  005212            TS213:  INC     (R2)            ;UPDATE TEST NUMBER
6213  020044  022712  000213            CMP     #213,(R2)       ;SEQUENCE ERROR?
6214  020050  001022                    BNE     TS214-10        ;BR TO ERROR HALT ON SEQ ERROR
6215  020052  000277                    SCC                     ;CC=1011
6216  020054  000244                    CLZ
6217  020056  005700                    TST     R0              ;CC=0100
6218  020060  100403                    BMI     TEST1
6219  020062  102402                    BVS     TEST1
6220  020064  103401                    BCS     TEST1
6221  020066  001404                    BEQ     TEST2
6222                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6223                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6224                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6225                                                            ;           WHICH FOLLOWS W/ 770           <====
6226  020070                    TEST1:
6227  020070  012742  000452            MOV     #452,-(R2)      ;MOVE TO MAILBOX #  *******  452  *******
6228  020074  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6229  020076  000000                    HALT                    ;TEST DID NOT SET CC'S CORRECTLY
6230  020100  005300            TEST2:  DEC     R0              ;MAKE R0 NEGATIVE
6231  020102  000277                    SCC                     ;CC=0111
```

```
6232  020104  000250                        CLN
6233  020106  005700                        TST     R0              ;CC=1000
6234  020110  101402                        BLOS    TEST3
6235  020112  102401                        BVS     TEST3
6236  020114  100404                        BMI     TS214
6237                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6238                                        ;           CONDITIONAL BRANCH INST. AND   <====
6239                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6240                                        ;           WHICH FOLLOWS W/ 755           <====
6241  020116                        TEST3:
6242  020116  012742  000453                MOV     #453,-(R2)      ;MOVE TO MAILBOX #  *******  453  *******
6243  020122  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6244  020124  000000                        HALT                    ;TEST DID NOT SET CC'S CORRECTLY
6245                                        ;  OR SEQUENCE ERROR
6246                                ;*****************************************************************************
6247                                ;TEST 214       TEST SWAB INSTRUCTION
6248                                ;*****************************************************************************
6249  020126  005212                TS214:  INC     (R2)            ;UPDATE TEST NUMBER
6250  020130  022712  000214                CMP     #214,(R2)       ;SEQUENCE ERROR?
6251  020134  001023                        BNE     TS215-10        ;BR TO ERROR HALT ON SEQ ERROR
6252  020136  012700  170000                MOV     #170000,R0      ;R0=170000
6253  020142  000277                        SCC                     ;CC=0111
6254  020144  000250                        CLN
6255  020146  000300                        SWAB    R0              ;CC=1000   R0=360
6256  020150  101402                        BLOS    SWB1
6257  020152  102401                        BVS     SWB1
6258  020154  100404                        BMI     SWB2
6259                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6260                                        ;           CONDITIONAL BRANCH INST. AND   <====
6261                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6262                                        ;           WHICH FOLLOWS W/ 767           <====
6263  020156                        SWB1:
6264  020156  012742  000454                MOV     #454,-(R2)      ;MOVE TO MAILBOX #  *******  454  *******
6265  020162  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6266  020164  000000                        HALT                    ;SWAB DID NOT SET CC'S CORRECTLY
6267  020166  000277                SWB2:   SCC                     ;CC=1011
6268  020170  000244                        CLZ
6269  020172  000300                        SWAB    R0              ;CC=0100   R0=170000
6270  020174  102403                        BVS     SWB3
6271  020176  103402                        BCS     SWB3
6272  020200  100401                        BMI     SWB3
6273  020202  001404                        BEQ     TS215
6274                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6275                                        ;           CONDITIONAL BRANCH INST. AND   <====
6276                                        ;           REPLACE THE MOVE INSTRUCTION   <====
6277                                        ;           WHICH FOLLOWS W/ 754           <====
6278  020204                        SWB3:
6279  020204  012742  000455                MOV     #455,-(R2)      ;MOVE TO MAILBOX #  *******  455  *******
6280  020210  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6281  020212  000000                        HALT                    ;
```

```
6282
6283                            ;*****************************************************************************
6284                            ;
6285                            ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE ADD AND
6286                            ;ADC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE C AND
6287                            ;V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
6288                            ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
6289                            ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
6290                            ;BRANCHES.  THES PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
6291                            ;DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.
6292                            ;
6293                            ;*****************************************************************************
6294                            ;TEST 215       TEST ADD INSTRUCTION
6295                            ;*****************************************************************************
6296  020214  005212            TS215:  INC     (R2)            ;UPDATE TEST NUMBER
6297  020216  022712  000215            CMP     #215,(R2)       ;SEQUENCE ERROR?
6298  020222  001062                    BNE     TS216-10        ;BR TO ERROR HALT ON SEQ ERROR
6299  020224  012700  040000            MOV     #40000,R0       ;R0=40000
6300  020230  000277                    SCC                     ;CC=1111
6301  020232  062700  030000            ADD     #30000,R0       ;CC=0000   R0=70000
6302  020236  101402                    BLOS    ADD1
6303  020240  102401                    BVS     ADD1
6304  020242  100004                    BPL     ADD2
6305                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6306                                                    ;           CONDITIONAL BRANCH INST. AND   <====
6307                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
6308                                                    ;           WHICH FOLLOWS W/ 767           <====
6309  020244                    ADD1:
6310  020244  012742  000456            MOV     #456,-(R2)      ;MOVE TO MAILBOX #  ******* 456 *******
6311  020250  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6312  020252  000000                    HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6313  020254  000264            ADD2:   SEZ                     ;CC=0100
6314
6315  020256  062700  010000            ADD     #10000,R0       ;CC=1010   40=100000
6316  020262  101402                    BLOS    ADD3
6317  020264  102001                    BVC     ADD3
6318  020266  100404                    BMI     ADD4
6319                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6320                                                    ;           CONDITIONAL BRANCH INST. AND   <====
6321                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
6322                                                    ;           WHICH FOLLOWS W/ 755           <====
6323  020270                    ADD3:
6324  020270  012742  000457            MOV     #457,-(R2)      ;MOVE TO MAILBOX #  ******* 457 *******
6325  020274  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6326  020276  000000                    HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6327  020300  000257            ADD4:   CCC                     ;CC=1000
6328  020302  000270                    SEN
6329  020304  062700  100000            ADD     #100000,R0      ;CC=0111   R0=0
6330  020310  101002                    BHI     ADD5
6331  020312  102001                    BVC     ADD5
6332  020314  100004                    BPL     ADD6
6333                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6334                                                    ;           CONDITIONAL BRANCH INST. AND   <====
6335                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
6336                                                    ;           WHICH FOLLOWS W/ 742           <====
6337  020316                    ADD5:
```

```
6338  020316  012742  000460          MOV     #460,-(R2)      ;MOVE TO MAILBOX #  *******  460  *******
6339  020322  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6340  020324  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6341  020326  062700  177777  ADD6:   ADD     #177777,R0      ;CC=1000   R0=177777
6342  020332  101402                  BLOS    ADD7
6343  020334  102401                  BVS     ADD7
6344  020336  100404                  BMI     ADD8
6345                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6346                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6347                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6348                                                          ;           WHICH FOLLOWS W/ 731           <====
6349  020340          ADD7:
6350  020340  012742  000461          MOV     #461,-(R2)      ;MOVE TO MAILBOX #  *******  461  *******
6351  020344  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6352  020346  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6353  020350  000277  ADD8:   SCC                             ;CC=1010
6354  020352  000245                  +CLC!CLZ
6355  020354  062700  000001          ADD     #1,R0           ;CC=0101   R=0
6356  020360  102403                  BVS     ADD9
6357  020362  103002                  BCC     ADD9
6358  020364  100401                  BMI     ADD9
6359  020366  001404                  BEQ     TS216
6360                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6361                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6362                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6363                                                          ;           WHICH FOLLOWS W/ 715           <====
6364  020370          ADD9:
6365  020370  012742  000462          MOV     #462,-(R2)      ;MOVE TO MAILBOX #  *******  462  *******
6366  020374  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6367  020376  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6368                                                          ;  OR SEQUENCE ERROR
6369
6370                    ;*****************************************************************************************
6371                    ;TEST 216       TEST ADC INSTRUCTION
6372                    ;*****************************************************************************************
6373  020400  005212  TS216:  INC     (R2)            ;UPDATE TEST NUMBER
6374  020402  022712  000216          CMP     #216,(R2)       ;SEQUENCE ERROR?
6375  020406  001037                  BNE     TS217-10        ;BR TO ERROR HALT ON SEQ ERROR
6376  020410  012700  077777          MOV     #077777,R0
6377  020414  000277                  SCC                     ;CC=0101
6378  020416  000252                  +CLN!CLV
6379  020420  005500                  ADC     R0              ;CC=1010
6380  020422  101402                  BLOS    ADC1
6381  020424  102001                  BVC     ADC1
6382  020426  100404                  BMI     ADC2
6383                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6384                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6385                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6386                                                          ;           WHICH FOLLOWS W/ 767           <====
6387  020430          ADC1:
6388  020430  012742  000463          MOV     #463,-(R2)      ;MOVE TO MAILBOX #  *******  463  *******
6389  020434  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6390  020436  000000                  HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6391  020440  052700  077777  ADC2:   BIS     #77777,R0
6392  020444  000277                  SCC                     ;CC=1011
6393  020446  000244                  CLZ
```

```
                                                                L 10
 6394  020450  005500                ADC     R0              ;CC=0101   R0=0
 6395  020452  101002                BHI     ADC3
 6396  020454  102401                BVS     ADC3
 6397  020456  100004                BPL     ADC4
 6398                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6399                                                        ;           CONDITIONAL BRANCH INST. AND   <====
 6400                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
 6401                                                        ;           WHICH FOLLOWS W/ 753           <====
 6402  020460                    ADC3:
 6403  020460  012742  000464          MOV     #464,-(R2)      ;MOVE TO MAILBOX #  *******  464  *******
 6404  020464  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6405  020466  000000                HALT                    ;ADC DID NOT SET CC'S CORRECTLY
 6406  020470  000277            ADC4:SCC
 6407  020472  000245                +CLZ!CLC                        ;CC=1010
 6408  020474  005500                ADC     R0              ;CC=0100
 6409  020476  102403                BVS     ADC5
 6410  020500  103402                BCS     ADC5
 6411  020502  100401                BMI     ADC5
 6412  020504  001404                BEQ     TS217
 6413                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6414                                                        ;           CONDITIONAL BRANCH INST. AND   <====
 6415                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
 6416                                                        ;           WHICH FOLLOWS W/ 740           <====
 6417  020506                    ADC5:
 6418  020506  012742  000465          MOV     #465,-(R2)      ;MOVE TO MAILBOX #  *******  465  *******
 6419  020512  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6420  020514  000000                HALT                    ;ADC DID NOT SET CC'S CORRECTLY
 6421                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
                                ;CMP, AND COM INSTRUCTIONS.  EACH OF THESE INSTRUCTIONS GENERATE
                                ;THE C AND V BITS IDENTICALLY.  THE CONDITION CODES ARE PRESET,
                                ;THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
                                ;OF CONDITIONAL BRANCH INSTRUCTIONS.  THIS PROCEDURE IS REPEATED
                                ;SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
                                ;COMBINATIONS OF THE C AND V BITS.
                                ;
                                ;*****************************************************************************
                                ;TEST 217        TEST NEG INSTRUCTION
                                ;*****************************************************************************
020516  005212                  TS217:  INC     (R2)            ;UPDATE TEST NUMBER
020520  022712  000217                  CMP     #217,(R2)       ;SEQUENCE ERROR?
020524  001042                          BNE     TS220-10        ;BR TO ERROR HALT ON SEQ ERROR
020526  012700  000001                  MOV     #1,R0
020532  000277                          SCC                     ;CC=0110
020534  000251                          +CLN!CLC
020536  005400                          NEG     R0              ;CC=1001   R0=177777
020540  103003                          BCC     NEG1
020542  102402                          BVS     NEG1
020544  001401                          BEQ     NEG1
020546  100404                          BMI     NEG2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
020550                          NEG1:
020550  012742  000466                          MOV     #466,-(R2)      ;MOVE TO MAILBOX #  *******  466  *******
020554  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020556  000000                          HALT                    ;NEG DID NOT SET CC'S CORRECTLY
020560  042700  077777          NEG2:   BIC     #77777,R0
020564  000257                          CCC                     ;CC=0100
020566  000264                          SEZ
020570  005400                          NEG     R0              ;CC=1011   R0=100000
020572  102003                          BVC     NEG3
020574  103002                          BCC     NEG3
020576  001401                          BEQ     NEG3
020600  100404                          BMI     NEG4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ ?51           <====
020602                          NEG3:
020602  012742  000467                          MOV     #467,-(R2)      ;MOVE TO MAILBOX #  *******  467  *******
020606  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020610  000000                          HALT                    ;NEG DID NOT SET CC'S CORRECTLY
020612  005000          NEG4:   CLR     R0
020614  000277                          SCC                     ;CC=1011
020616  000244                          CLZ
020620  005400                          NEG     R0              ;CC=0100   R0=0
020622  102403                          BVS     NEG5
020624  103402                          BCS     NEG5
020626  001001                          BNE     NEG5
```

```
6478  020630  100004                  BPL     TS220
6479                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6480                                  ;           CONDITIONAL BRANCH INST. AND   <====
6481                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6482                                  ;           WHICH FOLLOWS W/ 735           <====
6483  020632                  NEG5:
6484  020632  012742  000470          MOV     #470,-(R2)      ;MOVE TO MAILBOX #  ******* 470 *******
6485  020636  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6486  020640  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
6487                                                          ;  OR SEQUENCE ERROR
6488
6489                          ;*****************************************************************************
6490                          ;TEST 220       TEST CMP INSTRUCTION
6491                          ;*****************************************************************************
6492  020642  005212          TS220:  INC     (R2)            ;UPDATE TEST NUMBER
6493  020644  022712  000220          CMP     #220,(R2)       ;SEQUENCE ERROR?
6494  020650  001060                  BNE     TS221-10        ;BR TO ERROR HALT ON SEQ ERROR
6495  020652  012700  000005          MOV     #5,R0
6496  020656  000257                  CCC                     ;CC=1010
6497  020660  000271                  +SEN!SEC
6498  020662  022700  000005          CMP     #5,R0           ;CC=0101
6499  020666  101002                  BHI     CMP1
6500  020670  102401                  BVS     CMP1
6501  020672  100004                  BPL     CMP2
6502                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6503                                  ;           CONDITIONAL BRANCH INST. AND   <====
6504                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6505                                  ;           WHICH FOLLOWS W/ 766           <====
6506  020674                  CMP1:
6507  020674  012742  000471          MOV     #471,-(R2)      ;MOVE TO MAILBOX #  ******* 471 *******
6508  020700  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6509  020702  000000                  HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6510  020704  012700  100000  CMP2:   MOV     #100000,R0
6511  020710  000277                  SCC                     ;CC=1101
6512  020712  000242                  CLV
6513  020714  020027  077777          CMP     R0,#77777       ;CC=0010
6514  020720  101402                  BLOS    CMP3
6515  020722  102001                  BVC     CMP3
6516  020724  100004                  BPL     CMP4
6517                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6518                                  ;           CONDITIONAL BRANCH INST. AND   <====
6519                                  ;           REPLACE THE MOVE INSTRUCTION   <====
6520                                  ;           WHICH FOLLOWS W/ 751           <====
6521  020726                  CMP3:
6522  020726  012742  000472          MOV     #472,-(R2)      ;MOVE TO MAILBOX #  ******* 472 *******
6523  020732  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6524  020734  000000                  HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6525  020736  052700  040000  CMP4:   BIS     #40000,R0       ;R0=140000
6526  020742  000257                  CCC                     ;CC=0100
6527  020744  000264                  SEZ
6528  020746  022700  040000          CMP     #40000,R0       ;CC=1011
6529  020752  102003                  BVC     CMP5
6530  020754  103002                  BCC     CMP5
6531  020756  001401                  BEQ     CMP5
6532  020760  100404                  BMI     CMP6
6533                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
6534                                                    ;          CONDITIONAL BRANCH INST. AND   <====
6535                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
6536                                                    ;          WHICH FOLLOWS W/ 733           <====
6537  020762                                CMP5:
6538  020762  012742  000473                        MOV     #473,-(R2)      ;MOVE TO MAILBOX #  *******  473  *******
6539  020766  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6540  020770  000000                                HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6541  020772  042700  040000                CMP6:   BIC     #40000,R0
6542  020776  000277                                SCC                     ;CC=1111
6543  021000  022700  177777                        CMP     #-1,R0          ;CC=0000
6544  021004  101402                                BLOS    CMP7
6545  021006  102401                                BVS     CMP7
6546  021010  100004                                BPL     TS221
6547                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6548                                                    ;          CONDITIONAL BRANCH INST. AND   <====
6549                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
6550                                                    ;          WHICH FOLLOWS W/ 717           <====
6551  021012                                CMP7:
6552  021012  012742  000474                        MOV     #474,-(R2)      ;MOVE TO MAILBOX #  *******  474  *******
6553  021016  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6554  021020  000000                                HALT                    ;CMP DID NOT SET CC'S CORRECTLY
6555                                                                        ;  OR SEQUENCE ERROR
6556
6557                                    ;*************************************************************************************
6558                                    ;TEST 221       TEST COM INSTRUCTION
6559                                    ;*************************************************************************************
6560  021022  005212                TS221:  INC     (R2)            ;UPDATE TEST NUMBER
6561  021024  022712  000221                CMP     #221,(R2)       ;SEQUENCE ERROR?
6562  021030  001010                        BNE     TS222-10        ;BR TO ERROR HALT ON SEQ ERROR
6563  021032  012700  177777                MOV     #-1,R0
6564  021036  000257                        CCC                     ;CC=1010
6565  021040  000265                        +SEC!SEZ
6566  021042  005100                        COM     R0              ;CC=0101
6567  021044  101002                        BHI     COM1
6568  021046  102401                        BVS     COM1
6569  021050  100004                        BPL     TS222
6570                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6571                                                    ;          CONDITIONAL BRANCH INST. AND   <====
6572                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
6573                                                    ;          WHICH FOLLOWS W/ 767           <====
6574  021052                                COM1:
6575  021052  012742  000475                        MOV     #475,-(R2)      ;MOVE TO MAILBOX #  *******  475  *******
6576  021056  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6577  021060  000000                                HALT                    ;COM DID NOT SET CC'S CORRECTLY
6578                                                                        ;  OR SEQUENCE ERROR
6579
```

```
6580
6581                                    ;*****************************************************************************
6582                                    ;
6583                                    ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
6584                                    ;AND SBC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE
6585                                    ;C AND V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
6586                                    ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
6587                                    ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
6588                                    ;BRANCHES.  THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
6589                                    ;DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
6590                                    ;
6591                                    ;*****************************************************************************
6592                                    ;TEST 222       TEST SUB INSTRUCTION
6593                                    ;*****************************************************************************
6594  021062  005212                TS222:  INC     (R2)            ;UPDATE TEST NUMBER
6595  021064  022712  000222                CMP     #222,(R2)       ;SEQUENCE ERROR?
6596  021070  001055                        BNE     TS223-10        ;BR TO ERROR HALT ON SEQ ERROR
6597  021072  012700  125252                MOV     #125252,R0
6598  021076  000257                        CCC                     ;CC=1010
6599  021100  000271                        +SEN!SEC
6600  021102  162700  125252                SUB     #125252,R0      ;CC=0101   R0=0
6601  021106  101002                        BHI     SUB1
6602  021110  102401                        BVS     SUB1
6603  021112  100004                        BPL     SUB2
6604                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6605                                                                ;           CONDITIONAL BRANCH INST. AND   <====
6606                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6607                                                                ;           WHICH FOLLOWS W/ 766           <====
6608  021114                        SUB1:
6609  021114  012742  000476                MOV     #476,-(R2)      ;MOVE TO MAILBOX #  *******  476  *******
6610  021120  005242                        INC,    -(R2)           ;SET MSGTYP TO FATAL ERROR
6611  021122  000000                        HALT                    ;SUB DID NOT SET CC'S CORRECTLY
6612  021124  052700  100000        SUB2:   BIS     #100000,R0
6613  021130  000277                        SCC                     ;CC=1101
6614  021132  000242                        CLV
6615  021134  162700  077777                SUB     #77777,R0       ;CC=0010   R0=1
6616  021140  101402                        BLOS    SUB3
6617  021142  102001                        BVC     SUB3
6618  021144  100004                        BPL     SUB4
6619                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6620                                                                ;           CONDITIONAL BRANCH INST. AND   <====
6621                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
6622                                                                ;           WHICH FOLLOWS W/ 751           <====
6623  021146                        SUB3:
6624  021146  012742  000477                MOV     #477,-(R2)      ;MOVE TO MAILBOX #  *******  477  *******
6625  021152  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6626  021154  000000                        HALT                    ;
6627  021156  005100        SUB4:   COM     R0                      ;R0=177777
6628  021160  000277                        SCC                     ;CC=11111
6629
6630  021162  162700  100000                SUB     #100000,R0      ;CC=0000   R0=77777
6631  021166  101402                        BLOS    SUB5
6632  021170  102401                        BVS     SUB5
6633  021172  100004                        BPL     SUB6
6634                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6635                                                                ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 736           <====
021174                                          SUB5:
021174  012742  000500                                  MOV     #500,-(R2)      ;MOVE TO MAILBOX #  *******  500  *******
021200  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021202  000000                                          HALT                    ;SUB DID NOT SET CC'S CORRECTLY
021204  000257                                  SUB6:   CCC                     ;CC=0100
021206  000264                                          SEZ
021210  162700  140000                                  SUB     #140000,R0      ;CC=1011
021214  102003                                          BVC     SUB7
021216  103002                                          BCC     SUB7
021220  001401                                          BEQ     SUB7
021222  100404                                          BMI     TS223
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 722           <====
021224                                          SUB7:
021224  012742  000501                                  MOV     #501,-(R2)      ;MOVE TO MAILBOX #  *******  501  *******
021230  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021232  000000                                          HALT                    ;
                                                ;*******************************************************************************
                                                ;TEST 223       TEST SBC INSTRUCTION
                                                ;*******************************************************************************
021234  005212                                  TS223:  INC     (R2)            ;UPDATE TEST NUMBER
021236  022712  000223                                  CMP     #223,(R2)       ;SEQUENCE ERROR?
021242  001053                                          BNE     TS224-10        ;BR TO ERROR HALT ON SEQ ERROR
021244  012700  000001                                  MOV     #1,R0
021250  000277                                          SCC                     ;CC=1011
021252  000244                                          CLZ
021254  005600                                          SBC     R0              ;CC=0100   R=0
021256  103403                                          BCS     SBC1
021260  102402                                          BVS     SBC1
021262  100401                                          BMI     SBC1
021264  001404                                          BEQ     SBC2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 766           <====
021266                                          SBC1:
021266  012742  000502                                  MOV     #502,-(R2)      ;MOVE TO MAILBOX #  *******  502  *******
021272  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021274  000000                                          HALT                    ;SBC DID NOT SET CC'S CORRECTLY
021276  000277                                  SBC2:   SCC                     ;CC=1010
021300  000245                                          +CLZ!CLC
021302  005600                                          SBC     R0              ;CC=0100   R=0
021304  103403                                          BCS     SBC3
021306  102402                                          BVS     SBC3
021310  100401                                          BMI     SBC3
021312  001404                                          BEQ     SBC4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 753           <====
021314                                          SBC3:
```

```
6692  021314  012742  000503          MOV     #503,-(R2)      ;MOVE TO MAILBOX #  *******  503  *******
6693  021320  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6694  021322  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6695  021324  000277          SBC4:   SCC                     ;CC=0111
6696  021326  000250                  CLN
6697  021330  005600                  SBC     R0              ;CC=1001   R0=177777
6698  021332  103003                  BCC     SBC5
6699  021334  102402                  BVS     SBC5
6700  021336  001401                  BEQ     SBC5
6701  021340  100404                  BMI     SBC6
6702                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6703                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6704                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6705                                                          ;           WHICH FOLLOWS W/ 740           <====
6706  021342          SBC5:
6707  021342  012742  000504          MOV     #504,-(R2)      ;MOVE TO MAILBOX #  *******  504  *******
6708  021346  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6709  021350  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6710  021352  042700  077777  SBC6:   BIC     #77777,R0       ;R0=100000
6711  021356  000277                  SCC                     ;CC=1101
6712  021360  000242                  CLV
6713  021362  005600                  SBC     R0              ;CC=0010
6714  021364  101402                  BLOS    SBC7
6715  021366  102001                  BVC     SBC7
6716  021370  100004                  BPL     TS224
6717                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6718                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6719                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6720                                                          ;           WHICH FOLLOWS W/ 724           <====
6721  021372          SBC7:
6722  021372  012742  000505          MOV     #505,-(R2)      ;MOVE TO MAILBOX #  *******  505  *******
6723  021376  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6724  021400  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6725                                                          ;  OR SEQUENCE ERROR
6726
```

```
                                 ;*************************************************************************
                                 ;
                                 ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
                                 ;ROR, ASL AND ASR INSTRUCTIONS.  SPECIAL DATA PATTERNS ARE LOADED
                                 ;AND ROTATED SEVERAL TIMES FOR EACH TEST.  THE CONDITION CODES
                                 ;ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
                                 ;CHECKED AFTER EACH ROTATION.  THE FINAL CHECK IN EACH TEST IS
                                 ;TO VERIFY THE COMMULATIVE DATA RESULT.  THE DATA PATTERNS HAVE
                                 ;BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.
                                 ;
                                 ;*************************************************************************
                                 ;TEST 224         TEST ROL INSTRUCTION
                                 ;*************************************************************************
021402  005212                   TS224:  INC     (R2)            ;UPDATE TEST NUMBER
021404  022712  000224                   CMP     #224,(R2)       ;SEQUENCE ERROR?
021410  001053                           BNE     TS225-10        ;BR TO ERROR HALT ON SEQ ERROR
021412  012700  144000                   MOV     #144000,R0      ;R0=144000
021416  000257                           CCC                     ;CC=0110
021420  000266                           +SEZ!SEV
021422  006100                           ROL     R0              ;CC=1001   R0=110000
021424  103003                           BCC     ROL1
021426  102402                           BVS     ROL1
021430  001401                           BEQ     ROL1
021432  100404                           BMI     ROL2
                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                 ;           CONDITIONAL BRANCH INST. AND    <====
                                                                 ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                 ;           WHICH FOLLOWS W/ 766            <====
021434                           ROL1:
021434  012742  000506                   MOV     #506,-(R2)      ;MOVE TO MAILBOX #  *******  506  *******
021440  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021442  000000                           HALT                    ;
021444  000277                   ROL2:   SCC                     ;CC=1100
021446  000243                           +CLV!CLC
021450  006100                           ROL     R0              ;CC=0011   R0=020000
021452  103003                           BCC     ROL3
021454  102002                           BVC     ROL3
021456  001401                           BEQ     ROL3
021460  100004                           BPL     ROL4
                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                 ;           CONDITIONAL BRANCH INST. AND    <====
                                                                 ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                 ;           WHICH FOLLOWS W/ 753            <====
021462                           ROL3:
021462  012742  000507                   MOV     #507,-(R2)      ;MOVE TO MAILBOX #  *******  507  *******
021466  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021470  000000                           HALT                    ;ROL DID NOT SET CC'S CORRECTLY
021472  000277                   ROL4:   SCC                     ;CC=0111
021474  000250                           CLN
021476  006100                           ROL     R0              ;CC=0000   R0=040001
021500  101402                           BLOS    ROL5
021502  102401                           BVS     ROL5
021504  100004                           BPL     ROL6
                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                 ;           CONDITIONAL BRANCH INST. AND    <====
                                                                 ;           REPLACE THE MOVE INSTRUCTION    <====
```

```
6783
6784  021506
6785  021506  012742  000510                ROL5:
6786  021512  005242                              MOV     #510,-(R2)      ;MOVE TO MAILBOX #  *******  510  *******
6787  021514  000000                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6788  021516  000257                              HALT                    ;ROL DID NOT SET CC'S CORRECTLY
6789  021520  000265                ROL6:         CCC                     ;CC=0101
6790  021522  006100                              +SEZ!SEC
6791  021524  101405                              ROL     R0              ;CC=1010   R0=100003
6792  021526  102004                              BLOS    ROL7
6793  021530  100003                              BVC     ROL7
6794  021532  022700  100003                    BPL     ROL7
6795  021536  001404                              CMP     #100003,R0
6796                                              BEQ     TS225
6797                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6798                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
6799                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
6800  021540                                                              ;           WHICH FOLLOWS W/ 724           <====
6801  021540  012742  000511                ROL7:
6802  021544  005242                              MOV     #511,-(R2)      ;MOVE TO MAILBOX #  *******  511  *******
6803  021546  000000                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6804                                              HALT                    ;ROL MALFUNCTIONED
6805                                                                      ;  OR SEQUENCE ERROR
6806                                      ;*****************************************************************************
6807                                      ;TEST 225       TEST ROR INSTRUCTION
6808  021550  005212                      ;*****************************************************************************
6809  021552  022712  000225              TS225:  INC     (R2)            ;UPDATE TEST NUMBER
6810  021556  001051                              CMP     #225,(R2)       ;SEQUENCE ERROR?
6811  021560  012700  000023                      BNE     TS226-10        ;BR TO ERROR HALT ON SEQ ERROR
6812  021564  000277                              MOV     #23,R0          ;R0=23
6813  021566  000250                              SCC                     ;CC=0111
6814  021570  006000                              CLN
6815  021572  102403                              ROR     R0              ;CC=1001   R0=100011
6816  021574  103002                              BVS     ROR1
6817  021576  001401                              BCC     ROR1
6818  021600  100404                              BEQ     ROR1
6819                                              BMI     ROR2
6820                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6821                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
6822                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
6823  021602                                                              ;           WHICH FOLLOWS W/ 766           <====
6824  021602  012742  000512                ROR1:
6825  021606  005242                              MOV     #512,-(R2)      ;MOVE TO MAILBOX #  *******  512  *******
6826  021610  000000                              INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6827  021612  000257                              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6828  021614  000274                ROR2:         CCC                     ;CC=1100
6829  021616  006000                              +SEN!SEZ
6830  021620  102003                              ROR     R0              ;CC=0011   R0=040004
6831  021622  103002                              BVC     ROR3
6832  021624  001401                              BCC     ROR3
6833  021626  100004                              BEQ     ROR3
6834                                              BPL     ROR4
6835                                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6836                                                                      ;           CONDITIONAL BRANCH INST. AND   <====
6837                                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
6838  021630                                                              ;           WHICH FOLLOWS W/ 753           <====
                                        ROR3:
```

```
6839  021630  012742  000513          MOV     #513,-(R2)      ;MOVE TO MAILBOX #  *******  513  *******
6840  021634  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6841  021636  000000                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6842  021640  000277          ROR4:   SCC                     ;CC=1110
6843  021642  000241                  CLC
6844  021644  006000                  ROR     R0              ;CC=0000   R0=020002
6845  021646  101403                  BLOS    ROR5
6846  021650  102402                  BVS     ROR5
6847  021652  001401                  BEQ     ROR5
6848  021654  100004                  BPL     ROR6
6849                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6850                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6851                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6852                                                          ;           WHICH FOLLOWS W/ 740           <====
6853  021656          ROR5:
6854  021656  012742  000514          MOV     #514,-(R2)      ;MOVE TO MAILBOX #  *******  514  *******
6855  021662  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6856  021664  000000                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6857  021666  000257          ROR6:   CCC                     ;CC=0101
6858  021670  000265                  +SEC!SEZ
6859  021672  006000                  ROR     R0              ;CC=1010   R0=110001
6860  021674  101402                  BLOS    ROR7
6861  021676  102001                  BVC     ROR7
6862  021700  100404                  BMI     TS226
6863                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6864                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6865                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6866                                                          ;           WHICH FOLLOWS W/ 726           <====
6867  021702          ROR7:
6868  021702  012742  000515          MOV     #515,-(R2)      ;MOVE TO MAILBOX #  *******  515  *******
6869  021706  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6870  021710  000000                  HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
6871                                                          ;  OR SEQUENCE ERROR
6872                                ;*************************************************************************
6873                                ;TEST 226       TEST ASL INSTRUCTION
6874                                ;*************************************************************************
6875  021712  005212          TS226:  INC     (R2)            ;UPDATE TEST NUMBER
6876  021714  022712  000226          CMP     #226,(R2)       ;SEQUENCE ERROR?
6877  021720  001054                  BNE     TS227-10        ;BR TO ERROR HALT ON SEQ ERROR
6878  021722  012700  144000          MOV     #144000,R0      ;R0=14000
6879  021726  000257                  CCC                     ;CC=0110
6880  021730  000271                  +SEN!SEC
6881  021732  006300                  ASL     R0              ;CC=1001   R0=110000
6882  021734  103003                  BCC     ASL1
6883  021736  102402                  BVS     ASL1
6884  021740  001401                  BEQ     ASL1
6885  021742  100404                  BMI     ASL2
6886                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6887                                                          ;           CONDITIONAL BRANCH INST. AND   <====
6888                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
6889                                                          ;           WHICH FOLLOWS W/ 766           <====
6890  021744          ASL1:
6891  021744  012742  000516          MOV     #516,-(R2)      ;MOVE TO MAILBOX #  *******  516  *******
6892  021750  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6893  021752  000000                  HALT                    ;
6894  021754  000277          ASL2:   SCC                     ;CC=1100
```

```
6895  021756  000243                    +CLV!CLC
6896  021760  006300                    ASL     R0              ;CC=0011   R0=020000
6897  021762  103003                    BCC     ASL3
6898  021764  102002                    BVC     ASL3
6899  021766  001401                    BEQ     ASL3
6900  021770  100004                    BPL     ASL4
6901                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6902                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6903                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6904                                                            ;           WHICH FOLLOWS W/ 753           <====
6905  021772                    ASL3:
6906  021772  012742  000517            MOV     #517,-(R2)      ;MOVE TO MAILBOX #  *******  517  *******
6907  021776  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6908  022000  000000                    HALT                    ;ASL DID NOT SET CC'S CORRECTLY
6909  022002  000277            ASL4:   SCC                     ;CC=0111
6910  022004  000250                    CLN
6911  022006  006300                    ASL     R0              ;CC=0000   R0=040000
6912  022010  101402                    BLOS    ASL5
6913  022012  102401                    BVS     ASL5
6914  022014  100004                    BPL     ASL6
6915                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6916                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6917                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6918                                                            ;           WHICH FOLLOWS W/ 741           <====
6919  022016                    ASL5:
6920  022016  012742  000520            MOV     #520,-(R2)      ;MOVE TO MAILBOX #  *******  520  *******
6921  022022  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6922  022024  000000                    HALT                    ;ASL DID NOT SET CC'S CORRECTLY
6923  022026  000257            ASL6:   CCC                     ;CC=0101
6924  022030  000265                    +SEZ!SEC
6925  022032  006300                    ASL     R0              ;CC=1010   R0=100000
6926  022034  103406                    BCS     ASL7
6927  022036  001405                    BEQ     ASL7
6928  022040  102004                    BVC     ASL7
6929  022042  100003                    BPL     ASL7
6930  022044  022700  100000            CMP     #100000,R0
6931  022050  001404                    BEQ     TS227
6932                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6933                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6934                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6935                                                            ;           WHICH FOLLOWS W/ 723           <====
6936  022052                    ASL7:
6937  022052  012742  000521            MOV     #521,-(R2)      ;MOVE TO MAILBOX #  *******  521  *******
6938  022056  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6939  022060  000000                    HALT                    ;ASL MALFUNCTIONED
6940                                                            ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;TEST 227       TEST ASR INSTRUCTION
                                ;*****************************************************************************************
022062  005212        TS227:  INC     (R2)            ;UPDATE TEST NUMBER
022064  022712  000227        CMP     #227,(R2)       ;SEQUENCE ERROR?
022070  001060                BNE     TS230-10        ;BR TO ERROR HALT ON SEQ ERROR
022072  012700  100023        MOV     #100023,R0      ;R0=100023
022076  000277                SCC                     ;CC=0110
022100  000250                CLN
022102  006200                ASR     R0              ;CC=1001   RP=140011
022104  102403                BVS     ASR1
022106  103002                BCC     ASR1
022110  001401                BEQ     ASR1
022112  100404                BMI     ASR2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
022114          ASR1:
022114  012742  000522        MOV     #522,-(R2)      ;MOVE TO MAILBOX #  *******  522  *******
022120  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022122  000000                HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022124  042700  100000 ASR2:  BIC     #100000,R0      ;R0=40011
022130  000277                SCC                     ;CC=1100
022132  000243                +CLV!CLC
022134  006200                ASR     R0              ;CC=0011   R0=020004
022136  102003                BVC     ASR3
022140  103002                BCC     ASR3
022142  001401                BEQ     ASR3
022144  100004                BPL     ASR4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 751           <====
022146          ASR3:
022146  012742  000523        MOV     #523,-(R2)      ;MOVE TO MAILBOX #  *******  523  *******.
022152  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022154  000000                HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022156  000277          ASR4:  SCC                     ;CC=1111
022160  006200                ASR     R0              ;CC=0000   R0=010002
022162  101403                BLOS    ASR5
022164  102402                BVS     ASR5
022166  001401                BEQ     ASR5
022170  100004                BPL     ASR6
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 737           <====
022172          ASR5:
022172  012742  000524        MOV     #524,-(R2)      ;MOVE TO MAILBOX #  *******  524  *******
022176  005242                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022200  000000                HALT                    ;ASR DID NOT SET CC'S CORRECTLY
022202  052700  100000 ASR6:  BIS     #100000,R0      ;R0=110002
022206  000257                CCC                     ;CC=0101
022210  000265                +SEZ!SEC
```

```
6997  022212  006200                  ASR     R0              ;C=1010   R0=144001
6998  022214  101406                  BLOS    ASR7
6999  022216  102005                  BVC     ASR7
7000  022220  100004                  BPL     ASR7
7001  022222  001403                  BEQ     ASR7
7002  022224  022700  144001          CMP     #144001,R0      ;CHECK RESULT OF ASR'S
7003  022230  001404                  BEQ     TS230
7004                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7005                                                          ;           CONDITIONAL BRANCH INST. AND   <====
7006                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7007                                                          ;           WHICH FOLLOWS W/ 717           <====
7008  022232                    ASR7:
7009  022232  012742  000525          MOV     #525,-(R2)      ;MOVE TO MAILBOX #  *******  525  *******
7010  022236  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7011  022240  000000                  HALT                    ;ASR DID NOT FUNCTION CORRECTLY
7012                                                          ;  OR SEQUENCE ERROR
7013
7014
7015
7016                    ;*****************************************************************************
7017                    ;
7018                    ;       THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION CODES
7019                    ;ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
7020                    ;THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
7021                    ;CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
7022                    ;IS VERIFIED BY CONDITIONAL BRANCHES.
7023                    ;
7024                    ;*****************************************************************************
7025                    ;TEST 230       TEST THE SXT INSTRUCTION
7026                    ;*****************************************************************************
7027  022242  005212    TS230:  INC     (R2)            ;UPDATE TEST NUMBER
7028  022244  022712  000230          CMP     #230,(R2)       ;SEQUENCE ERROR?
7029  022250  001033                  BNE     TS231-10        ;BR TO ERROR HALT ON SEQ ERROR
7030  022252  005000                  CLR     R0
7031  022254  000277                  SCC                     ;SET CC=1011
7032  022256  000244                  CLZ
7033  022260  006700                  SXT     R0              ;TRY SXT
7034  022262  100006                  BPL     SXT0            ;TEST CC=1001
7035  022264  001405                  BEQ     SXT0
7036  022266  102404                  BVS     SXT0
7037  022270  103003                  BCC     SXT0
7038  022272  022700  177777          CMP     #-1,R0          ;CHECK DATA RESULT
7039  022276  001404                  BEQ     SXT1
7040                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7041                                                          ;           CONDITIONAL BRANCH INST. AND   <====
7042                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
7043                                                          ;           WHICH FOLLOWS W/ 764           <====
7044  022300                    SXT0:
7045  022300  012742  000526          MOV     #526,-(R2)      ;MOVE TO MAILBOX #  *******  526  *******
7046  022304  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7047  022306  000000                  HALT                    ;RESULTS OF SXT INCORRECT
7048  022310  005000    SXT1:   CLR     R0              ;R0=0
7049  022312  005010                  CLR     (R0)            ;LOC. 0=0
7050  022314  005110                  COM     (R0)            ;LOC. 0=177777
7051  022316  000257                  CCC                     ;SET CC=0110
7052  022320  000266                  +SEZ!SEV
```

```
 7053  022322  006710                    SXT     (R0)
 7054  022324  001005                    BNE     SXT2            ;TEST CC=0100
 7055  022326  103404                    BCS     SXT2
 7056  022330  102403                    BVS     SXT2
 7057  022332  100402                    BMI     SXT2
 7058  022334  005710                    TST     (R0)
 7059  022336  001404                    BEQ     TS231
 7060                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7061                                                ;           CONDITIONAL BRANCH INST. AND   <====
 7062                                                ;           REPLACE THE MOVE INSTRUCTION   <====
 7063                                                ;           WHICH FOLLOWS W/ 744           <====
 7064  022340                    SXT2:
 7065  022340  012742  000527           MOV     #527,-(R2)      ;MOVE TO MAILBOX #  *******  527  *******
 7066  022344  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7067  022346  000000                    HALT                    ;RESULTS OF SXT INCORRECT
 7068                                                ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************************
                        ;
                        ;       THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
                        ;OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.
                        ;AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
                        ;EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
                        ;REPRODUCE THE ORIGINAL VALUE IF R0=31525.
                        ;
                        ;*****************************************************************************************
                        ;TEST 231       TEST THE XOR INSTRUCTION
                        ;*****************************************************************************************
022350  005212          TS231:  INC     (R2)            ;UPDATE TEST NUMBER
022352  022712  000231          CMP     #231,(R2)       ;SEQUENCE ERROR?
022356  001035                  BNE     TS232-10        ;BR TO ERROR HALT ON SEQ ERROR
022360  012700  007463          MOV     #7463,R0        ;SET UP R0
022364  012701  031525          MOV     #31525,R1       ;SET UP R1
022370  000277                  SCC                     ;SET CC=1110
022372  000241                  CLC
022374  074100                  XOR     R1,R0           ;TRY XOR
022376  101406                  BLOS    XOR1            ;CC=0000?
022400  102405                  BVS     XOR1
022402  001404                  BEQ     XOR1
022404  100403                  BMI     XOR1
022406  022700  036146          CMP     #36146,R0       ;DATA RESULT CORRECT?
022412  001404                  BEQ     XOR2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
022414          XOR1:
022414  012742  000530          MOV     #530,-(R2)      ;MOVE TO MAILBOX #  *******  530  *******
022420  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022422  000000                  HALT                    ;
022424  010104          XOR2:   MOV     R1,R4
022426  000261                  SEC                     ;CC=1110
022430  000241                  CLC
022432  074400                  XOR     R4,R0           ;TRY XOR MODE 0,0
022434  101406                  BLOS    XOR3            ;CC=0000?
022436  102405                  BVS     XOR3
022440  001404                  BEQ     XOR3
022442  100403                  BMI     XOR3
022444  022700  007463          CMP     #7463,R0
022450  001404                  BEQ     TS232
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 742           <====
022452          XOR3:
022452  012742  000531          MOV     #531,-(R2)      ;MOVE TO MAILBOX #  *******  531  *******
022456  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022460  000000                  HALT                    ;RESULT OF XOR INCORRECT
                                                        ;  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
                                ;*****************************************************************************
022462  005212                  TS232:  INC     (R2)            ;UPDATE TEST NUMBER
022464  022712  000232                  CMP     #232,(R2)       ;SEQUENCE ERROR?
022470  001023                          BNE     TS233-10        ;BR TO ERROR HALT ON SEQ ERROR
022472  012700  000525                  MOV     #525,R0
022476  010004                          MOV     R0,R4
022500  000277                          SCC                     ;SET CC=1111
022502  101002                  SOB1:   BHI     SOB2            ;CC=1111?
022504  100001                          BPL     SOB2
022506  102404                          BVS     SOB3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
022510                          SOB2:
022510  012742  000532                  MOV     #532,-(R2)      ;MOVE TO MAILBOX #  *******  532  *******
022514  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022516  000000                          HALT                    ;
022520  005304                  SOB3:   DEC     R4              ;COUNT ITERATIONS
022522  000277                          SCC                     ;CC=1111
022524  077012                          SOB     R0,SOB1         ;DO SOB W/ R0
022526  101004                          BHI     SOB4            ;CHECK CC=1111
022530  100003                          BPL     SOB4
022532  102002                          BVC     SOB4
022534  005704                          TST     R4              ;ITERATION COUNT OK?
022536  001404                          BEQ     TS233
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 754           <====
022540                          SOB4:
022540  012742  000533                  MOV     #533,-(R2)      ;MOVE TO MAILBOX #  *******  533  *******
022544  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022546  000000                          HALT                    ;INCORRECT # OF BRANCHES OR CC'S CHANGED
                                                                ;  OR SEQUENCE ERROR
```

```
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175  022550  005212
7176  022552  022712  000233
7177  022556  001061
7178  022560  012706  001000
7179  022564  012746  125252
7180  022570  162706  000074
7181  022574  012705  022620
7182  022600  012746  006436
7183  022604  000277
7184  022606  000116
7185  022610  012742  000534
7186  022614  005242
7187  022616  000000
7188  022620  101010
7189  022622  100007
7190  022624  102006
7191  022626  020527  125252
7192  022632  001003
7193  022634  022706  001000
7194  022640  001404
7195
7196
7197
7198
7199  022642
7200  022642  012742  000535
7201  022646  005242
7202  022650  000000
7203  022652  012746  052525
7204  022656  012746  006400
7205  022662  010605
7206  022664  004737  022674
7207  022670  000137  022706
7208  022674  000205
7209  022676  012742  000536
7210  022702  005242
7211  022704  000000
7212  022706  022706  001000
7213  022712  001003
7214  022714  022705  052525
7215  022720  001404
7216
7217
7218
7219
7220  022722

;*****************************************************************************
;
;       THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
;OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
;THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
;OF THE TWO ROUTINES IN THE TEST.
;
;*****************************************************************************
;TEST 233        TEST MARK INSTRUCTION
;*****************************************************************************
TS233:  INC     (R2)            ;UPDATE TEST NUMBER
        CMP     #233,(R2)       ;SEQUENCE ERROR?
        BNE     TS234-10        ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #STBOT,SP
        MOV     #125252,-(SP)   ;PUT R5 VALUE ON STACK
        SUB     #74,SP          ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
        MOV     #MRK1,R5        ;SET NEW PC IN R5
        MOV     #6436,-(SP)     ;PUT MARK 36 INST. ON STACK
        SCC                     ;SET CC=1111
        JMP     (SP)            ;XFER CONTL TO MARK 36 INST. ON STACK
        MOV     #534,-(R2)      ;MOVE TO MAILBOX #  *******  534  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;MARK INST. SHOULD HAVE JUMPED TO MRK1
MRK1:   BHI     MRK2            ;TEST CC UNAFFECTED
        BPL     MRK2            ;IE. CC=1111
        BVC     MRK2
        CMP     R5,#125252      ;CHECK R5 RESTORED FROM STACK
        BNE     MRK2
        CMP     #STBOT,R6       ;CHECK STACK POINTER READJUSTED CORRECTLY.
        BEQ     MRK3
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 746           <====
MRK2:
        MOV     #535,-(R2)      ;MOVE TO MAILBOX #  *******  535  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;RESULTS OF MARK INCORRECT
MRK3:   MOV     #52525,-(SP)
        MOV     #6400,-(SP)     ;PUT MARK 0 INST. ON STACK
        MOV     SP,R5           ;SET ADDR. OF MARK INST. IN R5
        JSR     PC,@#MRK4       ;DO JSR
        JMP     @#MRK5          ;
MRK4:   RTS     R5              ;DO RTS WITH R5 TO MARK INST ON STACK
        MOV     #536,-(R2)      ;MOVE TO MAILBOX #  *******  536  *******
        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
        HALT                    ;RTS,MARK SEQUENCE FAILED
MRK5:   CMP     #STBOT,R6       ;STACK ADJUSTED CORRECTLY
        BNE     MRK6            ;IF NOT:  BR
        CMP     #52525,R5       ;CHECK IF R5 RESTORED FROM STACK
        BEQ     TS234
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                ;           CONDITIONAL BRANCH INST. AND   <====
                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                ;           WHICH FOLLOWS W/ 716           <====
MRK6:
```

```
7221  022722  012742  000537          MOV     #537,-(R2)      ;MOVE TO MAILBOX #  *******  537  *******
7222  022726  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7223  022730  000000                  HALT                    ;RESULTS OF MARK INCORRECT
7224                                                          ;  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
                                                ;*****************************************************************************
022732  005212                                  TS234:  INC     (R2)            ;UPDATE TEST NUMBER
022734  022712  000234                                  CMP     #234,(R2)       ;SEQUENCE ERROR?
022740  001024                                          BNE     TS235-10        ;BR TO ERROR HALT ON SEQ ERROR
022742  012700  000377                                  MOV     #377,R0
022746  000257                                          CCC
022750  106400                                          MTPS    R0
022752  022767  000357  155016                         CMP     #357,PS
022760  001404                                          BEQ     MTPS1
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 767           <====
022762  012742  000540                                          MOV     #540,-(R2)      ;MOVE TO MAILBOX #  *******  540  *******
022766  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022770  000000                                          HALT                    ;MTPS FAILED
022772  005000                                  MTPS1:  CLR     R0
022774  005010                                          CLR     (R0)
022776  000277                                          SCC                     ;CC=1111
023000  106410                                          MTPS    (R0)            ;TRY MTPS MODE 1
023002  100403                                          BMI     MTPS1A          ;CHECK PS
023004  102402                                          BVS     MTPS1A
023006  103401                                          BCS     MTPS1A
023010  001004                                          BNE     TS235
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 753           <====
023012                                          MTPS1A:
023012  012742  000541                                  MOV     #541,-(R2)      ;MOVE TO MAILBOX #  *******  541  *******
023016  005242                                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023020  000000                                          HALT                    ;MTPS FAILED
                                                                                ;  OR SEQUENCE ERROR
                                                ;*****************************************************************************
                                                ;TEST 235       TEST MTPS MODE 2
                                                ;*****************************************************************************
023022  005212                                  TS235:  INC     (R2)            ;UPDATE TEST NUMBER
023024  022712  000235                                  CMP     #235,(R2)       ;SEQUENCE ERROR?
023030  001021                                          BNE     TS236-10        ;BR TO ERROR HALT ON SEQ ERROR
023032  005000                                          CLR     R0              ;R0=0
023034  012710  177777                                  MOV     #-1,(R0)        ;LOC. 0=-1
023040  005037  177776                                  CLR     @#PS            ;PS=0
023044  106420                                          MTPS    (R0)+           ;TRY MTPS W/MODE 2
```

```
7281  023046  022737  000357  177776          CMP     #357,@#PS       ;CHECK DATA
7282  023054  001404                          BEQ     MTPS2
7283                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7284                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7285                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7286                                                                    ;           WHICH FOLLOWS W/ 765           <====
7287  023056  012742  000542                  MOV     #542,-(R2)      ;MOVE TO MAILBOX #  *******  542  *******
7288  023062  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7289  023064  000000                          HALT                    ;DEST. DATA INCORRECT
7290  023066  022700  000001          MTPS2:  CMP     #1,R0           ;CHECK DEST. REGISTER.
7291  023072  001404                          BEQ     TS236
7292                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7293                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7294                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7295                                                                    ;           WHICH FOLLOWS W/ 756           <====
7296  023074  012742  000543                  MOV     #543,-(R2)      ;MOVE TO MAILBOX #  *******  543  *******
7297  023100  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7298  023102  000000                          HALT                    ;DEST REGISTER NOT INCREMENTED BY 1
7299                                                                    ;  OR SEQUENCE ERROR
7300
7301                                  ;*************************************************************************************
7302                                  ;TEST 236       TEST MTPS MODE 3
7303                                  ;*************************************************************************************
7304  023104  005212                  TS236:  INC     (R2)            ;UPDATE TEST NUMBER
7305  023106  022712  000236                  CMP     #236,(R2)       ;SEQUENCE ERROR?
7306  023112  001024                          BNE     TS237-10        ;BR TO ERROR HALT ON SEQ ERROR
7307  023114  012700  000402                  MOV     #402,R0         ;R0=402
7308  023120  005010                          CLR     (R0)            ;LOC. 402=0
7309  023122  012737  052652  000000          MOV     #52652,@#0      ;LOC. 0=52652
7310  023130  005037  177776                  CLR     @#PS            ;PS=0
7311  023134  106430                          MTPS    @(R0)+          ;TRY MTPS W/MODE 3
7312  023136  022737  000252  177776          CMP     #252,@#PS       ;CHECK DEST. DATA
7313  023144  001404                          BEQ     MTPS3
7314                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7315                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7316                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7317                                                                    ;           WHICH FOLLOWS W/ 762           <====
7318  023146  012742  000544                  MOV     #544,-(R2)      ;MOVE TO MAILBOX #  *******  544  *******
7319  023152  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7320  023154  000000                          HALT                    ;DEST. DATA INCORRECT
7321  023156  022700  000404          MTPS3:  CMP     #404,R0         ;CHECK MODE 3 REGISTER.
7322  023162  001404                          BEQ     TS237
7323                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7324                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7325                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7326                                                                    ;           WHICH FOLLOWS W/ 753           <====
7327  023164  012742  000545                  MOV     #545,-(R2)      ;MOVE TO MAILBOX #  *******  545  *******
7328  023170  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7329  023172  000000                          HALT                    ;MODE 3 REGISTER INCORRECT
7330                                                                    ;  OR SEQUENCE ERROR
7331
7332                                  ;*************************************************************************************
7333                                  ;TEST 237       TEST MTPS MODE 4
7334                                  ;*************************************************************************************
7335  023174  005212                  TS237:  INC     (R2)            ;UPDATE TEST NUMBER
7336  023176  022712  000237                  CMP     #237,(R2)       ;SEQUENCE ERROR?
```

```
7337  023202  001022                        BNE     TS240-10        ;BR TO ERROR HALT ON SEQ ERROR
7338  023204  012700  000001                MOV     #1,R0           ;R0=1
7339  023210  012737  125125  000000        MOV     #125125,@#0     ;LOC. 0 = 125125
7340  023216  005037  177776                CLR     @#PS            ;PS=0
7341  023222  106440                        MTPS    -(R0)           ;TRY MTPS W/MODE 4
7342  023224  022737  000105  177776        CMP     #105,@#PS       ;CHECK DEST. DATA
7343  023232  001404                        BEQ     MTPS4
7344                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7345                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7346                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
7347                                                                ;           WHICH FOLLOWS W/ 763           <====
7348  023234  012742  000546                MOV     #546,-(R2)      ;MOVE TO MAILBOX #  *******  546  *******
7349  023240  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7350  023242  000000                        HALT                    ;DEST. DATA INCORRECT
7351  023244  005700                MTPS4:  TST     R0              ;CHECK MODE 4 REGISTER
7352  023246  001404                        BEQ     TS240
7353                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7354                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7355                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
7356                                                                ;           WHICH FOLLOWS W/ 755           <====
7357  023250  012742  000547                MOV     #547,-(R2)      ;MOVE TO MAILBOX #  *******  547  *******
7358  023254  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7359  023256  000000                        HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
7360                                                                ;  OR SEQUENCE ERROR
7361
7362                          ;*****************************************************************************
7363                          ;TEST 240       TEST MTPS MODE 5
7364                          ;*****************************************************************************
7365  023260  005212          TS240:  INC     (R2)            ;UPDATE TEST NUMBER
7366  023262  022712  000240                CMP     #240,(R2)       ;SEQUENCE ERROR?
7367  023266  001021                        BNE     TS241-10        ;BR TO ERROR HALT ON SEQ ERROR
7368  023270  012700  000404                MOV     #404,R0         ;R0=404
7369  023274  012737  177400  000000        MOV     #177400,@#0     ;LOC. 0=177400
7370  023302  000277                        SCC                     ;SET ALL COND. CODES
7371  023304  106450                        MTPS    @-(R0)          ;TRY MTPS W/MODE 5
7372  023306  005737  177776                TST     @#PS            ;CHECK DEST. DATA.
7373  023312  001404                        BEQ     MTPS5
7374                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7375                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7376                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
7377                                                                ;           WHICH FOLLOWS W/ 765           <====
7378  023314  012742  000550                MOV     #550,-(R2)      ;MOVE TO MAILBOX #  *******  550  *******
7379  023320  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7380  023322  000000                        HALT                    ;DESTINATION DATA INCORRECT
7381  023324  022700  000402        MTPS5:  CMP     #402,R0         ;CHECK MODE 5 REGISTER
7382  023330  001404                        BEQ     TS241
7383                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7384                                                                ;           CONDITIONAL BRANCH INST. AND   <====
7385                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
7386                                                                ;           WHICH FOLLOWS W/ 756           <====
7387  023332  012742  000551                MOV     #551,-(R2)      ;MOVE TO MAILBOX #  *******  551  *******
7388  023336  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7389  023340  000000                        HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
7390                                                                ;  OR SEQUENCE ERROR
7391
7392                          ;*****************************************************************************
```

```
                                        ;TEST 241       TEST MTPS MODE 6
                                        ;*************************************************************************************
023342  005212                  TS241:  INC     (R2)            ;UPDATE TEST NUMBER
023344  022712  000241                  CMP     #241,(R2)       ;SEQUENCE ERROR?
023350  001024                          BNE     TS242-10        ;BR TO ERROR HALT ON SEQ ERROR
023352  012737  052652  000000          MOV     #52652,@#0      ;LOC. 0=52652
023360  012700  000406                  MOV     #406,R0         ;R0=406
023364  005037  177776                  CLR     @#PS            ;PS=0
023370  106460  177372                  MTPS    -406(R0)        ;TRY MTPS W/MODE 6
023374  022737  000252  177776          CMP     #252,@#PS       ;CHECK DEST. DATA
023402  001404                          BEQ     MTPS6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
023404  012742  000552                  MOV     #552,-(R2)      ;MOVE TO MAILBOX #  *******  552  *******
023410  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023412  000000                          HALT                    ;DEST. DATA INCORRECT
023414  022700  000406          MTPS6:  CMP     #406,R0         ;CHECK MODE 6 REGISTER
023420  001404                          BEQ     TS242
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 753           <====
023422  012742  000553                  MOV     #553,-(R2)      ;MOVE TO MAILBOX #  *******  553  *******
023426  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023430  000000                          HALT                    ;MODE 6 REGISTER MODIFIED
                                                                ;  OR SEQUENCE ERROR

                                        ;*************************************************************************************
                                        ;TEST 242       TEST MTPS MODE 7
                                        ;*************************************************************************************
023432  005212                  TS242:  INC     (R2)            ;UPDATE TEST NUMBER
023434  022712  000242                  CMP     #242,(R2)       ;SEQUENCE ERROR?
023440  001024                          BNE     TS243-10        ;BR TO ERROR HALT ON SEQ ERROR
023442  012737  052652  000000          MOV     #52652,@#0      ;LOC. 0=52652
023450  012700  000410                  MOV     #410,R0         ;R0=410
023454  005037  177776                  CLR     @#PS            ;PS=0
023460  106470  177776                  MTPS    @-2(R0)         ;TRY MTPS W/MODE 7
023464  022737  000105  177776          CMP     #105,@#PS       ;CHECK DEST. DATA
023472  001404                          BEQ     MTPS7
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
023474  012742  000554                  MOV     #554,-(R2)      ;MOVE TO MAILBOX #  *******  554  *******
023500  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023502  000000                          HALT                    ;DESTINATION DATA INCORRECT
023504  022700  000410          MTPS7:  CMP     #410,R0         ;CHECK MODE 7 REGISTER
023510  001404                          BEQ     TS243
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 753           <====
023512  012742  000555                  MOV     #555,-(R2)      ;MOVE TO MAILBOX #  *******  555  *******
023516  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
                                                                H 12
7449  023520  000000                    HALT                    ;MODE 7 REGISTER MODIFIED
7450                                                              ;  OR SEQUENCE ERROR
7451
```

```
                                ;*****************************************************************************
                                ;
                                ;       THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
                                ;MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES IS MOVED TO THE
                                ;PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
                                ;BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
                                ;USED TO CHECK PROPER ADDRESSING AND DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 243       TEST MFPS INSTRUCTION
                                ;*****************************************************************************
023522  005212                TS243:  INC     (R2)            ;UPDATE TEST NUMBER
023524  022712  000243              CMP     #243,(R2)       ;SEQUENCE ERROR?
023530  001025                      BNE     TS244-10        ;BR TO ERROR HALT ON SEQ ERROR
023532  012737  000377  177776      MOV     #377,@#PS
023540  106700                      MFPS    R0
023542  022700  177757              CMP     #177757,R0
023546  001404                      BEQ     MFPS1
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 770           <====
023550  012742  000556              MOV     #556,-(R2)      ;MOVE TO MAILBOX #  *******  556  *******
023554  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023556  000000                      HALT                    ;MFPS FAILED
                                
023560  005000                MFPS1:  CLR     R0
023562  012737  177777  000000      MOV     #-1,@#0
023570  005037  177776              CLR     @#PS
023574  106710                      MFPS    (R0)
023576  105737  000000              TSTB    @#0
023602  001404                      BEQ     TS244
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 752           <====
023604  012742  000557              MOV     #557,-(R2)      ;MOVE TO MAILBOX #  *******  557  *******
023610  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023612  000000                      HALT                    ;MFPS FAILED
                                                            ;  OR SEQUENCE ERROR
                                
                                ;*****************************************************************************
                                ;TEST 244       TEST MFPS MODE 2
                                ;*****************************************************************************
023614  005212                TS244:  INC     (R2)            ;UPDATE TEST NUMBER
023616  022712  000244              CMP     #244,(R2)       ;SEQUENCE ERROR?
023622  001031                      BNE     TS245-10        ;BR TO ERROR HALT ON SEQ ERROR
023624  005000                      CLR     R0              ;R0=0
023626  005010                      CLR     (R0)            ;LOC. 0=0
023630  012737  000377  177776      MOV     #377,@#PS       ;SET PS=357
023636  106720                      MFPS    (R0)+           ;TRY MFPS W/MODE 2
023640  103003                      BCC     MFPS2A          ;BR TO ERROR IF C BIT CLEAR
023642  102402                      BVS     MFPS2A          ;BR TO ERROR IF V BIT SET
023644  001401                      BEQ     MFPS2A          ;BR TO ERROR IF Z BIT SET
023646  100404                      BMI     MFPS2B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
7508                                                    ;          CONDITIONAL BRANCH INST. AND   <====
7509                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
7510                                                    ;          WHICH FOLLOWS W/ 765           <====
7511  023650                            MFPS2A:
7512  023650  012742  000560                  MOV     #560,-(R2)      ;MOVE TO MAILBOX #  *******  560  *******
7513  023654  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7514  023656  000000                    HALT                    ;COND. CODES INCORRECT
7515  023660  022737  000357  000000 MFPS2B: CMP     #357,@#0        ;CHECK DEST. DATA
7516  023666  001404                    BEQ     MFPS2C
7517                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7518                                                    ;          CONDITIONAL BRANCH INST. AND   <====
7519                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
7520                                                    ;          WHICH FOLLOWS W/ 755           <====
7521  023670  012742  000561                  MOV     #561,-(R2)      ;MOVE TO MAILBOX #  *******  561  *******
7522  023674  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7523  023676  000000                    HALT                    ;DEST. DATA INCORRECT
7524  023700  022700  000001 MFPS2C: CMP     #1,R0           ;CHECK MODE 2 REGISTER
7525  023704  001404                    BEQ     TS245
7526                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7527                                                    ;          CONDITIONAL BRANCH INST. AND   <====
7528                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
7529                                                    ;          WHICH FOLLOWS W/ 746           <====
7530  023706  012742  000562                  MOV     #562,-(R2)      ;MOVE TO MAILBOX #  *******  562  *******
7531  023712  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7532  023714  000000                    HALT                    ;MODE 2 REGISTER NOT INCREMENTED 1
7533                                                    ;  OR SEQUENCE ERROR
7534
7535                                    ;*****************************************************************************
7536                                    ;TEST 245       TEST MFPS MODE 3
7537                                    ;*****************************************************************************
7538  023716  005212                    TS245:  INC     (R2)            ;UPDATE TEST NUMBER
7539  023720  022712  000245                    CMP     #245,(R2)       ;SEQUENCE ERROR?
7540  023724  001033                    BNE     TS246-10        ;BR TO ERROR HALT ON SEQ ERROR
7541  023726  012700  000406                    MOV     #406,R0         ;R0=406
7542  023732  005037  000000                    CLR     @#0             ;LOC. 0=0
7543  023736  012737  000252  177776          MOV     #252,@#PS       ;PS=252
7544  023744  106730                    MFPS    @(R0)+          ;TRY MFPS WITH MODE 3
7545  023746  103403                    BCS     MFPS3A          ;BR TO ERROR IF C-BIT SET
7546  023750  102402                    BVS     MFPS3A          ;BR TO ERROR IF V-BIT SET
7547  023752  001401                    BEQ     MFPS3A          ;BR TO ERROR IF Z-BIT SET
7548  023754  100404                    BMI     MFPS3B
7549                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
755C                                                    ;          CONDITIONAL BRANCH INST. AND   <====
7551                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
7552                                                    ;          WHICH FOLLOWS W/ 763           <====
7553  023756                            MFPS3A:
7554  023756  012742  000563                  MOV     #563,-(R2)      ;MOVE TO MAILBOX #  *******  563  *******
7555  023762  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7556  023764  000000                    HALT                    ;CONDITION CODES INCORRECT
7557  023766  022737  125000  000000 MFPS3B: CMP     #125000,@#0     ;CHECK DEST. DATA
7558  023774  001404                    BEQ     MFPS3C
7559                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7560                                                    ;          CONDITIONAL BRANCH INST. AND   <====
7561                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
7562                                                    ;          WHICH FOLLOWS W/ 753           <====
7563  023776  012742  000564                  MOV     #564,-(R2)      ;MOVE TO MAILBOX #  *******  564  *******
```

```
7564  024002  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7565  024004  000000                          HALT                    ;DEST DATA INCORRECT
7566  024006  020027  000410        MFPS3C: CMP     R0,#410         ;CHECK MODE 3 REGISTER.
7567  024012  001404                          BEQ     TS246
7568                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7569                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7570                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7571                                                                  ;           WHICH FOLLOWS W/ 744           <====
7572  024014  012742  000565                  MOV     #565,-(R2)      ;MOVE TO MAILBOX #  *******  565  *******
7573  024020  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7574  024022  000000                          HALT                    ;MODE 3 REGISTER NOT INCREMENTED BY 2
7575                                                                  ;  OR SEQUENCE ERROR
7576
7577                                  ;*****************************************************************************
7578                                  ;TEST 246       TEST MFPS MODE 4
7579                                  ;*****************************************************************************
7580  024024  005212                  TS246:  INC     (R2)            ;UPDATE TEST NUMBER
7581  024026  022712  000246                  CMP     #246,(R2)       ;SEQUENCE ERROR?
7582  024032  001033                          BNE     TS247-10        ;BR TO ERROR HALT ON SEQ ERROR
7583  024034  012700  000002                  MOV     #2,R0           ;R0=2
7584  024040  005037  000000                  CLR     @#0             ;LOC. 0=0
7585  024044  012737  000125  177776          MOV     #125,@#PS       ;PS=125
7586  024052  106740                          MFPS    -(R0)           ;TRY MFPS W/MODE 4
7587  024054  103003                          BCC     MFPS4A          ;BR TO ERROR IF C-BIT CLEAR
7588  024056  102402                          BVS     MFPS4A          ;BR TO ERROR IF V-BIT SET
7589  024060  001401                          BEQ     MFPS4A          ;BR TO ERROR IF Z-BIT SET
7590  024062  100004                          BPL     MFPS4B
7591                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7592                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7593                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7594                                                                  ;           WHICH FOLLOWS W/ 763           <====
7595  024064                          MFPS4A:
7596  024064  012742  000566                  MOV     #566,-(R2)      ;MOVE TO MAILBOX #  *******  566  *******
7597  024070  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7598  024072  000000                          HALT                    ;COND. CODES INCORRECT
7599  024074  022737  042400  000000  MFPS4B: CMP     #42400,@#0      ;CHECK DEST. DATA
7600  024102  001404                          BEQ     MFPS4C
7601                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7602                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7603                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7604                                                                  ;           WHICH FOLLOWS W/ 753           <====
7605  024104  012742  000567                  MOV     #567,-(R2)      ;MOVE TO MAILBOX #  *******  567  *******
7606  024110  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7607  024112  000000                          HALT                    ;DEST. DATA INCORRECT
7608  024114  020027  000001          MFPS4C: CMP     R0,#1           ;CHECK MODE 4 REGISTER
7609  024120  001404                          BEQ     TS247
7610                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7611                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7612                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7613                                                                  ;           WHICH FOLLOWS W/ 744           <====
7614  024122  012742  000570                  MOV     #570,-(R2)      ;MOVE TO MAILBOX #  *******  570  *******
7615  024126  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7616  024130  000000                          HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
7617                                                                  ;  OR SEQUENCE ERROR
7618
7619                                  ;*****************************************************************************
```

```
7620                                    ;TEST 247       TEST MFPS MODE 5
7621                                    ;*************************************************************************************
7622  024132  005212                    TS247:  INC     (R2)            ;UPDATE TEST NUMBER
7623  024134  022712  000247                    CMP     #247,(R2)       ;SEQUENCE ERROR?
7624  024140  001033                            BNE     TS250-10        ;BR TO ERROR HALT ON SEQ ERROR
7625  024142  012700  000410                    MOV     #410,R0         ;R0=410
7626  024146  012737  177777  000000            MOV     #-1,@#0         ;LOC. 0=-1
7627  024154  005037  177776                    CLR     @#PS            ;PS=0
7628  024160  106750                            MFPS    @-(R0)          ;TRY MFPS W/MODE 5
7629  024162  103403                            BCS     MFPS5A          ;BR TO ERROR IF C-BIT SET
7630  024164  102402                            BVS     MFPS5A          ;BR TO ERROR IF V-BIT SET
7631  024166  100401                            BMI     MFPS5A          ;BR TO ERROR IF N-BIT SET
7632  024170  001404                            BEQ     MFPS5B
7633                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7634                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7635                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7636                                                                    ;           WHICH FOLLOWS W/ 763           <====
7637  024172                            MFPS5A:
7638  024172  012742  000571                    MOV     #571,-(R2)      ;MOVE TO MAILBOX #  *******  571  *******
7639  024176  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7640  024200  000000                            HALT                    ;COND. CODES INCORRECT
7641  024202  022737  000377  000000    MFPS5B: CMP     #377,@#0        ;CHECK DEST. DATA
7642  024210  001404                            BEQ     MFPS5C
7643                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7644                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7645                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7646                                                                    ;           WHICH FOLLOWS W/ 753           <====
7647  024212  012742  000572                    MOV     #572,-(R2)      ;MOVE TO MAILBOX #  *******  572  *******
7648  024216  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7649  024220  000000                            HALT                    ;DEST DATA INCORRECT
7650  024222  020027  000406            MFPS5C: CMP     R0,#406         ;CHECK MODE 5 REGISTER
7651  024226  001404                            BEQ     TS250
7652                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7653                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
7654                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
7655                                                                    ;           WHICH FOLLOWS W/ 744           <====
7656  024230  012742  000573                    MOV     #573,-(R2)      ;MOVE TO MAILBOX #  *******  573  *******
7657  024234  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7658  024236  000000                            HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
7659                                                                    ;  OR SEQUENCE ERROR
7660
7661                                    ;*************************************************************************************
7662                                    ;TEST 250       TEST MFPS MODE 6
7663                                    ;*************************************************************************************
7664  024240  005212                    TS250:  INC     (R2)            ;UPDATE TEST NUMBER
7665  024242  022712  000250                    CMP     #250,(R2)       ;SEQUENCE ERROR?
7666  024246  001034                            BNE     TS251-10        ;BR TO ERROR HALT ON SEQ ERROR
7667  024250  012700  000401                    MOV     #401,R0         ;R0=410
7668  024254  005037  000000                    CLR     @#0             ;LOC. 0=0
7669  024260  012737  000252  177776            MOV     #252,@#PS       ;PS=252
7670  024266  106760  177377                    MFPS    -401(R0)        ;TRY MFPS W/MODE 6
7671  024272  102403                            BVS     MFPS6A          ;BR TO ERROR IF V-BIT SET
7672  024274  103402                            BCS     MFPS6A          ;BR TO ERROR IF C-BIT SET
7673  024276  001401                            BEQ     MFPS6A          ;BR TO ERROR IF Z-BIT SET
7674  024300  100404                            BMI     MFPS6B
7675                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 762           <====
024302                                  MFPS6A:
024302  012742  000574                          MOV     #574,-(R2)      ;MOVE TO MAILBOX #  *******  574  *******
024306  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024310  000000                                  HALT                    ;COND. CODES INCORRECT
024312  022737  000252  000000  MFPS6B: CMP     #252,@#0                ;CHECK DEST. DATA
024320  001404                          BEQ     MFPS6C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 752           <====
024322  012742  000575                          MOV     #575,-(R2)      ;MOVE TO MAILBOX #  *******  575  *******
024326  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024330  000000                                  HALT                    ;DEST. DATA INCORRECT
024332  022700  000401          MFPS6C: CMP     #401,R0                 ;CHECK DEST. REGISTER
024336  001404                          BEQ     TS251
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 743           <====
024340  012742  000576                          MOV     #576,-(R2)      ;MOVE TO MAILBOX #  *******  576  *******
024344  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024346  000000                                  HALT                    ;DEST. DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR

;*******************************************************************************************
;TEST 251       TEST MFPS MODE 7
;*******************************************************************************************
024350  005212                  TS251:  INC     (R2)            ;UPDATE TEST NUMBER
024352  022712  000251                  CMP     #251,(R2)       ;SEQUENCE ERROR?
024356  001034                          BNE     TS252-10        ;BR TO ERROR HALT ON SEQ ERROR
024360  012700  000777                  MOV     #777,R0         ;R0=777
024364  005037  000000                  CLR     @#0             ;LOC. 0=0
024370  012737  000125  177776          MOV     #125,@#PS       ;PS=125
024376  106770  177407                  MFPS    @-371(R0)       ;TRY MFPS W/MODE 7
024402  102403                          BVS     MFPS7A          ;BR TO ERROR IF V-BIT SET
024404  103002                          BCC     MFPS7A          ;BR TO ERROR IF C-BIT SET
024406  001401                          BEQ     MFPS7A          ;BR TO ERROR IF Z-BIT SET
024410  100004                          BPL     MFPS7B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 762           <====
024412                                  MFPS7A:
024412  012742  000577                          MOV     #577,-(R2)      ;MOVE TO MAILBOX #  *******  577  *******
024416  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024420  000000                                  HALT                    ;CONDITION CODE INCORRECT
024422  022737  042400  000000  MFPS7B: CMP     #42400,@#0              ;CHECK DESTINATION DATA
024430  001404                          BEQ     MFPS7C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;          CONDITIONAL BRANCH INST. AND   <====
                                                        ;          REPLACE THE MOVE INSTRUCTION   <====
                                                        ;          WHICH FOLLOWS W/ 752           <====
024432  012742  000600                          MOV     #600,-(R2)      ;MOVE TO MAILBOX #  *******  600  *******
```

```
7732  024436  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7733  024440  000000                          HALT                    ;DEST. DATA INCORRECT
7734  024442  022700  000777        MFPS7C: CMP     #777,R0         ;CHECK MODE 7 REGISTER
7735  024446  001404                          BEQ     TS252
7736                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7737                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7738                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7739                                                                  ;           WHICH FOLLOWS W/ 743           <====
7740  024450  012742  000601                  MOV     #601,-(R2)      ;MOVE TO MAILBOX #  *******  601  *******
7741  024454  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7742  024456  000000                          HALT                    ;MODE 7 REGISTER MODIFIED
7743                                                                  ;  OR SEQUENCE ERROR
7744
7745                                  ;*****************************************************************************
7746                                  ;
7747                                  ;       THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
7748                                  ;THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
7749                                  ;CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
7750                                  ;CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 240 (DECIMAL)
7751                                  ;ITERATIONS OF PROGRAM.
7752                                  ;
7753                                  ;*****************************************************************************
7754                                  ;TEST 252       TEST THAT RESET DOES NOT CLEAR PSW
7755                                  ;*****************************************************************************
7756  024460  005212                  TS252:  INC     (R2)            ;UPDATE TEST NUMBER
7757  024462  022712  000252                  CMP     #252,(R2)       ;SEQUENCE ERROR?
7758  024466  001017                          BNE     TS253-10        ;BR TO ERROR HALT ON SEQ ERROR
7759  024470  122767  000001  153622          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
7760  024476  001003                          BNE     1$              ;IF NOT, DO THIS TEST
7761  024500  005767  153602                  TST     $PASS           ;IS THIS THE FIRST PASS?
7762  024504  001014                          BNE     REST            ;IF NOT FIRST PASS, SKIP TEST
7763  024506                          1$:
7764  024506  012737  000357  177776          MOV     #357,@#PS               ;MOV ONES TO PSW
7765  024514  000005                          RESET                   ;
7766  024516  022737  000357  177776          CMP     #357,@#PS               ;PSW CORRECT?
7767  024524  001404                          BEQ     TS253
7768                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7769                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7770                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7771                                                                  ;           WHICH FOLLOWS W/ 760           <====
7772  024526  012742  000602                  MOV     #602,-(R2)      ;MOVE TO MAILBOX #  *******  602  *******
7773  024532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7774  024534  000000                          HALT                    ;RESET ALTERED PSW
7775                                                                  ;  OR SEQUENCE ERROR
7776  024536                          REST:
7777
7778                                  ;*****************************************************************************
7779                                  ;
7780                                  ;       THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
7781                                  ;DATA PATH COMPONENTS WITH USER MODE SET.
7782                                  ;
7783                                  ;*****************************************************************************
7784                                  ;TEST 253       TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
7785                                  ;*****************************************************************************
7786  024536  005212                  TS253:  INC     (R2)            ;UPDATE TEST NUMBER
7787  024540  022712  000253                  CMP     #253,(R2)       ;SEQUENCE ERROR?
```

```
7788  024544  001017                          BNE     TS254-10        ;BR TO ERROR HALT ON SEQ ERROR
7789  024546  052767  140000  153222          BIS     #USRM,PS                ;SET USER MODE
7790  024554  012706  000001                  MOV     #1,R6                   ;SET BIT0
7791  024560  000241                          CLC                             ;CLEAR C-BIT
7792  024562  006106                  USP1:   ROL     R6                      ;ROTATE 1 POSITION
7793  024564  103376                          BCC     USP1                    ;BR IF NOT ALL DONE
7794  024566  001407                          BEQ     USP1A                   ;BR IF NO BITS PICKED
7795  024570  042767  140000  153200          BIC     #USRM,PS                ;CLEAR USER MODE
7796  024576  012742  000603                  MOV     #603,-(R2)      ;MOVE TO MAILBOX #  *******  603  *******
7797  024602  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7798  024604  000000                          HALT                    ;USER MODE R6 PICKED A BIT
7799  024606  042767  140000  153162  USP1A:  BIC     #USRM,PS                ;CLEAR USER MODE
7800
7801
7802                                  ;*******************************************************************************
7803                                  ;
7804                                  ;       THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
7805                                  ;AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
7806                                  ;OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
7807                                  ;OF EACH OTHER.
7808                                  ;
7809                                  ;*******************************************************************************
7810                                  ;TEST 254       TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
7811                                  ;*******************************************************************************
7812  024614  005212                  TS254:  INC     (R2)            ;UPDATE TEST NUMBER
7813  024616  022712  000254                  CMP     #254,(R2)       ;SEQUENCE ERROR?
7814  024622  001043                          BNE     USP4-10         ;BR TO ERROR HALT ON SEQ ERROR
7815  024624  052767  140000  153144          BIS     #USRM,PS                ;SET USER MODE
7816  024632  012706  177777                  MOV     #-1,R6                  ;SET USER R6 TO ALL ONES
7817  024636  022706  177777                  CMP     #-1,R6                  ;READ AND CHECK USER R6
7818  024642  001407                          BEQ     USP2                    ;BR IF NO ERROR
7819  024644  042767  140000  153124          BIC     #USRM,PS                ;CLEAR USER MODE
7820  024652  012742  000604                  MOV     #604,-(R2)      ;MOVE TO MAILBOX #  *******  604  *******
7821  024656  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7822  024660  000000                          HALT                    ;USER R6 WILL NOT HOLD ALL ONES
7823  024662  042767  140000  153106  USP2:   BIC     #USRM,PS                ;SET KERNEL MODE
7824  024670  022706  177777                  CMP     #-1,R6                  ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
7825  024674  001004                          BNE     USP3
7826                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7827                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
7828                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
7829                                                                  ;           WHICH FOLLOWS W/ 752           <====
7830  024676  012742  000605                  MOV     #605,-(R2)      ;MOVE TO MAILBOX #  *******  605  *******
7831  024702  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7832  024704  000000                          HALT                    ;DUAL ADDRESSING ERROR USER/KERNEL R6
7833  024706  005006                  USP3:   CLR     R6                      ;CLEAR KERNEL MODE SP
7834  024710  052767  140000  153060          BIS     #USRM,PS                ;SET USER MODE
7835  024716  022706  177777                  CMP     #-1,R6                  ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7836  024722  042767  140000  153046          BIC     #USRM,PS                ;CLEAR USER MODE
7837  024730  001404                          BEQ     USP4                    ;BR IF NO ERROR
7838  024732  012742  000606                  MOV     #606,-(R2)      ;MOVE TO MAILBOX #  *******  606  *******
7839  024736  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7840  024740  000000                          HALT                    ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7841  024742  012706  001000          USP4:   MOV     #STBOT,R6               ;RESTORE SP USER
7842  024746  042767  140000  153022          BIC     #USRM,PS                ;SET KERNEL MODE
7843  024754  012706  001000                  MOV     #STBOT,R6               ;RESTORE SP KERNEL
```

```
                                ;*****************************************************************************
                                ;
                                ;       THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
                                ;WITH R6 IN MODE 0.
                                ;
                                ;*****************************************************************************
                                ;TEST 255       TEST MFPI WITH R6 IN MODE 0
                                ;*****************************************************************************
024760  005212                  TS255:  INC     (R2)            ;UPDATE TEST NUMBER
024762  022712  000255                  CMP     #255,(R2)       ;SEQUENCE ERROR?
024766  001032                          BNE     TS256-10        ;BR TO ERROR HALT ON SEQ ERROR
024770  012706  001000                  MOV     #STBOT,R6               ;INITIALIZE KERNEL STACK POINTER
024774  012767  140000  152774          MOV     #USRM,PS                ;SET USER MODE.PREVIOUS KERNEL
025002  012706  027352                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
025006  006506                          MFPI    R6                      ;TRY MFPI WITH MODE 0
025010  022767  140000  152760          CMP     #140000,PS              ;CHECK PSW
025016  001407                          BEQ     MFPI0                   ;BR IF NO ERROR
025020  042767  140000  152750          BIC     #USRM,PS                ;CLEAR USER MODE
025026  012742  000607                  MOV     #607,-(R2)      ;MOVE TO MAILBOX #  *******  607  *******
025032  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025034  000000                          HALT                    ;INCORRECT PSW FROM MFPI
025036  042767  140000  152732  MFPI0:  BIC     #USRM,PS                ;CLEAR USER MODE
025044  022767  001000  002276          CMP     #STBOT,USTBOT-2         ;CHECK DATA ON STACK
025052  001404                          BEQ     MFPI0A                  ;BR IF NO ERROR
025054  012742  000610                  MOV     #610,-(R2)      ;MOVE TO MAILBOX #  *******  610  *******
025060  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025062  000000                          HALT                    ;INCORRECT DATA FROM MFPI
025064                          MFPI0A:
                                ;*****************************************************************************
                                ;TEST 256       TEST MTPI WITH R6 IN MODE 0
                                ;*****************************************************************************
025064  005212                  TS256:  INC     (R2)            ;UPDATE TEST NUMBER
025066  022712  000256                  CMP     #256,(R2)       ;SEQUENCE ERROR?
025072  001033                          BNE     TS257-10        ;BR TO ERROR HALT ON SEQ ERROR
025074  005067  152676                  CLR     PS                      ;SET KERNEL MODE
025100  005006                          CLR     R6                      ;INITIALIZE KERNEL R6
025102  012767  140000  152666          MOV     #USRM,PS                ;SET USER MODE/PREVIOUS KERNEL
025110  012706  027352                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
025114  012746  001000                  MOV     #STBOT,-(R6)            ;SET UP TARGET DATA
025120  006606                          MTPI    R6                      ;TRY MODE 0 MTPI
025122  022767  140000  152646          CMP     #USRM,PS                ;CHECK PSW
025130  001407                          BEQ     MTPI0                   ;BR IF NO ERROR
025132  042767  140000  152636          BIC     #USRM,PS                ;CLEAR USER MODE
025140  012742  000611                  MOV     #611,-(R2)      ;MOVE TO MAILBOX #  *******  611  *******
025144  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025146  000000                          HALT                    ;PS INCORRECT FOLLOWING MTPI
025150  005067  152622          MTPI0:  CLR     PS                      ;SET KERNEL MODE
025154  020627  001000                  CMP     R6,#STBOT               ;CHECK TARGET DATA
025160  001404                          BEQ     TS257
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 744           <====
025162  012742  000612                  MOV     #612,-(R2)      ;MOVE TO MAILBOX #  *******  612  *******
```

```
7900  025166  005242          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7901  025170  000000          HALT                    ;DATA INCORRECT FOLLOWING MTPI
7902                                                  ;  OR SEQUENCE ERROR
7903
```

```
7904
7905                                    ;*****************************************************************************
7906                                    ;
7907                                    ;       THIS TEST EXECUTES EVERY POSSIBLE BRANCH WITH EVERY POSSIBLE
7908                                    ;CONDITION CODE COMBINATION.
7909                                    ;       THE ROUTINE USES TWO TABLES.  THE BRANCH TABLE HOLDS ALL THE
7910                                    ;POSSIBLE BRANCH INSTRUCTIONS, THE OTHER TABLE (YNTAB) HOLDS BIT MAPS FOR
7911                                    ;EACH BRANCH.  A ONE IN THE BIT MAP INDICATES THAT THE CORRESPONDING
7912                                    ;BRANCH INSTRUCTION SHOULD BRANCH FOR THE CONDITION CODE SETTING WHICH
7913                                    ;CORRESPONDS TO THE BIT POSITION WITHIN THE MAP.  FOR EXAMPLE IF THE LEFT
7914                                    ;MOST BIT IS A ONE THEN THE CORRESPONDING BRANCH INSTRUCTION SHOULD BRANCH
7915                                    ;WHEN THE CONDITION CODES ARE 0.
7916                                    ;       THE ROUTINE CONSISTS OF NESTED LOOPS; THE OUTER LOOP SETS UP
7917                                    ;ALL THE POSSIBLE BRANCH INSTRUCTIONS.  THE INNER LOOP SETS UP EVERY POSSIBLE
7918                                    ;CONDITION CODE FOR EACH BRANCH.
7919                                    ;       THE BIT MAP IS USED TO SET THE ADDRESS LOCATION IN TWO
7920                                    ;JUMP MODE 3 INSTRUCTIONS.  THE ADDRESSES ARE CHANGED TO ALLOW THE
7921                                    ;PROGRAM TO CONTINUE OR JUMP TO AN ERROR ROUTINE DEPENDING UPON
7922                                    ;WHETHER IT HANDLED THE BRANCH INSTRUCTION CORRECTLY.
7923                                    ;       AT ANY ERROR HALT, LOCATION, BRH, HOLDS THE BRANCH INSTRUCTION
7924                                    ;UNDER TEST AND LOCATION, CC, HOLDS THE VALUE OF THE CONDITION CODES
7925                                    ;AT THE TIME THE BRANCH WAS EXECUTED.
7926                                    ;
7927                                    ;*****************************************************************************
7928                                    ;TEST 257       TEST THE BRANCH ROM
7929                                    ;*****************************************************************************
7930  025172  005212                    TS257:  INC     (R2)            ;UPDATE TEST NUMBER
7931  025174  022712  000257                    CMP     #257,(R2)       ;SEQUENCE ERROR?
7932  025200  001062                            BNE     ER              ;BR TO ERROR HALT ON SEQ ERROR
7933  025202  012700  027242            SETUP:  MOV     #BRTAB,R0       ;INITIALIZE BRANCH TABLE POINTER
7934  025206  012704  027300                    MOV     #YNTAB,R4       ;INITIALIZE YES/NO BRANCH MAP POINTER
7935  025212  012767  000017  000142            MOV     #15.,BRCT       ;INITIALIZE BRANCH TABLE COUNT
7936  025220  012067  000110            SETBR:  MOV     (R0)+,BRH       ;GET NEXT BRANCH INST.
7937  025224  012401                            MOV     (R4)+,R1        ;GET NEXT BRANCH MAP
7938  025226  012767  177777  000074            MOV     #-1,CC1         ;INITIALIZE CONDITION CODE VALUE
7939  025234  012703  000020                    MOV     #16.,R3         ;INITIALIZE CONDITION CODE COUNT
7940  025240  005267  000064            SETCC:  INC     CC1             ;SET FOR NEXT CC VALUE
7941  025244  032701  100000                    BIT     #100000,R1      ;SEE IF SHOULD BR W/ THESE CC'S
7942  025250  013705  177776                    MOV     @#177776,R5     ;SIMULATE A JNE
7943  025254  042705  177773                    BIC     #177773,R5      ;       (JUMP NOT EQUAL)
7944  025260  000165  025264                    JMP     .+4(R5)         ;       TO SET2BR
7945  025264  000167  000020                    JMP     SET2BR
7946  025270  012767  025364  000042            MOV     #CONT,NBR       ;SET TO CONTINUE IF NO BRANCH
7947  025276  012767  025346  000040            MOV     #ER,YBR         ;SET TO REPORT ERROR IF BRANCH
7948  025304  000167  000014                    JMP     AROUN           ;GO AROUND OPPOSITE CONDITION
7949  025310  012767  025346  000022  SET2BR:   MOV     #ER,NBR         ;SET TO REPORT ERROR IF NO BRANCH
7950  025316  012767  025364  000020            MOV     #CONT,YBR       ;SET TO CONTINUE IF BRANCH
7951  025324  006101                    AROUN:  ROL     R1              ;UPDATE BIT MAP
7952
7953  025326  012737                            MOV     (PC)+,@(PC)+    ;SET CONDITION CODE
7954  025330  000000                    CC1:    0                       ;NEW CC VALUE GOES HERE
7955  025332  177776                            177776
7956  025334  000000                    BRH:    0                       ;BRANCH INST. GOES HERE
7957  025336  000137                            JMP     @(PC)+          ;THIS JUMP IF NO BRANCH
7958  025340  000000                    NBR:    0                       ;WHERE TO GO IF NO BRANCH OCCURS
7959  025342  000137                            JMP     @(PC)+          ;THIS JUMP IF BRANCH OCCURS
```

```
 7960  025344  000000
 7961  025346  012702  000304
 7962  025352  012742  000613
 7963  025356  005242
 7964  025360  000000
 7965  025362  000000
 7966  025364  005303
 7967  025366  013705  177776
 7968  025372  042705  177773
 7969  025376  000165  025402
 7970  025402  000167  177632
 7971  025406  005367  177750
 7972  025412  013705  177776
 7973  025416  042705  177773
 7974  025422  000165  025426
 7975  025426  000167  177566

YBR:    0
ER:     MOV     #$TESTN,R2      ;WHERE TO GO IF BRANCH OCCURS
        MOV     #613,-(R2)      ;RESTORE POINTER
        INC     -(R2)           ;MOVE TO MAILBOX #  *******  613  *******
        HALT                    ;SET MSGTYP TO FATAL ERROR
                                ;
BRCT:   0
CONT:   DEC     R3              ;CC'S DONE?
        MOV     @#177776,R5     ;SIMULATE A JNE
        BIC     #177773,R5      ;       (JUMP NOT EQUAL)
        JMP     .+4(R5)         ;       TO SETCC
        JMP     SETCC
        DEC     BRCT            ;BR'S DONE?
        MOV     @#177776,R5     ;SIMULATE A JNE
        BIC     #177773,R5      ;       (JUMP NOT EQUAL)
        JMP     .+4(R5)         ;       TO SETBR
        JMP     SETBR
```

```
                        ;*****************************************************************************
                        ;
                        ;THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL
                        ;REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET
                        ;IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH
                        ;REGISTER.
                        ;
                        ;*****************************************************************************
                        ;TEST 260       DUAL REGISTER ADDRESSING TEST
                        ;*****************************************************************************
025432  005212          TS260:  INC     (R2)            ;UPDATE TEST NUMBER
025434  022712  000260          CMP     #260,(R2)       ;SEQUENCE ERROR?
025440  001052                  BNE     DAERR           ;BR TO ERROR HALT ON SEQ ERROR
025442  005000          BITCLR: CLR     R0              ;INITIALIZE ALL REGISTERS
025444  005001                  CLR     R1
025446  005002                  CLR     R2
025450  005003                  CLR     R3
025452  005004                  CLR     R4
025454  005005                  CLR     R5
025456  005006                  CLR     R6
025460  052700  000001  BITSET: BIS     #1,R0           ;SET R0=1
025464  052701  000002          BIS     #2,R1           ;R1=2
025470  052702  000004          BIS     #4,R2           ;R2=4
025474  052703  000010          BIS     #10,R3          ;R3=10
025500  052704  000020          BIS     #20,R4          ;R4=20
025504  052705  000040          BIS     #40,R5          ;R5=40
025510  052706  000100          BIS     #100,R6         ;R6=100
025514  022706  000100  BITCHK: CMP     #100,R6         ;TEST THAT NO DUAL ADDRESSING OCCURRED
025520  001022                  BNE     DAERR           ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
025522  022705  000040          CMP     #40,R5
025526  001017                  BNE     DAERR
025530  022704  000020          CMP     #20,R4
025534  001014                  BNE     DAERR
025536  022703  000010          CMP     #10,R3
025542  001011                  BNE     DAERR
025544  022702  000004          CMP     #4,R2
025550  001006                  BNE     DAERR
025552  022701  000002          CMP     #2,R1
025556  001003                  BNE     DAERR
025560  022700  000001          CMP     #1,R0
025564  001404                  BEQ     BITCON
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 725           <====
025566                  DAERR:
025566  012742  000614          MOV     #614,-(R2)      ;MOVE TO MAILBOX #  *******  614  *******
025572  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025574  000000                  HALT                    ;DUAL ADDRESSING ERROR
025576  012702  000304  BITCON: MOV     #$TESTN,R2      ;RESTORE POINTER
```

```
                                ;*****************************************************************************
                                ;       THIS TEST VERIFIES THAT THE UPPER BYTE OF THE PSW IS NOT AFFECTED
                                ;WHEN THE PRIORITY LEVEL OR CC'S ARE CHANGED. ALL BITS ARE
                                ;INITIALLY SET IN THE PSW, AND THE LOW BYTE IS CLEARED. A BIT
                                ;INSTRUCTION VERIFIES THE DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 261       TEST BYTE INSTRUCTION ON PSW
                                ;*****************************************************************************
025602  005212                  TS261:  INC     (R2)            ;UPDATE TEST NUMBER
025604  022712  000261                  CMP     #261,(R2)       ;SEQUENCE ERROR?
025610  001012                          BNE     BTERR           ;BR TO ERROR HALT ON SEQ ERROR
025612  052737  170357  177776          BIS     #170357,@#PS    ;SET ALL POSSIBLE BITS IN PSW
025620  105037  177776                  CLRB    @#PS            ;CLR PR LEVEL AND CC'S
025624  013700  177776                  MOV     @#PS,R0         ;COPY CONTENTS OF PSW
025630  032700  170000                  BIT     #170000,R0      ;TEST THAT UPPER BYTE IS UNAFFECTED
025634  001006                          BNE     BTCON           ;CONTINUE IF OK
025636  005037  177776          BTERR:  CLR     @#PS            ;RETURN TO KERNEL MODE
025642  012742  000615                  MOV     #615,-(R2)      ;MOVE TO MAILBOX #  *******  615  *******
025646  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025650  000000                          HALT                    ;BYTE INSTRUCTION ALTERED PSW
025652  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
                                ;*****************************************************************************
025656  005212                  TS262:  INC     (R2)            ;UPDATE TEST NUMBER
025660  022712  000262                  CMP     #262,(R2)       ;SEQUENCE ERROR?
025664  001010                          BNE     TS263-10        ;BR TO ERROR HALT ON SEQ ERROR
025666  000277                          SCC
025670  000252                          +CLN!CLV                ;CC=0101
025672  000167  000000                  JMP     JMPT            ;JUMP TO TEST PSW
025676  100403                  JMPT:   BMI     JMPERR          ;BR TO ERROR HALT IF N-BIT IS SET
025700  001002                          BNE     JMPERR          ;BR TO ERROR HALT IF Z-BIT IS CLEAR
025702  102401                          BVS     JMPERR          ;BR TO ERROR HALT IF V-BIT IF SET
025704  103404                          BCS     TS263
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
025706                  JMPERR:
025706  012742  000616                  MOV     #616,-(R2)      ;MOVE TO MAILBOX #  *******  616  *******
025712  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025714  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
                                ;*******************************************************************************
025716  005212                TS263:  INC     (R2)            ;UPDATE TEST NUMBER
025720  022712  000263                CMP     #263,(R2)       ;SEQUENCE ERROR?
025724  001062                        BNE     CCERR           ;BR TO ERROR HALT ON SEQ ERROR
025726  012767  000240  000024        MOV     #240,CC3        ;INITIALIZE CLR CC INSTRUCTION CODES
025734  012767  000017  000032        MOV     #17,CC2         ;INITIALIZE OCTAL MAP
025742  012767  000261  000102        MOV     #261,SC3        ;INITIALIZE SET CC INSTRUCTION CODES
025750  012767  000001  000110        MOV     #1,SC4          ;INITIALIZE OCTAL MAP
025756  000277                CLRCD:  SCC                     ;SET ALL CONDITION CODES
025760  000000                CC3:    0                       ;CONDITION CODE INSTRUCTION
025762  013704  177776                MOV     @#PS,R4         ;COPY THE PSW
025766  042704  177760                BIC     #177760,R4      ;ISOLATE CONDITION CODES
025772  022704                        CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE CLEARED
025774  000000                CC2:    0                       ;OCTAL REPRESENTATION OF CC'S
025776  001404                        BEQ     CON1
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 752           <====
026000  012742  000617                MOV     #617,-(R2)      ;MOVE TO MAILBOX #  *******  617  *******
026004  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026006  000000                        HALT                    ;CLEAR CC INSTRUCTION FAILED
026010  005367  177760        CON1:   DEC     CC2             ;SET NEXT OCTAL MAP OF CC'S
026014  005267  177740                INC     CC3             ;GET NEXT CLEAR CC INSTRUCTION
026020  026727  177734  000257        CMP     CC3,#257        ;TEST FOR CCC INSTRUCTION
026026  003753                        BLE     CLRCD           ;GO TEST NEXT INSTRUCTION IF NOT FOUND
026030  026727  177724  000260        CMP     CC3,#260        ;CHECK FOR NOP=260
026036  001004                        BNE     SETCD           ;GO TEST SET CC INSTRUCTIONS
026040  012767  000017  177726        MOV     #17,CC2         ;SET OCTAL MAP TO TEST NOP
026046  000743                        BR      CLRCD           ;GO TEST NOP
026050  000257                SETCD:  CCC                     ;CLEAR ALL CONDITION CODES
026052  000000                SC3:    0                       ;CONDITION CODE INSTRUCTION
026054  013704  177776                MOV     @#PS,R4         ;COY PSW
026060  042704  177760                BIC     #177760,R4      ;CLEAR AWAY UNWANTED BITS
026064  022704                        CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE SET
026066  000000                SC4:    0                       ;OCTAL REPRESENTATION OF CC'S
026070  001404                        BEQ     CON2
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
 8135
 8136  026072                                  CCERR:                                  ;          WHICH FOLLOWS W/ 715          <====
 8137  026072  012742  000620                          MOV     #620,-(R2)      ;MOVE TO MAILBOX #  *******  620  *******
 8138  026076  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8139  026100  000000                                  HALT                    ;SET CC FAILED OR SEQUENCE ERROR
 8140  026102  005267  177760                  CON2:   INC     SC4             ;SET NEXT OCTAL MAP
 8141  026106  005267  177740                          INC     SC3             ;PREPARE NEXT SET CC INSTRUCTION
 8142  026112  026727  177734  000277                  CMP     SC3,#277        ;FINISHED?
 8143  026120  003753                                  BLE     SETCD           ;BR IF NO
 8144  026122  000167  000006                          JMP     MORO            ;JUMP TO NEXT TESTS
```

```
                                ;*****************************************************************************
                                .SBTTL  TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION
                                ;
                                ;IN AUTO INCREMENT (DECREMENT) MODES AND
                                ;AUTO INCREMENT (DECREMENT) DEFERRED MODES,
                                ;CONTENTS OF THE REGISTER IN USED ARE
                                ;INCREMENTED (DECREMENTED) BY 2
                                ;BEFORE USED AS THE SOURCE OPERAND.
                                ;
026126  000000  000000  000000  A:      .WORD 0,0,0
026134                          MOR0:
                                ;*****************************************************************************
                                ;TEST 264       TEST AUTO-INCREMENT MODE, USING R0
                                ;*****************************************************************************
026134  005212                  TS264:  INC     (R2)            ;UPDATE TEST NUMBER
026136  022712  000264                  CMP     #264,(R2)       ;SEQUENCE ERROR?
026142  001020                          BNE     TS265-10        ;BR TO ERROR HALT ON SEQ ERROR
026144  005037  026126                  CLR     @#A             ;CLEAR LOC A
026150  012700  026126                  MOV     #A,R0           ;R0 STORES ADDR OF A
026154  060020                          ADD     R0,(R0)+        ;CHECK THAT R0 IS INCR BY 2 BEFORE
                                                                ;BEING USED AS THE SOURCE OPERAND
026156  022700  026130                  CMP     #A+2,R0         ;R0 INCR BY 2?
026162  001404                          BEQ     MOR1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
026164  012742  000621                  MOV     #621,-(R2)      ;MOVE TO MAILBOX #  *******  621  *******
026170  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026172  000000                          HALT                    ;R0 WAS NOT INCREMENTED BY 2
                                                                ;
026174  022737  026130  026126  MOR1:   CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
                                                                ;BEING USED IN THE "ADD" INSTR
                                                                ;LOC A CONTAINS (A+2)?
026202  001404                          BEQ     TS265
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
026204  012742  000622                  MOV     #622,-(R2)      ;MOVE TO MAILBOX #  *******  622  *******
026210  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026212  000000                          HALT                    ;WRONG SUM IN LOC A
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 265       AUTO-DECREMENT MODE, USING R0
                                ;*****************************************************************************
026214  005212                  TS265:  INC     (R2)            ;UPDATE TEST NUMBER
026216  022712  000265                  CMP     #265,(R2)       ;SEQUENCE ERROR?
026222  001020                          BNE     TS266-10        ;BR TO ERROR HALT ON SEQ ERROR
026224  005037  026126                  CLR     @#A             ;CLEAR LOC A
026230  012700  026130                  MOV     #A+2,R0         ;R0 STORES ADDR OF A+2
026234  060040                          ADD     R0,-(R0)        ;CHECK THAT R0 IS DECR BY 2 BEFORE
                                                                ;BEING USED AS THE SOURCE OPERAND
026236  022700  026126                  CMP     #A,R0           ;R0 DECR BY 2?
026242  001404                          BEQ     MOR2
```

```
8201                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8202                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8203                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8204                                                    ;           WHICH FOLLOWS W/ 767           <====
8205  026244  012742  000623              MOV     #623,-(R2)      ;MOVE TO MAILBOX #  *******  623  *******
8206  026250  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8207  026252  000000                      HALT                    ;R0 WAS NOT DECREMENTED BY 2
8208                                                            ;
8209  026254  022737  026126  026126 MOR2: CMP     #A,@#A          ;CONTENT OF R0 WAS DECR BY 2 BEFORE
8210                                                            ;BEING USED IN THE "ADD" INSTR
8211                                                            ;LOC A CONTAINS (R0)
8212  026262  001404                      BEQ     TS266
8213                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8214                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8215                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8216                                                    ;           WHICH FOLLOWS W/ 757           <====
8217  026264  012742  000624              MOV     #624,-(R2)      ;MOVE TO MAILBOX #  *******  624  *******
8218  026270  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8219  026272  000000                      HALT                    ;WRONG SUM IN LOC A
8220                                                            ;  OR SEQUENCE ERROR
8221                                                            ;
8222                                    ;*******************************************************************************
8223                                    ;TEST 266       TEST AUTO-INCREMENT DEFERRED MODE, USING R0
8224                                    ;*******************************************************************************
8225  026274  005212                TS266: INC     (R2)            ;UPDATE TEST NUMBER
8226  026276  022712  000266              CMP     #266,(R2)       ;SEQUENCE ERROR?
8227  026302  001044                      BNE     TS267-10        ;BR TO ERROR HALT ON SEQ ERROR
8228  026304  005037  026126              CLR     @#A             ;CLEAR LOC A
8229  026310  005037  026132              CLR     @#A+4           ;CLEAR LOC A+4
8230  026314  012737  026126  026130      MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
8231  026322  012700  026130              MOV     #A+2,R0         ;R0 STORES ADDR A+2
8232  026326  060030                      ADD     R0,@(R0)+       ;CHECK THAT R0 IS INCR BY 2 BEFORE
8233                                                            ;BEING USED AS THE SOURCE OPERAND
8234  026330  022700  026132              CMP     #A+4,R0         ;R0 INCR BY 2?
8235  026334  001404                      BEQ     MOR3
8236                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8237                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8238                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8239                                                    ;           WHICH FOLLOWS W/ 762           <====
8240  026336  012742  000625              MOV     #625,-(R2)      ;MOVE TO MAILBOX #  *******  625  *******
8241  026342  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8242  026344  000000                      HALT                    ;R0 WAS NOT INCREMENTED BY 2
8243                                                            ;
8244  026346  022737  026126  026130 MOR3: CMP     #A,@#A+2        ;LOC A+2 STILL STORES ADDR A?
8245  026354  001404                      BEQ     MOR4
8246                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8247                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8248                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8249                                                    ;           WHICH FOLLOWS W/ 752           <====
8250  026356  012742  000626              MOV     #626,-(R2)      ;MOVE TO MAILBOX #  *******  626  *******
8251  026362  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8252  026364  000000                      HALT                    ;LOC A+2 STORES WRONG DATA
8253                                                            ;
8254  026366  022737  026132  026126 MOR4: CMP     #A+4,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8255                                                            ;BEING USED IN THE "ADD" INSTR
8256  026374  001404                      BEQ     MOR5
```

```
8257                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8258                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8259                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8260                                                    ;           WHICH FOLLOWS W/ 742           <====
8261  026376  012742  000627          MOV     #627,-(R2)      ;MOVE TO MAILBOX #  *******  627  *******
8262  026402  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8263  026404  000000                  HALT                    ;LOC A STORES WRONG DATA
8264                                                    ;
8265  026406  005737  026132  MOR5:   TST     @#A+4           ;LOC A+4 STILL STORES 0?
8266  026412  001404                  BEQ     TS267
8267                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8268                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8269                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8270                                                    ;           WHICH FOLLOWS W/ 733           <====
8271  026414  012742  000630          MOV     #630,-(R2)      ;MOVE TO MAILBOX #  *******  630  *******
8272  026420  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8273  026422  000000                  HALT                    ;LOC A+4 DID NOT STAY CLEAR
8274                                                    ;  OR SEQUENCE ERROR
8275
8276                                    ;*****************************************************************************
8277                                    ;TEST 267       TEST AUTO-DECREMENT DEFERRED, USING R0
8278                                    ;*****************************************************************************
8279  026424  005212                  TS267:  INC     (R2)            ;UPDATE TEST NUMBER
8280  026426  022712  000267          CMP     #267,(R2)       ;SEQUENCE ERROR?
8281  026432  001044                  BNE     TS270-10        ;BR TO ERROR HALT ON SEQ ERROR
8282  026434  005037  026126          CLR     @#A             ;CLEAR LOC A
8283  026440  005037  026132          CLR     @#A+4           ;CLEAR LOC A+4
8284  026444  012700  026132          MOV     #A+4,R0         ;R0 STORES ADDR A+4
8285  026450  012737  026126  026130  MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
8286  026456  060050                  ADD     R0,@-(R0)       ;CHECK THAT R0 IS DECR BY 2 BEFORE
8287                                                    ;BEING USED AS THE SOURCE OPERAND
8288  026460  022700  026130          CMP     #A+2,R0         ;R0 DECREMENTED BY 2?
8289  026464  001404                  BEQ     MOR6
8290                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8291                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8292                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8293                                                    ;           WHICH FOLLOWS W/ 762           <====
8294  026466  012742  000631          MOV     #631,-(R2)      ;MOVE TO MAILBOX #  *******  631  *******
8295  026472  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8296  026474  000000                  HALT                    ;R0 WAS NOT DECREMENTED BY 2
8297                                                    ;
8298  026476  022737  026130  026126  MOR6:   CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
8299                                                    ;BEING USED IN THE "ADD" INSTR
8300  026504  001404                  BEQ     MOR7
8301                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8302                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8303                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8304                                                    ;           WHICH FOLLOWS W/ 752           <====
8305  026506  012742  000632          MOV     #632,-(R2)      ;MOVE TO MAILBOX #  *******  632  *******
8306  026512  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8307  026514  000000                  HALT                    ;LOC A STORES WRONG DATA
8308
8309                                                    ;
8310  026516  022737  026126  026130  MOR7:   CMP     #A,@#A+2        ;LOC A+2 STILL STORES A?
8311  026524  001404                  BEQ     MOR8
8312                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
8313                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8314                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8315                                                    ;           WHICH FOLLOWS W/ 742           <====
8316  026526  012742  000633          MOV     #633,-(R2)  ;MOVE TO MAILBOX #  *******  633  *******
8317  026532  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
8318  026534  000000                  HALT                ;LOC A+2 STORES WRONG DATA
8319                                                    ;
8320  026536  005737  026132  MOR8:   TST     @#A+4       ;LOC A+4 STILL STORES 0?
8321  026542  001404                  BEQ     TS270
8322                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8323                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8324                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8325                                                    ;           WHICH FOLLOWS W/ 733           <====
8326  026544  012742  000634          MOV     #634,-(R2)  ;MOVE TO MAILBOX #  *******  634  *******
8327  026550  005242                  INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
8328  026552  000000                  HALT                ;LOC A+4 DID NOT STAY CLEAR
8329                                                    ;  OR SEQUENCE ERROR
8330                                                    ;
```

```
                        ;*****************************************************************************
                        .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
                        ;*****************************************************************************
026554  005212          TS270:  INC     (R2)            ;UPDATE TEST NUMBER
026556  022712  000270          CMP     #270,(R2)       ;SEQUENCE ERROR?
026562  001006                  BNE     TS271-10        ;BR TO ERROR HALT ON SEQ ERROR
026564  012700  177777          MOV     #-1,R0          ;SET ALL 1 IN R0
026570  010700          PCN01:  MOV     PC,R0           ;STORES PC IN R0
026572  022700  026572          CMP     #PCN01+2,R0     ;R0 STORES PC+2?
026576  001404                  BEQ     TS271
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
026600  012742  000635          MOV     #635,-(R2)      ;MOVE TO MAILBOX #  *******  635  *******
026604  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026606  000000                  HALT                    ;R0 STORED WRONG VALUE
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 271       TEST PC AS SOURCE IN MODE 6, USING R0
                        ;*****************************************************************************
026610  005212          TS271:  INC     (R2)            ;UPDATE TEST NUMBER
026612  022712  000271          CMP     #271,(R2)       ;SEQUENCE ERROR?
026616  001010                  BNE     TS272-10        ;BR TO ERROR HALT ON SEQ ERROR
026620  012700  026126          MOV     #A,R0           ;R0 STORES ADDR A
026624  010760  000004  PCN2:   MOV     PC,4(R0)                ;EFFECTIVE ADDR IS A+4
026630  022737  026630  026132  CMP     #PCN2+4,@#A+4   ;LOC A+4 STORES PC+4?
026636  001404                  BEQ     TS272
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
026640  012742  000636          MOV     #636,-(R2)      ;MOVE TO MAILBOX #  *******  636  *******
026644  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
026646  000000                  HALT                    ;LOC A+4 STORED WRONG VALUE
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 272       TEST PC AS SOURCE IN MODE 7, USING R0
                        ;*****************************************************************************
026650  005212          TS272:  INC     (R2)            ;UPDATE TEST NUMBER
026652  022712  000272          CMP     #272,(R2)       ;SEQUENCE ERROR?
026656  001013                  BNE     TS273-10        ;BR TO ERROR HALT ON SEQ ERROR
026660  012737  026126  026132  MOV     #A,@#A+4        ;LOC A+4 STORES ADDR A
026666  012700  026126          MOV     #A,R0           ;R0 STORES ADDR A
026672  010770  000004  PCN3:   MOV     PC,@4(R0)       ;EFFECTIVE ADDR IS A
026676  022737  026676  026126  CMP     #PCN3+4,@#A     ;LOC A STORES PC+4?
026704  001404                  BEQ     TS273
```

```
8387                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8388                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8389                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8390                                                    ;           WHICH FOLLOWS W/ 764           <====
8391  026706  012742  000637              MOV     #637,-(R2)      ;MOVE TO MAILBOX #  *******  637  *******
8392  026712  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8393  026714  000000                      HALT                    ;LOC A STORED WRONG VALUE
8394                                                            ;  OR SEQUENCE ERROR
8395
8396                                  ;*************************************************************************************
8397                                  ;TEST 273       TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0
8398                                  ;*************************************************************************************
8399  026716  005212                  TS273:  INC     (R2)            ;UPDATE TEST NUMBER
8400  026720  022712  000273                  CMP     #273,(R2)       ;SEQUENCE ERROR?
8401  026724  001011                          BNE     TS274-10        ;BR TO ERROR HALT ON SEQ ERROR
8402  026726  012737  026130  026126          MOV     #A+2,@#A        ;LOC A STORES ADDR A+2
8403  026734  010777  177166          PCN4:   MOV     PC,@A           ;EFFECTIVE ADDR IS A+2
8404  026740  022737  026740  026130          CMP     #PCN4+4,@#A+2   ;LOC A+2 STORES PC+4?
8405  026746  001404                          BEQ     TS274
8406                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8407                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8408                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8409                                                    ;           WHICH FOLLOWS W/ 766           <====
8410  026750  012742  000640              MOV     #640,-(R2)      ;MOVE TO MAILBOX #  *******  640  *******
8411  026754  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8412  026756  000000                      HALT                    ;LOC A+2 STORED WRONG VALUE
8413                                                            ;  OR SEQUENCE ERROR
8414
8415                                  ;*************************************************************************************
8416                                  ;TEST 274       TEST PC AS SOURCE IN RELATIVE MODE ,USING R0
8417                                  ;*************************************************************************************
8418  026760  005212                  TS274:  INC     (R2)            ;UPDATE TEST NUMBER
8419  026762  022712  000274                  CMP     #274,(R2)       ;SEQUENCE ERROR?
8420  026766  001010                          BNE     TS275-10        ;BR TO ERROR HALT ON SEQ ERROR
8421  026770  005037  026126                  CLR     @#A             ;CLEAR A
8422  026774  010767  177126          PCN5:   MOV     PC,A            ;EFFECTIVE ADDR IS A
8423  027000  022737  027000  026126          CMP     #PCN5+4,@#A     ;LOC A STORES PC+4?
8424  027006  001404                          BEQ     TS275
8425                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8426                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8427                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8428                                                    ;           WHICH FOLLOWS W/ 767           <====
8429  027010  012742  000641              MOV     #641,-(R2)      ;MOVE TO MAILBOX #  *******  641  *******
8430  027014  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8431  027016  000000                      HALT                    ;LOCATION A STORED WRONG VALUE
8432                                                            ;  OR SEQUENCE ERROR
8433
8434                                  ;*************************************************************************************
8435                                  .SBTTL  THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.
8436                                  ;*************************************************************************************
8437                                  ;TEST 275       TEST SUB INSTRUCTION, SM=0, DM=2
8438                                  ;*************************************************************************************
8439  027020  005212                  TS275:  INC     (R2)            ;UPDATE TEST NUMBER
8440  027022  022712  000275                  CMP     #275,(R2)       ;SEQUENCE ERROR?
8441  027026  001013                          BNE     TS276-10        ;BR TO ERROR HALT ON SEQ ERROR
8442  027030  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
```

```
8443  027036  012701  050505          MOV     #050505,R1      ;SET UP R1
8444  027042  005000                  CLR     R0              ;CLEAR R0
8445  027044  160120                  SUB     R1,(R0)+        ;SUBTRACTION, SM=0,DM=2
8446  027046  022737  002020  000000  CMP     #2020,@#0       ;CHECK DIFFERENCE AT LOC 0
8447  027054  001404                  BEQ     TS276
8448                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8449                                                          ;           CONDITIONAL BRANCH INST. AND   <====
8450                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
8451                                                          ;           WHICH FOLLOWS W/ 764           <====
8452  027056  012742  000642          MOV     #642,-(R2)      ;MOVE TO MAILBOX #  *******  642  *******
8453  027062  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8454  027064  000000                  HALT                    ;WRONG RESULT FROM SUBTRACTION
8455                                                          ;  OR SEQUENCE ERROR
8456
8457                            ;*****************************************************************************
8458                            ;TEST 276       TEST MFPD WITH R0, IN MODE 2
8459                            ;*****************************************************************************
8460  027066  005212            TS276:  INC     (R2)            ;UPDATE TEST NUMBER
8461  027070  022712  000276          CMP     #276,(R2)       ;SEQUENCE ERROR?
8462  027074  001020                  BNE     TS277-10        ;BR TO ERROR HALT ON SEQ ERROR
8463  027076  012737  052525  000000  MOV     #052525,@#0     ;SET UP LOC 0
8464  027104  005000                  CLR     R0              ;CLEAR R0
8465  027106  012767  170000  150662  MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
8466  027114  012706  027352          MOV     #USTBOT,R6      ;SET USER STACK POINTER
8467  027120  106520                  MFPD    (R0)+           ;MODE 2, MFPD
8468  027122  005067  150650          CLR     PS              ;SET KERNEL MODE
8469  027126  022767  052525  000214  CMP     #052525,USTBOT-2        ;CHECK DATA ON STACK
8470  027134  001404                  BEQ     BRMFPD          ;BR IF NO ERROR
8471  027136  012742  000643          MOV     #643,-(R2)      ;MOVE TO MAILBOX #  *******  643  *******
8472  027142  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8473  027144  000000                  HALT                    ;INCORRECT DATA FROM MFPD
8474  027146                  BRMFPD:
8475
8476                            ;*****************************************************************************
8477                            ;TEST 277       TEST MTPD WITH R0, IN MODE 2
8478                            ;*****************************************************************************
8479  027146  005212            TS277:  INC     (R2)            ;UPDATE TEST NUMBER
8480  027150  022712  000277          CMP     #277,(R2)       ;SEQUENCE ERROR?
8481  027154  001026                  BNE     END1            ;BR TO ERROR HALT ON SEQ ERROR
8482  027156  012767  170000  150612  MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
8483  027164  012706  027352          MOV     #USTBOT,R6      ;SET USER STACK POINTER
8484  027170  012746  125252          MOV     #125252,-(R6)   ;PUSH DATA IN USER STACK
8485  027174  012737  000000  000000  MOV     #0,@#0          ;CLEAR LOC 0
8486  027202  005000                  CLR     R0              ;CLEAR R0
8487  027204  106620                  MTPD    (R0)+           ;MODE 2, MTPD
8488  027206  005067  150564          CLR     PS              ;SET KERNEL MODE
8489  027212  022737  125252  000000  CMP     #125252,@#0     ;CHECK DATA ON LOC 0
8490  027220  001514                  BEQ     SECPRT          ;BR TO TRAP TEST IF NO ERROR
8491  027222  012742  000644          MOV     #644,-(R2)      ;MOVE TO MAILBOX #  *******  644  *******
8492  027226  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8493  027230  000000                  HALT                    ;INCORRECT DATA FROM MTPD
8494  027232                  END1:
8495  027232  012742  000645          MOV     #645,-(R2)      ;MOVE TO MAILBOX #  *******  645  *******
8496  027236  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8497  027240  000000                  HALT                    ;SEQUENCE ERROR
8498
```

```
8499
8500  027242  000402          BRTAB:  BR      .+6
8501  027244  001002                  BNE     .+6
8502  027246  001402                  BEQ     .+6
8503  027250  002002                  BGE     .+6
8504  027252  002402                  BLT     .+6
8505  027254  003002                  BGT     .+6
8506  027256  003402                  BLE     .+6
8507  027260  100002                  BPL     .+6
8508  027262  100402                  BMI     .+6
8509  027264  101002                  BHI     .+6
8510  027266  101402                  BLOS    .+6
8511  027270  102002                  BVC     .+6
8512  027272  102402                  BVS     .+6
8513  027274  103002                  BCC     .+6             ;SAME AS BHIS
8514  027276  103402                  BCS     .+6             ;SAME AS BLO
8515
8516          000002          .RADIX  2
8517  027300  177777          YNTAB:  1111111111111111        ;BR
8518  027302  170360                  1111000011110000        ;BNE:  Z=0
8519  027304  007417                  0000111100001111        ;BEQ:  Z=1
8520  027306  146063                  1100110000110011        ;BGE:  N XOR V =0
8521  027310  031714                  0011001111001100        ;BLT:  N XOR V =1
8522  027312  140060                  1100000000110000        ;BGT:  Z+(N XOR V) =0
8523  027314  037717                  0011111111001111        ;BLE:  Z+(N XOR V) =1
8524
8525  027316  177400                  1111111100000000        ;BPL:  N=0
8526  027320  000377                  0000000011111111        ;BMI:  N=1
8527  027322  120240                  1010000010100000        ;BHI:  C+Z=0
8528  027324  057537                  0101111101011111        ;BLOS: C+Z=1
8529  027326  146314                  1100110011001100        ;BVC:  V=0
8530  027330  031463                  0011001100110011        ;BVS:  V=1
8531  027332  125252                  1010101010101010        ;BCC:  C=0
8532  027334  052525                  0101010101010101        ;BCS:  C=1
8533          000010          .RADIX  8
8534
8535                          .EVEN
8536  027336  000006          .BLKW   6
8537  027352                  USTBOT:
8538                          ;*****************************************************************************************
8539                          ; THE FOLLOWING ARE SPECIAL CPU TRAP
8540                          ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
8541                          ;
8542                          ;*****************************************************************************************
8543
8544  027352                  TO4:
8545  027352  012742  000646          MOV     #646,-(R2)      ;MOVE TO MAILBOX #  *******  646  *******
8546  027356  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8547  027360  000000                  HALT                    ;TRAPPED THRU LOC. 4
8548  027362                  TO10:
8549  027362  012742  000647          MOV     #647,-(R2)      ;MOVE TO MAILBOX #  *******  647  *******
8550  027366  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8551  027370  000000                  HALT                    ;TRAPPED THRU LOC. 10
8552  027372                  TO14:
8553  027372  012742  000650          MOV     #650,-(R2)      ;MOVE TO MAILBOX #  *******  650  *******
8554  027376  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
8555  027400  000000
8556  027402
8557  027402  012742  000651
8558  027406  005242
8559  027410  000000
8560  027412
8561  027412  012742  000652
8562  027416  005242
8563  027420  000000
8564  027422
8565  027422  012742  000653
8566  027426  005242
8567  027430  000000
8568  027432
8569  027432  012742  000654
8570  027436  005242
8571  027440  000000
8572  027442
8573  027442  012742  000655
8574  027446  005242
8575  027450  000000
8576
8577  027452
```

```
        HALT                            ;TRAPPED THRU LOC. 14
T030:
        MOV     #651,-(R2)              ;MOVE TO MAILBOX #  *******  651  *******
        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
        HALT                            ;TRAPPED THRU LOC. 30
T034:
        MOV     #652,-(R2)              ;MOVE TO MAILBOX #  *******  652  *******
        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
        HALT                            ;TRAPPED THRU LOC. 34
T0114:
        MOV     #653,-(R2)              ;MOVE TO MAILBOX #  *******  653  *******
        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
        HALT                            ;TRAPPED THRU LOC. 114
T0244:
        MOV     #654,-(R2)              ;MOVE TO MAILBOX #  *******  654  *******
        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
        HALT                            ;TRAPPED THRU LOC. 244
T0250:
        MOV     #655,-(R2)              ;MOVE TO MAILBOX #  *******  655  *******
        INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
        HALT                            ;TRAPPED THRU LOC. 250
.SBTTL ** STARTING OF TRAP TEST **
SECPRT:
```

```
8578
8579            000000                  .REPT 0
8580
8581
8582                                    PART TWO:               F11 TRAP TEST, THIS IS THE SECOND
8583                                                            PART OF THE MAIN PROGRAM.
8584
8585
8586                                    ABSTRACT
8587
8588                                    THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE
8589                                    TRAPS.  ALSO TESTED ARE TRAP OVERFLOW CONDITIONS,
8590                                    ODDITIES OF REGISTER 6, INTERRUPTS , THE RESET AND WAIT INSTRUCTIONS.
8591
8592                                    .ENDR
8593
8594                                    .LIST ME
8595                                    .NLIST MC,MD,CND
8596                                    .ABS
8597            000006                  SP=%6
8598            000006                  R6=%6
8599            000003                  TAB=%3
8600            000001                  LAST=%1
8601            000005                  FIRST=%5
8602            000002                  R2=%2
8603            000000                  HLT=HALT
8604            000003                  TRT=3
8605            000004                  ITRAP5=4
8606            000004                  RTRAP5=4                ;RESERVED INST AND ILLEGAL ADDRESSES
8607            000014                  RTRAP4=14               ;FOR TRACE TRAP
8608            000030                  RTRAP3=30               ;FOR EMULATOR TRAP
8609            000020                  RTRAP2=20               ;FOR IOT TRAP
8610            000034                  RTRAP1=34               ;FOR TRAP INST
8611            177564                  TTCSR=177564
8612            177560                  TRCSR=177560
8613            177564                  TPS=177564
8614            177566                  TPB=177566
8615            000240                  BELL=240
8616            000240                  NOP=240
8617            177776                  STATUS=177776
8618            000077                  TRAPA=77
8619            000010                  RTRAP=10
8620            004700                  ILLA=004700
8621            000100                  ILLB=100
8622            177776                  CC=177776
8623            001000                  BUFF=STBOT
8624
8625
8626
```

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

027452  000167  000024                JMP     TESTN1
027456  000000                  K1:     0
027460  000000                  K2:     0
027462  000000                  K3:     0
027464  000000                  K4:     0
027466  000000                  K5:     0
027470  000000                  K6:     0
027472  052525                  K7:     052525
027474  052400                  K10:    052400
027476  000000                  K11:    0
027500  000000                  K12:    0

027502                          TESTN1:
                                ;*************************************************************************************
                                ;TEST 300       TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
                                ;*************************************************************************************
027502  005212                  TS300:  INC     (R2)            ;UPDATE TEST NUMBER
027504  022712  000300                  CMP     #300,(R2)       ;SEQUENCE ERROR?
027510  001127                          BNE     TS301-10        ;BR TO ERROR HALT ON SEQ ERROR
027512  005006                          CLR     %6
027514  112667  150260                  MOVB    (6)+,HERE       ;SIX SHOULD INCREMENT BY TWO
027520  020627  000002                  CMP     %6,#2
027524  001404                          BEQ     BR1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
027526  012742  000656                  MOV     #656,-(R2)      ;MOVE TO MAILBOX #  *******  656  *******
027532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
027534  000000                          HALT                    ;R6 DID NOT AUTO INCREMENT BY TWO

027536  012706  001000          BR1:    MOV     #1000,%6
027542  114627  000000                  MOVB    -(6),#HERE      ;SHOULD DECREMENT BY TWO
027546  020627  000776                  CMP     %6,#776
027552  001404                          BEQ     BR2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 756           <====
027554  012742  000657                  MOV     #657,-(R2)      ;MOVE TO MAILBOX #  *******  657  *******
027560  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
027562  000000                          HALT                    ;R6 DID NOT AUTO DECREMENT BY 2

027564  005006          BR2:    CLR     %6
027566  112626                  MOVB    (6)+,(6)+       ;DOUBLES AUTO INCREMENT OF R6
027570  020627  000004                  CMP     %6,#4
027574  001404                          BEQ     BR3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 745           <====
027576  012742  000660                  MOV     #660,-(R2)      ;MOVE TO MAILBOX #  *******  660  *******
```

```
8683  027602  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8684  027604  000000                  HALT                    ;WRONG AUTO INCREMENT OF R6
8685
8686  027606  005006          BR3:    CLR     %6
8687  027610  005004                  CLR     %4
8688  027612  122624                  CMPB    (6)+,(4)+       ;TEST INCREMENT OF R6
8689  027614  020627  000002          CMP     %6,#2
8690  027620  001404                  BEQ     BR4
8691                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8692                                                          ;           CONDITIONAL BRANCH INST. AND   <====
8693                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
8694                                                          ;           WHICH FOLLOWS W/ 733           <====
8695  027622  012742  000661                MOV     #661,-(R2)      ;MOVE TO MAILBOX #  *******  661  *******
8696  027626  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8697  027630  000000                  HALT                    ;WRONG INCREMENT OF R6
8698
8699  027632  005006          BR4:    CLR     %6
8700  027634  005004                  CLR     %4
8701  027636  122426                  CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
8702  027640  020627  000002          CMP     %6,#2
8703  027644  001404                  BEQ     BR5
8704                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8705                                                          ;           CONDITIONAL BRANCH INST. AND   <====
8706                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
8707                                                          ;           WHICH FOLLOWS W/ 721           <====
8708  027646  012742  000662                MOV     #662,-(R2)      ;MOVE TO MAILBOX #  *******  662  *******
8709  027652  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8710  027654  000000                  HALT                    ;WRONG INCREMENT OF R6
8711
8712  027656  005006          BR5:    CLR     %6
8713  027660  005004                  CLR     %4
8714  027662  122624                  CMPB    (6)+,(4)+       ;TEST INCREMENT OF R4
8715  027664  020427  000001          CMP     %4,#1
8716  027670  001404                  BEQ     BR6
8717                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8718                                                          ;           CONDITIONAL BRANCH INST. AND   <====
8719                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
8720                                                          ;           WHICH FOLLOWS W/ 707           <====
8721  027672  012742  000663                MOV     #663,-(R2)      ;MOVE TO MAILBOX #  *******  663  *******
8722  027676  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8723  027700  000000                  HALT                    ;WRONG INCREMENT OF R4
8724  027702  005006          BR6:    CLR     %6
8725  027704  005004                  CLR     %4
8726  027706  122426                  CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
8727  027710  020627  000002          CMP     %6,#2
8728  027714  001404                  BEQ     BR7
8729                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8730                                                          ;           CONDITIONAL BRANCH INST. AND   <====
8731                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
8732                                                          ;           WHICH FOLLOWS W/ 675           <====
8733  027716  012742  000664                MOV     #664,-(R2)      ;MOVE TO MAILBOX #  *******  664  *******
8734  027722  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8735  027724  000000                  HALT                    ;WRONG INCREMENT OF R6
8736
8737  027726  005006          BR7:    CLR     %6
8738  027730  005004                  CLR     %4
```

```
8739  027732  122426                        CMPB    (4)+,(6)+       ;TEST INCREMENT OF R4
8740  027734  020427  000001                CMP     %4,#1
8741  027740  001404                        BEQ     BR10
8742                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8743                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8744                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8745                                                                  ;           WHICH FOLLOWS W/ 663           <====
8746  027742  012742  000665                MOV     #665,-(R2)      ;MOVE TO MAILBOX #  *******  665  *******
8747  027746  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8748  027750  000000                        HALT                    ;WRONG INCREMENT OF R4
8749
8750  027752  012706  001000        BR10:   MOV     #1000,%6
8751  027756  124627  000000                CMPB    -(6),#HERE      ;TEST DECREMENT OF R6
8752  027762  022706  000776                CMP     #776,%6
8753  027766  001404                        BEQ     TS301
8754                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8755                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8756                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8757                                                                  ;           WHICH FOLLOWS W/ 650           <====
8758  027770  012742  000666                MOV     #666,-(R2)      ;MOVE TO MAILBOX #  *******  666  *******
8759  027774  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8760  027776  000000                        HALT                    ;WRONG DECREMENT OF R6,OR WRONG $TSTNM
8761                                                                  ;  OR SEQUENCE ERROR
8762                                ;*************************************************************************
8763                                ;TEST 301       TEST TRANSFER OF .BYTE USING R6
8764                                ;*************************************************************************
8765  030000  005212                TS301:  INC     (R2)            ;UPDATE TEST NUMBER
8766  030002  022712  000301                CMP     #301,(R2)       ;SEQUENCE ERROR?
8767  030006  001133                        BNE     TS302-10        ;BR TO ERROR HALT ON SEQ ERROR
8768  030010  012767  123456  177450        MOV     #123456,K5
8769  030016  012767  050505  177432        MOV     #050505,K1
8770  030024  012705  027456                MOV     #K1,%5          ;%5=(050505)K1
8771  030030  012706  027466                MOV     #K5,%6          ;%6=(123456)K5
8772  030034  112625                        MOVB    (6)+,(5)+       ;LOW .BYTE OF R6 TO R5
8773  030036  022767  050456  177412        CMP     #050456,K1
8774  030044  001404                        BEQ     BR11
8775                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8776                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8777                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8778                                                                  ;           WHICH FOLLOWS W/ 760           <====
8779  030046  012742  000667                MOV     #667,-(R2)      ;MOVE TO MAILBOX #  *******  667  *******
8780  030052  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8781  030054  000000                        HALT                    ;FALSE TRANSFER OF .BYTE
8782
8783  030056  012767  123456  177402  BR11:   MOV     #123456,K5
8784  030064  012767  050505  177364          MOV     #050505,K1
8785  030072  012705  027456                MOV     #K1,%5          ;%5(050505)K1
8786  030076  012706  027470                MOV     #K6,%6          ;%6(123456)K5
8787  030102  114625                        MOVB    -(6),(5)+       ;LOW .BYTE OF R6 TO R5 (DECREMENT)
8788  030104  026727  177346  050456        CMP     K1,#050456
8789  030112  001404                        BEQ     BR12
8790                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8791                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8792                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8793                                                                  ;           WHICH FOLLOWS W/ 735           <====
8794  030114  012742  000670                MOV     #670,-(R2)      ;MOVE TO MAILBOX #  *******  670  *******
```

```
8795  030120  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8796  030122  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
8797
8798  030124  012767  123456  177324  BR12:   MOV     #123456,K1
8799  030132  012767  050505  177326          MOV     #050505,K5
8800  030140  012705  027456                  MOV     #K1,%5          ;(123456)
8801  030144  012706  027466                  MOV     #K5,%6          ;(050505)
8802  030150  112526                          MOVB    (5)+,(6)+       ;LOW OF R5 TO LOW OF R6
8803  030152  022767  050456  177306          CMP     #050456,K5
8804  030160  001404                          BEQ     BR13
8805                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8806                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8807                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8808                                                                  ;           WHICH FOLLOWS W/ 712           <====
8809  030162  012742  000671                  MOV     #671,-(R2)      ;MOVE TO MAILBOX #  *******  671  *******
8810  030166  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8811  030170  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
8812
8813  030172  012767  123456  177256  BR13:   MOV     #123456,K1
8814  030200  012767  050505  177260          MOV     #050505,K5
8815  030206  012705  027457                  MOV     #K1+1,%5        ;123456
8816  030212  012706  027466                  MOV     #K5,%6          ;050505
8817  030216  112526                          MOVB    (5)+,(6)+       ;HIGH OF R5 TO LOW OF R6
8818  030220  026727  177242  050647          CMP     K5,#050647
8819  030226  001404                          BEQ     BR14
8820                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8821                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8822                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8823                                                                  ;           WHICH FOLLOWS W/ 667           <====
8824  030230  012742  000672                  MOV     #672,-(R2)      ;MOVE TO MAILBOX #  *******  672  *******
8825  030234  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8826  030236  000000                          HALT                    ;FALSE R6 .BYTE TRANSFER
8827
8828  030240  012767  123456  177210  BR14:   MOV     #123456,K1
8829  030246  012767  050505  177212          MOV     #050505,K5
8830  030254  012705  027457                  MOV     #K1+1,%5        ;R5-123456-ODD ADDRESS
8831  030260  012706  027466                  MOV     #K5,%6          ;R6-050505--.EVEN ADDRESS
8832  030264  112625                          MOVB    (6)+,(5)+       ;LOW OF R6 TO HIGH OF R5
8833  030266  022767  042456  177162          CMP     #042456,K1
8834  030274  001404                          BEQ     TS302
8835                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8836                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8837                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8838                                                                  ;           WHICH FOLLOWS W/ 644           <====
8839  030276  012742  000673                  MOV     #673,-(R2)      ;MOVE TO MAILBOX #  *******  673  *******
8840  030302  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8841  030304  000000                          HALT                    ;FAILED LOW OF 6 TO HIGH OF 5,OR WRONG $TSTNM
8842                                                                  ;  OR SEQUENCE ERROR
8843                                ;*****************************************************************************
8844                                ;TEST 302       TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS
8845                                ;*****************************************************************************
8846  030306  005212                TS302:  INC     (R2)            ;UPDATE TEST NUMBER
8847  030310  022712  000302                  CMP     #302,(R2)       ;SEQUENCE ERROR?
8848  030314  001074                          BNE     TS303-10        ;BR TO ERROR HALT ON SEQ ERROR
8849  030316  126767  177150  177147          CMPB    K7,K7+1         ;SAME .WORD LOW TO HIGH
8850  030324  001404                          BEQ     BR15
```

```
8851                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8852                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8853                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8854                                                    ;           WHICH FOLLOWS W/ 773           <====
8855  030326  012742  000674              MOV     #674,-(R2)      ;MOVE TO MAILBOX #  *******  674  *******
8856  030332  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8857  030334  000000                      HALT                    ;SHOULD COMPARE LOW TO HIGH
8858
8859  030336  126767  177131  177126 BR15: CMPB    K7+1,K7         ;COMPARE ODD TO .EVEN SAME .WORD
8860  030344  001404                      BEQ     BR16
8861                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8862                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8863                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8864                                                    ;           WHICH FOLLOWS W/ 763           <====
8865  030346  012742  000675              MOV     #675,-(R2)      ;MOVE TO MAILBOX #  *******  675  *******
8866  030352  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8867  030354  000000                      HALT                    ;ODD TO .EVEN .BYTE FAILURE
8868
8869  030356  126767  177113  177106 BR16: CMPB    K10+1,K7        ;SEQUENTIAL .BYTES
8870  030364  001404                      BEQ     BR17
8871                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8872                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8873                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8874                                                    ;           WHICH FOLLOWS W/ 753           <====
8875  030366  012742  000676              MOV     #676,-(R2)      ;MOVE TO MAILBOX #  *******  676  *******
8876  030372  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8877  030374  000000                      HALT                    ;ODD TO .EVEN FAILED
8878
8879  030376  126767  177072  177064 BR17: CMPB    K10,K6
8880  030404  001404                      BEQ     BR20
8881                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8882                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8883                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8884                                                    ;           WHICH FOLLOWS W/ 743           <====
8885  030406  012742  000677              MOV     #677,-(R2)      ;MOVE TO MAILBOX #  *******  677  *******
8886  030412  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8887  030414  000000                      HALT                    ;.EVEN TO EVEN FAILED
8888  030416  126767  177051  177051 BR20: CMPB    K7+1,K10+1
8889  030424  001404                      BEQ     BR21
8890                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8891                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8892                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8893                                                    ;           WHICH FOLLOWS W/ 733           <====
8894  030426  012742  000700              MOV     #700,-(R2)      ;MOVE TO MAILBOX #  *******  700  *******
8895  030432  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8896  030434  000000                      HALT                    ;ODD TO ODD FAILED
8897
8898  030436  126767  177032  177031 BR21: CMPB    K10,K10+1
8899  030444  001004                      BNE     BR22
8900                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8901                                                    ;           CONDITIONAL BRANCH INST. AND   <====
8902                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
8903                                                    ;           WHICH FOLLOWS W/ 723           <====
8904  030446  012742  000701              MOV     #701,-(R2)      ;MOVE TO MAILBOX #  *******  701  *******
8905  030452  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8906  030454  000000                      HALT                    ;LOW TO HIGH IN SAME .WORD FAILED
```

```
8907
8908  030456  126767  177013  177011  BR22:   CMPB    K10+1,K10+1
8909  030464  001404                  BEQ     BR23
8910                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8911                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8912                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8913                                                  ;           WHICH FOLLOWS W/ 713           <====
8914  030466  012742  000702                  MOV     #702,-(R2)      ;MOVE TO MAILBOX #  *******  702  *******
8915  030472  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8916  030474  000000                          HALT                    ;HIGH TO LOW IN SAME .WORD FAILED
8917
8918  030476  126767  176772  176767  BR23:   CMPB    K10,K7+1
8919  030504  001004                  BNE     TS303
8920                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8921                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8922                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8923                                                  ;           WHICH FOLLOWS W/ 703           <====
8924  030506  012742  000703                  MOV     #703,-(R2)      ;MOVE TO MAILBOX #  *******  703  *******
8925  030512  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8926  030514  000000                          HALT                    ;.EVEN TO ODD FAILED,OR WRONG $TSTNM
8927                                                                  ;  OR SEQUENCE ERROR
8928
8929
8930                                  ;*************************************************************************************
8931                                  ;TEST 303       TEST THE CC BITS
8932                                  ;*************************************************************************************
8933  030516  005212                  TS303:  INC     (R2)            ;UPDATE TEST NUMBER
8934  030520  022712  000303                  CMP     #303,(R2)       ;SEQUENCE ERROR?
8935  030524  001053                          BNE     TS304-10        ;BR TO ERROR HALT ON SEQ ERROR
8936  030526  000277                          SCC                     ;SET STATUS
8937  030530  005067  147242                  CLR     STATUS          ;CLEAR STATUS
8938  030534  103004                          BCC     BR33
8939                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8940                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8941                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8942                                                  ;           WHICH FOLLOWS W/ 773           <====
8943  030536  012742  000704                  MOV     #704,-(R2)      ;MOVE TO MAILBOX #  *******  704  *******
8944  030542  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8945  030544  000000                          HALT                    ;C NOT CLEAR
8946  030546                          BR33:
8947  030546  102004                          BVC     BR34
8948                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8949                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8950                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8951                                                  ;           WHICH FOLLOWS W/ 766           <====
8952  030550  012742  000705                  MOV     #705,-(R2)      ;MOVE TO MAILBOX #  *******  705  *******
8953  030554  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8954  030556  000000                          HALT                    ;V NOT CLEAR
8955  030560                          BR34:
8956  030560  001004                          BNE     BR35
8957                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8958                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8959                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8960                                                  ;           WHICH FOLLOWS W/ 761           <====
8961  030562  012742  000706                  MOV     #706,-(R2)      ;MOVE TO MAILBOX #  *******  706  *******
8962  030566  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
8963  030570  000000                        HALT                    ;Z NOT CLEAR
8964  030572                        BR35:
8965  030572  100004                        BPL     BR36
8966                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8967                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8968                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8969                                                                  ;           WHICH FOLLOWS W/ 754           <====
8970  030574  012742  000707                MOV     #707,-(R2)      ;MOVE TO MAILBOX #  *******  707  *******
8971  030600  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8972  030602  000000                        HALT                    ;N NOT CLEAR
8973  030604  000257                BR36:   CCC                     ;CLEAR CONDITION CODES
8974  030606  052767  000017  147162        BIS     #17,STATUS      ;SET STATUS TO ONES
8975
8976  030614  103404                        BCS     BR37
8977                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8978                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8979                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8980                                                                  ;           WHICH FOLLOWS W/ 743           <====
8981  030616  012742  000710                MOV     #710,-(R2)      ;MOVE TO MAILBOX #  *******  710  *******
8982  030622  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8983  030624  000000                        HALT                    ;C NOT SET
8984  030626                        BR37:
8985  030626  102404                        BVS     BR40
8986                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8987                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8988                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8989                                                                  ;           WHICH FOLLOWS W/ 736           <====
8990  030630  012742  000711                MOV     #711,-(R2)      ;MOVE TO MAILBOX #  *******  711  *******
8991  030634  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
8992  030636  000000                        HALT                    ;V NOT SET
8993  030640                        BR40:
8994  030640  001404                        BEQ     BR41
8995                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
8996                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
8997                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
8998                                                                  ;           WHICH FOLLOWS W/ 731           <====
8999  030642  012742  000712                MOV     #712,-(R2)      ;MOVE TO MAILBOX #  *******  712  *******
9000  030646  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9001  030650  000000                        HALT                    ;Z NOT SET
9002  030652                        BR41:
9003  030652  100404                        BMI     TS304
9004                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9005                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9006                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9007                                                                  ;           WHICH FOLLOWS W/ 724           <====
9008  030654  012742  000713                MOV     #713,-(R2)      ;MOVE TO MAILBOX #  *******  713  *******
9009  030660  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9010  030662  000000                        HALT                    ;N NOT SET,OR WRONG $TSTNM
9011                                                                  ;  OR SEQUENCE ERROR
9012                          ;*************************************************************************************
9013                          ;TEST 304       TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION
9014                          ;*************************************************************************************
9015  030664  005212          TS304:  INC     (R2)            ;UPDATE TEST NUMBER
9016  030666  022712  000304        CMP     #304,(R2)       ;SEQUENCE ERROR?
9017  030672  001006                BNE     RETA            ;BR TO ERROR HALT ON SEQ ERROR
9018  030674  012706  001000        MOV     #BUFF,SP        ;STACK POINTER SETUP
```

```
                        MOV     #RETAH,RTRAP    ;RETURN LOCATION
                        TRAPA                   ;RESERVED INSTRUCTION, SHOULD TRAP
                RETA:
                        MOV     #714,-(R2)      ;MOVE TO MAILBOX #  *******  714  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        HALT                    ;RESERVE INSTRUCTION DIDN'T TRAP,OR WRONG $TSTNM
                RETAH:
                ;*****************************************************************************************
                ;TEST 305       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
                ;*****************************************************************************************
                TS305:  INC     (R2)            ;UPDATE TEST NUMBER
                        CMP     #305,(R2)       ;SEQUENCE ERROR?
                        BNE     TS306-10        ;BR TO ERROR HALT ON SEQ ERROR
                        MOV     #BUFF,SP        ;STACK POINTER SETUP
                        MOV     #RETB,RTRAP     ;RETURN POINTER
                        TRAPA                   ;RESERVED INSTRUCTION
                RETB:   CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
                        BEQ     TS306
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 766           <====
                        MOV     #715,-(R2)      ;MOVE TO MAILBOX #  *******  715  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
                                                ;  OR SEQUENCE ERROR
                ;*****************************************************************************************
                ;TEST 306       TEST THAT PROPER P.C. IS SAVED
                ;*****************************************************************************************
                TS306:  INC     (R2)            ;UPDATE TEST NUMBER
                        CMP     #306,(R2)       ;SEQUENCE ERROR?
                        BNE     TS307-10        ;BR TO ERROR HALT ON SEQ ERROR
                        MOV     #BUFF,SP        ;STACK POINTER SETUP
                        MOV     #RETC,RTRAP     ;RETURN FROM TRAP POINTER
                INSTC:  TRAPA                   ;TRAP ON THIS INSTRUCTION
                RETC:   CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
                        BEQ     TS307
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 765           <====
                        MOV     #716,-(R2)      ;MOVE TO MAILBOX #  *******  716  *******
                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
                        HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
                                                ;  OR SEQUENCE ERROR
                ;*****************************************************************************************
                ;TEST 307       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
                ;*****************************************************************************************
                TS307:  INC     (R2)            ;UPDATE TEST NUMBER
                        CMP     #307,(R2)       ;SEQUENCE ERROR?
                        BNE     TS310-10        ;BR TO ERROR HALT ON SEQ ERROR
                        MOV     #BUFF,SP        ;SET UP
                        MOV     #RETD,RTRAP     ;SET UP
                        CLR     CC              ;CLEAR CC AND PRIORITY
                        CCC
                        TRAPA                   ;TRAP
```

```
 9075  031060  026727  147712  000000  RETD:   CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
 9076  031066  001404                          BEQ     1$
 9077                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9078                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9079                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9080                                                                  ;           WHICH FOLLOWS W/ 762           <====
 9081  031070  012742  000717                  MOV     #717,-(R2)      ;MOVE TO MAILBOX #  *******  717  *******
 9082  031074  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9083  031076  000000                          HALT                    ;INCORRECT STATUS
 9084  031100  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
 9085  031104  012767  031124  146676          MOV     #RETE,RTRAP     ;SET UP
 9086  031112  012767  000357  146656          MOV     #357,CC         ;SET PRIORITY
 9087  031120  000277                          SCC                     ;SET CC
 9088  031122  000077                          TRAPA                   ;TRAP
 9089  031124  026727  147646  000357  RETE:   CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
 9090  031132  001404                          BEQ     TS310
 9091                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9092                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9093                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9094                                                                  ;           WHICH FOLLOWS W/ 740           <====
 9095  031134  012742  000720                  MOV     #720,-(R2)      ;MOVE TO MAILBOX #  *******  720  *******
 9096  031140  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9097  031142  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
 9098                                                                  ;  OR SEQUENCE ERROR
 9099                                  ;*****************************************************************************
 9100                                  ;TEST 310       TEST THAT "NEW" STATUS IS CORRECT
 9101                                  ;*****************************************************************************
 9102  031144  005212                  TS310:  INC     (R2)            ;UPDATE TEST NUMBER
 9103  031146  022712  000310                  CMP     #310,(R2)       ;SEQUENCE ERROR?
 9104  031152  001110                          BNE     STPP            ;BR TO ERROR HALT ON SEQ ERROR
 9105  031154  012706  001000                  MOV     #BUFF,SP
 9106  031160  012767  031174  146622          MOV     #RETF,RTRAP
 9107  031166  005067  146620                  CLR     RTRAP+2         ;CLEAR FUTURE PRIORITY AND CC
 9108  031172  000077                          TRAPA
 9109  031174                          RETF:                           ;TEST FOR "C" CLEARED
 9110  031174  100004                          BPL     1$
 9111                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9112                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9113                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9114                                                                  ;           WHICH FOLLOWS W/ 766           <====
 9115  031176  012742  000721                  MOV     #721,-(R2)      ;MOVE TO MAILBOX #  *******  721  *******
 9116  031202  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9117  031204  000000                          HALT                    ;N NOT CLEARED
 9118  031206                          1$:
 9119  031206  001004                          BNE     2$
 9120                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9121                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9122                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9123                                                                  ;           WHICH FOLLOWS W/ 761           <====
 9124  031210  012742  000722                  MOV     #722,-(R2)      ;MOVE TO MAILBOX #  *******  722  *******
 9125  031214  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9126  031216  000000                          HALT                    ;Z NOT CLEARED
 9127  031220                          2$:
 9128  031220  102004                          BVC     3$
 9129                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9130                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
```

```
9131
9132                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9133                                                    ;           WHICH FOLLOWS W/ 754           <====
9133  031222  012742  000723          MOV     #723,-(R2)      ;MOVE TO MAILBOX #  *******  723  *******
9134  031226  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9135  031230  000000                  HALT                    ;V NOT CLEARED
9136  031232                  3$:
9137  031232  103004                  BCC     4$
9138                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9139                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9140                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9141                                                    ;           WHICH FOLLOWS W/ 747           <====
9142  031234  012742  000724          MOV     #724,-(R2)      ;MOVE TO MAILBOX #  *******  724  *******
9143  031240  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9144  031242  000000                  HALT                    ;C NOT CLEARED
9145  031244  032767  000340  146524  4$:     BIT     #340,CC         ;TEST PRIORITY
9146  031252  001404                  BEQ     5$
9147                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9148                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9149                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9150                                                    ;           WHICH FOLLOWS W/ 737           <====
9151  031254  012742  000725          MOV     #725,-(R2)      ;MOVE TO MAILBOX #  *******  725  *******
9152  031260  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9153  031262  000000                  HALT                    ;PRIORITY NOT ZERO
9154  031264  012706  001000  5$:     MOV     #BUFF,SP
9155  031270  012767  031306  146512          MOV     #RETG,RTRAP
9156  031276  012767  000357  146506          MOV     #357,RTRAP+2    ;SET NEW "CC" AND PRIORITY
9157  031304  000077                  TRAPA                   ;TRAP HERE
9158  031306                  RETG:
9159  031306  100404                  BMI     1$
9160                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9161                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9162                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9163                                                    ;           WHICH FOLLOWS W/ 721           <====
9164  031310  012742  000726          MOV     #726,-(R2)      ;MOVE TO MAILBOX #  *******  726  *******
9165  031314  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9166  031316  000000                  HALT                    ;N NOT SET
9167  031320                  1$:
9168  031320  001404                  BEQ     2$
9169                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9170                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9171                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9172                                                    ;           WHICH FOLLOWS W/ 714           <====
9173  031322  012742  000727          MOV     #727,-(R2)      ;MOVE TO MAILBOX #  *******  727  *******
9174  031326  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9175  031330  000000                  HALT                    ;Z NOT SET
9176  031332                  2$:
9177  031332  102404                  BVS     3$
9178                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9179                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9180                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9181                                                    ;           WHICH FOLLOWS W/ 707           <====
9182  031334  012742  000730          MOV     #730,-(R2)      ;MOVE TO MAILBOX #  *******  730  *******
9183  031340  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9184  031342  000000                  HALT                    ;V NOT SET
9185  031344                  3$:
9186  031344  103404                  BCS     4$
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 702           <====
031346  012742  000731              MOV     #731,-(R2)    ;MOVE TO MAILBOX #  *******  731  *******
031352  005242                      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
031354  000000                      HALT                  ;C NOT SET
031356  016706  146414      4$:     MOV     CC,SP
031362  042706  000017              BIC     #17,SP
031366  022706  000340              CMP     #340,SP
031372  001404                      BEQ     STPPA
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 667           <====
031374                          STPP:
031374  012742  000732              MOV     #732,-(R2)    ;MOVE TO MAILBOX #  *******  732  *******
031400  005242                      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
031402  000000                      HALT                  ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
031404  012767  000012  146376  STPPA:  MOV     #12,10
031412  005067  146374              CLR     12
                        ;*************************************************************************************
                        ;TEST 311       TEST THAT A TRAP OCCURS FOR A "TRAP" INSTRUCTION
                        ;*************************************************************************************
031416  005212          TS311:  INC     (R2)          ;UPDATE TEST NUMBER
031420  022712  000311              CMP     #311,(R2)     ;SEQUENCE ERROR?
031424  001013                      BNE     TS312-10      ;BR TO ERROR HALT ON SEQ ERROR
031426  012767  000012  146354      MOV     #12,10
031434  005067  146352              CLR     12
031440  012706  001000              MOV     #BUFF,SP      ;STACK POINTER SETUP
031444  012767  031464  146362      MOV     #RETA1,RTRAP1 ;RETURN LOCATION
031452  104400                      TRAP                  ;RESERVED INSTRUCTION, SHOULD TRAP
031454  012742  000733              MOV     #733,-(R2)    ;MOVE TO MAILBOX #  *******  733  *******
031460  005242                      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
031462  000000                      HALT                  ;TRAP DIDN'T TRAP,OR WRONG $TSTNM
031464                  RETA1:
                        ;*************************************************************************************
                        ;TEST 312       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
                        ;*************************************************************************************
031464  005212          TS312:  INC     (R2)          ;UPDATE TEST NUMBER
031466  022712  000312              CMP     #312,(R2)     ;SEQUENCE ERROR?
031472  001011                      BNE     TS313-10      ;BR TO ERROR HALT ON SEQ ERROR
031474  012706  001000              MOV     #BUFF,SP      ;STACK POINTER SETUP
031500  012767  031510  146326      MOV     #RETB1,RTRAP1 ;RETURN POINTER
031506  104400                      TRAP                  ;RESERVED INSTRUCTION
031510  020627  000774      RETB1:  CMP     SP,#BUFF-4    ;TEST DECREMENT OF SP
031514  001404                      BEQ     TS313
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
031516  012742  000734              MOV     #734,-(R2)    ;MOVE TO MAILBOX #  *******  734  *******
031522  005242                      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
031524  000000                      HALT                  ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
                                                        ;  OR SEQUENCE ERROR
                        ;*************************************************************************************
```

```
                                ;TEST 313       TEST THAT PROPER P.C. IS SAVED
                                ;*****************************************************************************
031526  005212                  TS313:  INC     (R2)            ;UPDATE TEST NUMBER
031530  022712  000313                  CMP     #313,(R2)       ;SEQUENCE ERROR?
031534  001012                          BNE     TS314-10        ;BR TO ERROR HALT ON SEQ ERROR
031536  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
031542  012767  031552  146264          MOV     #RETC1,RTRAP1   ;RETURN FROM TRAP POINTER
031550  104400                          TRAP                    ;TRAP ON THIS INSTRUCTION
031552  022767  031552  147214  RETC1:  CMP     #.,BUFF-4       ;CHECK INCREMENTED P.C.
031560  001404                          BEQ     TS314
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <====
031562  012742  000735                  MOV     #735,-(R2)      ;MOVE TO MAILBOX #  *******  735  *******
031566  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031570  000000                          HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 314       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
                                ;*****************************************************************************
031572  005212                  TS314:  INC     (R2)            ;UPDATE TEST NUMBER
031574  022712  000314                  CMP     #314,(R2)       ;SEQUENCE ERROR?
031600  001036                          BNE     TS315-10        ;BR TO ERROR HALT ON SEQ ERROR
031602  012706  001000                  MOV     #BUFF,SP        ;SET UP
031606  012767  031624  146220          MOV     #RETD1,RTRAP1   ;SET UP
031614  005067  146156                  CLR     CC              ;CLEAR CC AND PRIORITY
031620  000257                          CCC
031622  104400                          TRAP                    ;TRAP
031624  026727  147146  000000  RETD1:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
031632  001404                          BEQ     1$
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
031634  012742  000736                  MOV     #736,-(R2)      ;MOVE TO MAILBOX #  *******  736  *******
031640  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031642  000000                          HALT                    ;INCORRECT STATUS
031644  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
031650  012767  031666  146156          MOV     #RETE1,RTRAP1   ;SET UP
031656  012767  000357  146112          MOV     #357,CC         ;SET PRIORITY
031664  104400                          TRAP                    ;SET CC
031666  026727  147104  000357  RETE1:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
031674  001404                          BEQ     TS315
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 741           <====
031676  012742  000737                  MOV     #737,-(R2)      ;MOVE TO MAILBOX #  *******  737  *******
031702  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
031704  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 315       TEST THAT "NEW" STATUS IS CORRECT
                                ;*****************************************************************************
031706  005212                  TS315:  INC     (R2)            ;UPDATE TEST NUMBER
```

```
 9299  031710  022712  000315                    CMP     #315,(R2)       ;SEQUENCE ERROR?
 9300  031714  001110                            BNE     TS316-10        ;BR TO ERROR HALT ON SEQ ERROR
 9301  031716  012706  001000                    MOV     #BUFF,SP
 9302  031722  012767  031736  146104            MOV     #RETF1,RTRAP1
 9303  031730  005067  146102                    CLR     RTRAP1+2        ;CLEAR FUTURE PRIORITY AND CC
 9304  031734  104400                            TRAP
 9305  031736                          RETF1:                          ;TEST FOR "C" CLEARED
 9306  031736  100004                            BPL     1$
 9307                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9308                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9309                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9310                                                                    ;           WHICH FOLLOWS W/ 766           <====
 9311  031740  012742  000740                    MOV     #740,-(R2)      ;MOVE TO MAILBOX #  *******  740  *******
 9312  031744  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9313  031746  000000                            HALT                    ;C NOT CLEARED
 9314  031750                          1$:
 9315  031750  001004                            BNE     2$
 9316                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9317                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9318                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9319                                                                    ;           WHICH FOLLOWS W/ 761           <====
 9320  031752  012742  000741                    MOV     #741,-(R2)      ;MOVE TO MAILBOX #  *******  741  *******
 9321  031756  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9322  031760  000000                            HALT                    ;Z NOT CLEARED
 9323  031762                          2$:
 9324  031762  102004                            BVC     3$
 9325                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9326                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9327                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9328                                                                    ;           WHICH FOLLOWS W/ 754           <====
 9329  031764  012742  000742                    MOV     #742,-(R2)      ;MOVE TO MAILBOX #  *******  742  *******
 9330  031770  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9331  031772  000000                            HALT                    ;V NOT CLEARED
 9332  031774                          3$:
 9333  031774  103004                            BCC     4$
 9334                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9335                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9336                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9337                                                                    ;           WHICH FOLLOWS W/ 747           <====
 9338  031776  012742  000743                    MOV     #743,-(R2)      ;MOVE TO MAILBOX #  *******  743  *******
 9339  032002  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9340  032004  000000                            HALT                    ;C NOT CLEARED
 9341  032006  032767  000340  145762  4$:       BIT     #340,CC         ;TEST PRIORITY
 9342  032014  001404                            BEQ     5$
 9343                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9344                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 9345                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 9346                                                                    ;           WHICH FOLLOWS W/ 737           <====
 9347  032016  012742  000744                    MOV     #744,-(R2)      ;MOVE TO MAILBOX #  *******  744  *******
 9348  032022  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9349  032024  000000                            HALT                    ;PRIORITY NOT ZERO
 9350  032026  012706  001000          5$:       MOV     #BUFF,SP
 9351  032032  012767  032050  145774            MOV     #RETG1,RTRAP1
 9352  032040  012767  000357  145770            MOV     #357,RTRAP1+2   ;SET NEW "CC" AND PRIORITY
 9353  032046  104400                            TRAP                    ;TRAP HERE
 9354  032050                          RETG1:
```

```
9355  032050  100404                        BMI     1$
9356                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9357                                        ;           CONDITIONAL BRANCH INST. AND   <====
9358                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9359                                        ;           WHICH FOLLOWS W/ 721           <====
9360  032052  012742  000745                MOV     #745,-(R2)      ;MOVE TO MAILBOX #  *******  745  *******
9361  032056  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9362  032060  000000                        HALT                    ;N NOT SET
9363  032062                          1$:
9364  032062  001404                        BEQ     2$
9365                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9366                                        ;           CONDITIONAL BRANCH INST. AND   <====
9367                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9368                                        ;           WHICH FOLLOWS W/ 714           <====
9369  032064  012742  000746                MOV     #746,-(R2)      ;MOVE TO MAILBOX #  *******  746  *******
9370  032070  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9371  032072  000000                        HALT                    ;Z NOT SET
9372  032074                          2$:
9373  032074  102404                        BVS     3$
9374                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9375                                        ;           CONDITIONAL BRANCH INST. AND   <====
9376                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9377                                        ;           WHICH FOLLOWS W/ 707           <====
9378  032076  012742  000747                MOV     #747,-(R2)      ;MOVE TO MAILBOX #  *******  747  *******
9379  032102  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9380  032104  000000                        HALT                    ;V NOT SET
9381  032106                          3$:
9382  032106  103404                        BCS     4$
9383                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9384                                        ;           CONDITIONAL BRANCH INST. AND   <====
9385                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9386                                        ;           WHICH FOLLOWS W/ 702           <====
9387  032110  012742  000750                MOV     #750,-(R2)      ;MOVE TO MAILBOX #  *******  750  *******
9388  032114  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9389  032116  000000                        HALT                    ;C NOT SET
9390  032120  016706  145652          4$:   MOV     CC,SP
9391  032124  042706  000017                BIC     #17,SP
9392  032130  022706  000340                CMP     #340,SP
9393  032134  001404                        BEQ     TS316
9394                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9395                                        ;           CONDITIONAL BRANCH INST. AND   <====
9396                                        ;           REPLACE THE MOVE INSTRUCTION   <====
9397                                        ;           WHICH FOLLOWS W/ 667           <====
9398  032136  012742  000751                MOV     #751,-(R2)      ;MOVE TO MAILBOX #  *******  751  *******
9399  032142  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9400  032144  000000                        HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9401                                                                ;  OR SEQUENCE ERROR
9402                            ;*************************************************************************************
9403                            ;TEST 316       TEST THAT ALL COMBINATION OF "TRAP" WILL CAUSE A TRAP
9404                            ;*************************************************************************************
9405  032146  005212            TS316:  INC     (R2)            ;UPDATE TEST NUMBER
9406  032150  022712  000316                CMP     #316,(R2)       ;SEQUENCE ERROR?
9407  032154  001011                        BNE     BR45            ;BR TO ERROR HALT ON SEQ ERROR
9408                                                                ;***** F11 **** ADD +376 TO SHORTEN TEST
9409  032156  012767  104776  000012        MOV     #TRAP+376,RB1   ;INITIALIZE BASE TRAP INSTRUCTION
9410  032164  012767  032210  145642        MOV     #RA1,34         ;RETURN FROM TRAP TO RA1
```

```
 9411  032172  012706  001000          RC1:    MOV     #BUFF,SP        ;SET UP STACK POINTER
 9412  032176  104400                  RB1:    TRAP                    ;TRAP INST WILL BE MODIFIED TO TRAP+377
 9413  032200                          BR45:
 9414  032200  012742  000752                  MOV     #752,-(R2)      ;MOVE TO MAILBOX #  *******  752  *******
 9415  032204  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9416  032206  000000                          HALT                    ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TSTNM
 9417  032210  005267  177762          RA1:    INC     RB1             ;INCREMENT TRAP INSTRUCTION
 9418  032214  022767  104777  177754          CMP     #104777,RB1     ;TRAP+377 TO UPPER LIMIT
 9419  032222  103363                          BHIS    RC1             ;HAVE WE TESTED ALL
 9420  032224  012767  000036  145602          MOV     #36,34
 9421  032232  005067  145600                  CLR     36
 9422                                  ;*****************************************************************************
 9423                                  ;TEST 317       TEST THAT A TRAP OCCURES ON AN "IOT" INSTRUCTION
 9424                                  ;*****************************************************************************
 9425  032236  005212                  TS317:  INC     (R2)            ;UPDATE TEST NUMBER
 9426  032240  022712  000317                  CMP     #317,(R2)       ;SEQUENCE ERROR?
 9427  032244  001006                          BNE     TS320-10        ;BR TO ERROR HALT ON SEQ ERROR
 9428  032246  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
 9429  032252  012767  032272  145540          MOV     #RETA2,RTRAP2   ;RETURN LOCATION
 9430  032260  000004                          IOT                     ;RESERVE INSTRUCTION, SHOULD TRAP
 9431  032262  012742  000753                  MOV     #753,-(R2)      ;MOVE TO MAILBOX #  *******  753  *******
 9432  032266  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9433  032270  000000                          HALT                    ;IOT DIDN'T TRAP,OR WRONG $TSTNM
 9434  032272                          RETA2:
 9435                                  ;*****************************************************************************
 9436                                  ;TEST 320       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
 9437                                  ;*****************************************************************************
 9438  032272  005212                  TS320:  INC     (R2)            ;UPDATE TEST NUMBER
 9439  032274  022712  000320                  CMP     #320,(R2)       ;SEQUENCE ERROR?
 9440  032300  001011                          BNE     TS321-10        ;BR TO ERROR HALT ON SEQ ERROR
 9441  032302  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
 9442  032306  012767  032316  145504          MOV     #RETB2,RTRAP2   ;RETURN POINTER
 9443  032314  000004                          IOT                     ;RESERVED INSTRUCTION
 9444  032316  020627  000774          RETB2:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
 9445  032322  001404                          BEQ     TS321
 9446                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9447                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9448                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9449                                                                  ;           WHICH FOLLOWS W/ 766           <====
 9450  032324  012742  000754                  MOV     #754,-(R2)      ;MOVE TO MAILBOX #  *******  754  *******
 9451  032330  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9452  032332  000000                          HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
 9453                                                                  ;  OR SEQUENCE ERROR
 9454                                  ;*****************************************************************************
 9455                                  ;TEST 321       TEST THAT PROPER P.C. IS SAVED
 9456                                  ;*****************************************************************************
 9457  032334  005212                  TS321:  INC     (R2)            ;UPDATE TEST NUMBER
 9458  032336  022712  000321                  CMP     #321,(R2)       ;SEQUENCE ERROR?
 9459  032342  001012                          BNE     TS322-10        ;BR TO ERROR HALT ON SEQ ERROR
 9460  032344  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
 9461  032350  012767  032360  145442          MOV     #RETC2,RTRAP2   ;RETURN FROM TRAP POINTER
 9462  032356  000004                          IOT                     ;TRAP ON THIS INSTRUCTION
 9463  032360  022767  032360  146406  RETC2:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
 9464  032366  001404                          BEQ     TS322
 9465                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9466                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
032370  012742  000755            MOV     #755,-(R2)    ;MOVE TO MAILBOX #  *******  755  *******
032374  005242                    INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
032376  000000                    HALT                  ;INCORRECT P.C.,OR WRONG $TSTNM
                                                        ;  OR SEQUENCE ERROR
                        ;***************************************************************************
                        ;TEST 322       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
                        ;***************************************************************************
032400  005212          TS322:  INC     (R2)          ;UPDATE TEST NUMBER
032402  022712  000322            CMP     #322,(R2)     ;SEQUENCE ERROR?
032406  001037                    BNE     TS323-10      ;BR TO ERROR HALT ON SEQ ERROR
032410  012706  001000            MOV     #BUFF,SP      ;SET UP
032414  012767  032432  145376    MOV     #RETD2,RTRAP2 ;SET UP
032422  005067  145350            CLR     CC            ;CLEAR CC AND PRIORITY
032426  000257                    CCC
032430  000004                    IOT                   ;TRAP
032432  026727  146340  000000    RETD2:  CMP     BUFF-2,#0     ;TEST THAT OLD STATUS WENT TO STACK
032440  001404                    BEQ     1$
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
032442  012742  000756            MOV     #756,-(R2)    ;MOVE TO MAILBOX #  *******  756  *******
032446  005242                    INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
032450  000000                    HALT                  ;INCORRECT STATUS
032452  012706  001000    1$:     MOV     #BUFF,SP      ;SET UP
032456  012767  032476  145334    MOV     #RETE2,RTRAP2 ;SET UP
032464  012767  000357  145304    MOV     #357,CC       ;SET PRIORITY
032472  000277                    SCC                   ;SET CC
032474  000004                    IOT                   ;TRAP
032476  026727  146274  000357    RETE2:  CMP     BUFF-2,#357   ;COMPARES STATUS ON STACK
032504  001404                    BEQ     TS323
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 740           <====
032506  012742  000757            MOV     #757,-(R2)    ;MOVE TO MAILBOX #  *******  757  *******
032512  005242                    INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
032514  000000                    HALT                  ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
                                                        ;  OR SEQUENCE ERROR
                        ;***************************************************************************
                        ;TEST 323       TEST THAT "NEW" STATUS IS CORRECT
                        ;***************************************************************************
032516  005212          TS323:  INC     (R2)          ;UPDATE TEST NUMBER
032520  022712  000323            CMP     #323,(R2)     ;SEQUENCE ERROR?
032524  001110                    BNE     BR46          ;BR TO ERROR HALT ON SEQ ERROR
032526  012706  001000            MOV     #BUFF,SP
032532  012767  032546  145260    MOV     #RETF2,RTRAP2
032540  005067  145256            CLR     RTRAP2+2      ;CLEAR FUTURE PRIORITY AND CC
032544  000004                    IOT
032546                  RETF2:                        ;TEST FOR "C" CLEARED
032546  100004                    BPL     1$
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
9523
9524  032550  012742  000760              MOV     #760,-(R2)      ;MOVE TO MAILBOX #  *******  760  *******
9525  032554  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9526  032556  000000                      HALT                    ;N NOT CLEARED
9527  032560                          1$:
9528  032560  001004                      BNE     2$
9529                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9530                                                            ;           CONDITIONAL BRANCH INST. AND   <====
9531                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9532                                                            ;           WHICH FOLLOWS W/ 761           <====
9533  032562  012742  000761              MOV     #761,-(R2)      ;MOVE TO MAILBOX #  *******  761  *******
9534  032566  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9535  032570  000000                      HALT                    ;Z NOT CLEARED
9536  032572                          2$:
9537  032572  102004                      BVC     3$
9538                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9539                                                            ;           CONDITIONAL BRANCH INST. AND   <====
9540                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9541                                                            ;           WHICH FOLLOWS W/ 754           <====
9542  032574  012742  000762              MOV     #762,-(R2)      ;MOVE TO MAILBOX #  *******  762  *******
9543  032600  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9544  032602  000000                      HALT                    ;V NOT CLEARED
9545  032604                          3$:
9546  032604  103004                      BCC     4$
9547                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9548                                                            ;           CONDITIONAL BRANCH INST. AND   <====
9549                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9550                                                            ;           WHICH FOLLOWS W/ 747           <====
9551  032606  012742  000763              MOV     #763,-(R2)      ;MOVE TO MAILBOX #  *******  763  *******
9552  032612  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9553  032614  000000                      HALT                    ;C NOT CLEARED
9554  032616  032767  000340  145152  4$: BIT     #340,CC         ;TEST PRIORITY
9555  032624  001404                      BEQ     5$
9556                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9557                                                            ;           CONDITIONAL BRANCH INST. AND   <====
9558                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9559                                                            ;           WHICH FOLLOWS W/ 737           <====
9560  032626  012742  000764              MOV     #764,-(R2)      ;MOVE TO MAILBOX #  *******  764  *******
9561  032632  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9562  032634  000000                      HALT                    ;PRIORITY NOT ZERO
9563  032636  012706  001000          5$: MOV     #BUFF,SP
9564  032642  012767  032660  145150      MOV     #RETG2,RTRAP2
9565  032650  012767  000357  145144      MOV     #357,RTRAP2+2   ;SET NEW "CC" AND PRIORITY
9566  032656  000004                      IOT                     ;TRAP HERE
9567  032660                      RETG2:
9568  032660  100404                      BMI     1$
9569                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9570                                                            ;           CONDITIONAL BRANCH INST. AND   <====
9571                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
9572                                                            ;           WHICH FOLLOWS W/ 721           <====
9573  032662  012742  000765              MOV     #765,-(R2)      ;MOVE TO MAILBOX #  *******  765  *******
9574  032666  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9575  032670  000000                      HALT                    ;N NOT SET
9576  032672                          1$:
9577  032672  001404                      BEQ     2$
9578                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
9579                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9580                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9581                                                    ;           WHICH FOLLOWS W/ 714           <====
9582  032674  012742  000766          MOV   #766,-(R2)  ;MOVE TO MAILBOX #  *******  766  *******
9583  032700  005242                  INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9584  032702  000000                  HALT              ;Z NOT SET
9585  032704                  2$:
9586  032704  102404                  BVS   3$
9587                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9588                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9589                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9590                                                    ;           WHICH FOLLOWS W/ 707           <====
9591  032706  012742  000767          MOV   #767,-(R2)  ;MOVE TO MAILBOX #  *******  767  *******
9592  032712  005242                  INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9593  032714  000000                  HALT              ;V NOT SET
9594  032716                  3$:
9595  032716  103404                  BCS   4$
9596                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9597                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9598                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9599                                                    ;           WHICH FOLLOWS W/ 702           <====
9600  032720  012742  000770          MOV   #770,-(R2)  ;MOVE TO MAILBOX #  *******  770  *******
9601  032724  005242                  INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9602  032726  000000                  HALT              ;C NOT SET
9603  032730  016706  145042  4$:     MOV   CC,SP
9604  032734  042706  000017          BIC   #17,SP
9605  032740  022706  000340          CMP   #340,SP
9606  032744  001404                  BEQ   BR46A
9607                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9608                                                    ;           CONDITIONAL BRANCH INST. AND   <====
9609                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
9610                                                    ;           WHICH FOLLOWS W/ 667           <====
9611  032746                  BR46:
9612  032746  012742  000771          MOV   #771,-(R2)  ;MOVE TO MAILBOX #  *******  771  *******
9613  032752  005242                  INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9614  032754  000000                  HALT              ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
9615  032756  012767  000022  145034  BR46A: MOV   #22,20      ;.+2
9616  032764  005067  145032          CLR   22          ;HALT
9617                  ;*****************************************************************************
9618                  ;TEST 324       TEST THAT A TRAP OCCURS ON AN EMT  INSTRUCTION
9619                  ;*****************************************************************************
9620  032770  005212          TS324:  INC   (R2)        ;UPDATE TEST NUMBER
9621  032772  022712  000324          CMP   #324,(R2)   ;SEQUENCE ERROR?
9622  032776  001006                  BNE   TS325-10    ;BR TO ERROR HALT ON SEQ ERROR
9623  033000  012706  001000          MOV   #BUFF,SP    ;STACK POINTER SETUP
9624  033004  012767  033024  145016  MOV   #RETA3,RTRAP3  ;RETURN LOCATION
9625  033012  104000                  EMT               ;RESERVE INSTRUCTIOM, SHOULD TRAP
9626  033014  012742  000772          MOV   #772,-(R2)  ;MOVE TO MAILBOX #  *******  772  *******
9627  033020  005242                  INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9628  033022  000000                  HALT              ;EMT DIDN'T TRAP,OR WRONG $TSTNM
9629  033024                  RETA3:
9630                  ;*****************************************************************************
9631                  ;TEST 325       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9632                  ;*****************************************************************************
9633  033024  005212          TS325:  INC   (R2)        ;UPDATE TEST NUMBER
9634  033026  022712  000325          CMP   #325,(R2)   ;SEQUENCE ERROR?
```

```
9635  033032  001011                        BNE     TS326-10        ;BR TO ERROR HALT ON SEQ ERROR
9636  033034  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9637  033040  012767  033050  144762        MOV     #RETB3,RTRAP3   ;RETURN POINTER
9638  033046  104000                        EMT                     ;RESERVED INSTRUCTION
9639  033050  020627  000774        RETB3:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
9640  033054  001404                        BEQ     TS326
9641                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9642                                                                ;           CONDITIONAL BRANCH INST. AND   <====
9643                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
9644                                                                ;           WHICH FOLLOWS W/ 766           <====
9645  033056  012742  000773                MOV     #773,-(R2)      ;MOVE TO MAILBOX #  *******  773  *******
9646  033062  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9647  033064  000000                        HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
9648                                                                ;  OR SEQUENCE ERROR
9649                                ;*****************************************************************************
9650                                ;TEST 326       TEST THAT PROPER P.C. IS SAVED
9651                                ;*****************************************************************************
9652  033066  005212                TS326:  INC     (R2)            ;UPDATE TEST NUMBER
9653  033070  022712  000326                CMP     #326,(R2)       ;SEQUENCE ERROR?
9654  033074  001012                        BNE     TS327-10        ;BR TO ERROR HALT ON SEQ ERROR
9655  033076  012706  001000                MOV     #BUFF,SP        ;STACK POINTER SETUP
9656  033102  012767  033112  144720        MOV     #RETC3,RTRAP3   ;RTURN FROM TRAP POINTER
9657  033110  104000                        EMT                     ;TRAP ON THIS INSTRICTION
9658  033112  022767  033112  145654  RETC3:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
9659  033120  001404                        BEQ     TS327
9660                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9661                                                                ;           CONDITIONAL BRANCH INST. AND   <====
9662                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
9663                                                                ;           WHICH FOLLOWS W/ 765           <====
9664  033122  012742  000774                MOV     #774,-(R2)      ;MOVE TO MAILBOX #  *******  774  *******
9665  033126  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9666  033130  000000                        HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
9667                                                                ;  OR SEQUENCE ERROR
9668                                ;*****************************************************************************
9669                                ;TEST 327       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
9670                                ;*****************************************************************************
9671  033132  005212                TS327:  INC     (R2)            ;UPDATE TEST NUMBER
9672  033134  022712  000327                CMP     #327,(R2)       ;SEQUENCE ERROR?
9673  033140  001037                        BNE     TS330-10        ;BR TO ERROR HALT ON SEQ ERROR
9674  033142  012706  001000                MOV     #BUFF,SP        ;SET UP
9675  033146  012767  033164  144654        MOV     #RETD3,RTRAP3   ;SET UP
9676  033154  005067  144616                CLR     CC              ;CLEAR CC AND PRIORITY
9677  033160  000257                        CCC
9678  033162  104000                        EMT                     ;TRAP
9679  033164  026727  145606  000000  RETD3:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
9680  033172  001404                        BEQ     1$
9681                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9682                                                                ;           CONDITIONAL BRANCH INST. AND   <====
9683                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
9684                                                                ;           WHICH FOLLOWS W/ 762           <====
9685  033174  012742  000775                MOV     #775,-(R2)      ;MOVE TO MAILBOX #  *******  775  *******
9686  033200  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9687  033202  000000                        HALT                    ;INCORRECT STATUS
9688  033204  012706  001000        1$:     MOV     #BUFF,SP        ;SET UP
9689  033210  012767  033230  144612        MOV     #RETE3,RTRAP3   ;SET UP
9690  033216  012767  000357  144552        MOV     #357,CC         ;SET PRIORITY
```

```
 9691  033224  000277                          SCC                     ;SET CC
 9692  033226  104000                          EMT                     ;TRAP
 9693  033230  026727  145542  000357  RETE3:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
 9694  033236  001404                          BEQ     TS330
 9695                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9696                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9697                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9698                                                                  ;           WHICH FOLLOWS W/ 740           <====
 9699  033240  012742  000776                  MOV     #776,-(R2)      ;MOVE TO MAILBOX #  *******  776  *******
 9700  033244  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9701  033246  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
 9702                                                                  ;  OR SEQUENCE ERROR
 9703                                ;*************************************************************************
 9704                                ;TEST 330       TEST THAT "NEW" STATUS IS CORRECT
 9705                                ;*************************************************************************
 9706  033250  005212                TS330:  INC     (R2)            ;UPDATE TEST NUMBER
 9707  033252  022712  000330                  CMP     #330,(R2)       ;SEQUENCE ERROR?
 9708  033256  001106                          BNE     TS331-10        ;BR TO ERROR HALT ON SEQ ERROR
 9709  033260  012706  001000                  MOV     #BUFF,SP
 9710  033264  012767  033300  144536          MOV     #RETF3,RTRAP3
 9711  033272  005067  144534                  CLR     RTRAP3+2        ;CLEAR FUTURE PRIORITY AND CC
 9712  033276  104000                          EMT
 9713  033300                        RETF3:                            ;TEST FOR "C" CLEARED
 9714  033300  100004                          BPL     1$
 9715                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9716                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9717                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9718                                                                  ;           WHICH FOLLOWS W/ 766           <====
 9719  033302  012742  000777                  MOV     #777,-(R2)      ;MOVE TO MAILBOX #  *******  777  *******
 9720  033306  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9721  033310  000000                          HALT                    ;C NOT CLEARED
 9722  033312                        1$:
 9723  033312  001004                          BNE     2$
 9724                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9725                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9726                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9727                                                                  ;           WHICH FOLLOWS W/ 761           <====
 9728  033314  012742  001000                  MOV     #1000,-(R2)     ;MOVE TO MAILBOX #  *******  1000  *******
 9729  033320  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9730  033322  000000                          HALT                    ;Z NOT CLEARED
 9731  033324                        2$:
 9732  033324  102004                          BVC     3$
 9733                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9734                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9735                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9736                                                                  ;           WHICH FOLLOWS W/ 754           <====
 9737  033326  012742  001001                  MOV     #1001,-(R2)     ;MOVE TO MAILBOX #  *******  1001  *******
 9738  033332  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9739  033334  000000                          HALT                    ;V NOT CLEARED
 9740  033336                        3$:
 9741  033336  103004                          BCC     4$
 9742                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9743                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9744                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9745                                                                  ;           WHICH FOLLOWS W/ 747           <====
 9746  033340  012742  001002                  MOV     #1002,-(R2)     ;MOVE TO MAILBOX #  *******  1002  *******
```

```
9747  033344  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9748  033346  000000                          HALT                    ;C NOT CLEARED
9749  033350  032767  000340  144420  4$:     BIT     #340,CC         ;TEST PRIORITY
9750  033356  001404                          BEQ     5$
9751                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9752                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9753                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9754                                                                  ;           WHICH FOLLOWS W/ 737           <====
9755  033360  012742  001003                          MOV     #1003,-(R2)     ;MOVE TO MAILBOX #  *******  1003  *******
9756  033364  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9757  033366  000000                          HALT                    ;PRIORITY NOT ZERO
9758  033370  012706  001000          5$:     MOV     #BUFF,SP
9759  033374  012767  033412  144426          MOV     #RETG3,RTRAP3
9760  033402  012767  000357  144422          MOV     #357,RTRAP3+2   ;SET NEW "CC" AND PRIORITY
9761  033410  104000                          EMT                     ;TRAP HERE
9762  033412                          RETG3:
9763  033412  100404                          BMI     1$
9764                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9765                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9766                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9767                                                                  ;           WHICH FOLLOWS W/ 721           <====
9768  033414  012742  001004                          MOV     #1004,-(R2)     ;MOVE TO MAILBOX #  *******  1004  *******
9769  033420  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9770  033422  000000                          HALT                    ;N NOT SET
9771  033424                          1$:
9772  033424  001404                          BEQ     2$
9773                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9774                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9775                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9776                                                                  ;           WHICH FOLLOWS W/ 714           <====
9777  033426  012742  001005                          MOV     #1005,-(R2)     ;MOVE TO MAILBOX #  *******  1005  *******
9778  033432  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9779  033434  000000                          HALT                    ;Z NOT SET
9780  033436                          2$:
9781  033436  102404                          BVS     3$
9782                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9783                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9784                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9785                                                                  ;           WHICH FOLLOWS W/ 707           <====
9786  033440  012742  001006                          MOV     #1006,-(R2)     ;MOVE TO MAILBOX #  *******  1006  *******
9787  033444  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9788  033446  000000                          HALT                    ;V NOT SET
9789  033450                          3$:
9790  033450  103404                          BCS     4$
9791                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9792                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9793                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9794                                                                  ;           WHICH FOLLOWS W/ 702           <====
9795  033452  012742  001007                          MOV     #1007,-(R2)     ;MOVE TO MAILBOX #  *******  1007  *******
9796  033456  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9797  033460  000000                          HALT                    ;C NOT SET
9798  033462  000257                  4$:     CCC
9799  033464  022767  000340  144304          CMP     #340,CC
9800  033472  001404                          BEQ     TS331
9801                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9802                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
```

```
                                                ;          REPLACE THE MOVE INSTRUCTION   <====
                                                ;          WHICH FOLLOWS W/ 671           <====
033474  012742  001010          MOV     #1010,-(R2)     ;MOVE TO MAILBOX #  *******  1010  *******
033500  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
033502  000000                  HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
                                                        ;  OR SEQUENCE ERROR
;*****************************************************************************************
;TEST 331       TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP
;*****************************************************************************************
033504  005212  TS331:  INC     (R2)            ;UPDATE TEST NUMBER
033506  022712  000331          CMP     #331,(R2)       ;SEQUENCE ERROR?
033512  001011                  BNE     BR47            ;BR TO ERROR HALT ON SEQ ERROR
                                                        ;**** F11 **** ADD +376 TO SHORTEN TEST
033514  012767  104376  000012          MOV     #EMT+376,RB             ;INITIALIZE BASE EMT INSTRUCTION
033522  012767  033546  144300          MOV     #RA,30          ;RETURN FROM TRAP TO RA
033530  012706  001000  RC:     MOV     #BUFF,SP        ;SET UP STACK POINTER
033534  104000          RB:     EMT                     ;TRAP INST. WILL BE MODIFIED TO EMT+377
033536          BR47:
033536  012742  001011          MOV     #1011,-(R2)     ;MOVE TO MAILBOX #  *******  1011  *******
033542  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
033544  000000                  HALT                    ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TSTNM
033546  005267  177762  RA:     INC     RB              ;INCREMENT TRAP INSTRUCTION
033552  022767  104377  177754          CMP     #104377,RB      ;EMT+377 TO EMT?
033560  103363                  BHIS    RC              ;HAVE WE TESTED ALL
                                                        ;YES
033562  012767  000032  144240          MOV     #32,30          ;/.+
033570  005067  144236          CLR     32              ;HALT
;*****************************************************************************************
;TEST 332       TEST THAT A TRAP OCCURES ON AN "TRACE-TRT" INSTRUCTION
;*****************************************************************************************
033574  005212  TS332:  INC     (R2)            ;UPDATE TEST NUMBER
033576  022712  000332          CMP     #332,(R2)       ;SEQUENCE ERROR?
033602  001006                  BNE     TS333-10        ;BR TO ERROR HALT ON SEQ ERROR
033604  012706  001000          MOV     #BUFF,SP        ;STACK POINTER SETUP
033610  012767  033630  144176          MOV     #RETA4,RTRAP4   ;RETURN LOCATION
033616  000003                  TRT                     ;RESERVED INSTRUCTION, SHOULD TRAP
033620  012742  001012          MOV     #1012,-(R2)     ;MOVE TO MAILBOX #  *******  1012  *******
033624  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
033626  000000                  HALT                    ;TRT DIDN'T TRAP,OR WRONG $TSTNM
033630          RETA4:
;*****************************************************************************************
;TEST 333       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
;*****************************************************************************************
033630  005212  TS333:  INC     (R2)            ;UPDATE TEST NUMBER
033632  022712  000333          CMP     #333,(R2)       ;SEQUENCE ERROR?
033636  001011                  BNE     TS334-10        ;BR TO ERROR HALT ON SEQ ERROR
033640  012706  001000          MOV     #BUFF,SP        ;STACK POINTER SETUP
033644  012767  033654  144142          MOV     #RETB4,RTRAP4   ;RETURN POINTER
033652  000003                  TRT                     ;RESERVED INSTRUCTION
033654  020627  000774  RETB4:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
033660  001404                  BEQ     TS334
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
033662  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
```

```
9915  034052  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9916                                                                  ;  OR SEQUENCE ERROR
9917                                  ;*********************************************************************************
9918                                  ;TEST 336       TEST THAT "NEW" STATUS IS CORRECT
9919                                  ;*********************************************************************************
9920  034054  005212                  TS336:  INC     (R2)            ;UPDATE TEST NUMBER
9921  034056  022712  000336                  CMP     #336,(R2)       ;SEQUENCE ERROR?
9922  034062  001110                          BNE     BR51            ;BR TO ERROR HALT ON SEQ ERROR
9923  034064  012706  001000                  MOV     #BUFF,SP
9924  034070  012767  034104  143716          MOV     #RETF4,RTRAP4
9925  034076  005067  143714                  CLR     RTRAP4+2        ;CLEAR FUTURE PRIORITY AND CC
9926  034102  000003                          TRT
9927  034104                          RETF4:                          ;TEST FOR "C" CLEARED
9928  034104  100004                          BPL     1$
9929                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9930                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9931                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9932                                                                  ;           WHICH FOLLOWS W/ 766           <====
9933  034106  012742  001017                  MOV     #1017,-(R2)     ;MOVE TO MAILBOX #  *******  1017  *******
9934  034112  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9935  034114  000000                          HALT                    ;C NOT CLEARED
9936  034116                          1$:
9937  034116  001004                          BNE     2$
9938                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9939                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9940                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9941                                                                  ;           WHICH FOLLOWS W/ 761           <====
9942  034120  012742  001020                  MOV     #1020,-(R2)     ;MOVE TO MAILBOX #  *******  1020  *******
9943  034124  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9944  034126  000000                          HALT                    ;Z NOT CLEARED
9945  034130                          2$:
9946  034130  102004                          BVC     3$
9947                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9948                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9949                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9950                                                                  ;           WHICH FOLLOWS W/ 754           <====
9951  034132  012742  001021                  MOV     #1021,-(R2)     ;MOVE TO MAILBOX #  *******  1021  *******
9952  034136  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9953  034140  000000                          HALT                    ;V NOT CLEARED
9954  034142                          3$:
9955  034142  103004                          BCC     4$
9956                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9957                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9958                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9959                                                                  ;           WHICH FOLLOWS W/ 747           <====
9960  034144  012742  001022                  MOV     #1022,-(R2)     ;MOVE TO MAILBOX #  *******  1022  *******
9961  034150  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
9962  034152  000000                          HALT                    ;C NOT CLEARED
9963  034154  032767  000340  143614  4$:     BIT     #340,CC         ;TEST PRIORITY
9964  034162  001404                          BEQ     5$
9965                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
9966                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
9967                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
9968                                                                  ;           WHICH FOLLOWS W/ 737           <====
9969  034164  012742  001023                  MOV     #1023,-(R2)     ;MOVE TO MAILBOX #  *******  1023  *******
9970  034170  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
 9971  034172  000000                          HALT                    ;PRIORITY NOT ZERO
 9972  034174  012706  001000          5$:     MOV     #BUFF,SP
 9973  034200  012767  034216  143606          MOV     #RETG4,RTRAP4
 9974  034206  012767  000357  143602          MOV     #357,RTRAP4+2   ;SET NEW "CC" AND PRIORITY
 9975  034214  000003                          TRT                     ;TRAP HERE
 9976  034216                          RETG4:
 9977  034216  100404                          BMI     1$
 9978                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9979                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9980                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9981                                                                  ;           WHICH FOLLOWS W/ 721           <====
 9982  034220  012742  001024                MOV     #1024,-(R2)     ;MOVE TO MAILBOX #  *******  1024  *******
 9983  034224  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9984  034226  000000                          HALT                    ;N NOT SET
 9985  034230                          1$:
 9986  034230  001404                          BEQ     2$
 9987                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9988                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9989                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9990                                                                  ;           WHICH FOLLOWS W/ 714           <====
 9991  034232  012742  001025                MOV     #1025,-(R2)     ;MOVE TO MAILBOX #  *******  1025  *******
 9992  034236  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 9993  034240  000000                          HALT                    ;Z NOT SET
 9994  034242                          2$:
 9995  034242  102404                          BVS     3$
 9996                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 9997                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 9998                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 9999                                                                  ;           WHICH FOLLOWS W/ 707           <====
10000  034244  012742  001026                MOV     #1026,-(R2)     ;MOVE TO MAILBOX #  *******  1026  *******
10001  034250  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10002  034252  000000                          HALT                    ;V NOT SET
10003  034254                          3$:
10004  034254  103404                          BCS     4$
10005                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10006                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10007                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10008                                                                  ;           WHICH FOLLOWS W/ 702           <====
10009  034256  012742  001027                MOV     #1027,-(R2)     ;MOVE TO MAILBOX #  *******  1027  *******
10010  034262  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10011  034264  000000                          HALT                    ;C NOT SET
10012  034266  016706  143504          4$:     MOV     CC,SP
10013  034272  042706  000017                  BIC     #17,SP
10014  034276  022706  000340                  CMP     #340,SP
10015  034302  001404                          BEQ     BR51A
10016                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10017                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10018                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10019                                                                  ;           WHICH FOLLOWS W/ 667           <====
10020  034304                          BR51:
10021  034304  012742  001030                MOV     #1030,-(R2)     ;MOVE TO MAILBOX #  *******  1030  *******
10022  034310  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10023  034312  000000                          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10024  034314  012767  000016  143472  BR51A:  MOV     #16,14
10025  034322  005067  143470                  CLR     16
10026
```

```
10027                                   ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
10028                                   ;ALL INSTRUCTIONS THAT ARE RESERVED
10029                                   ;SHOULD TRAP TO LOCATION 4, AND THE
10030                                   ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
10031                                   ;SHOULD BE PLACED ON THE STACK
10032
10033                                   ;*****************************************************************************
10034                                   ;TEST 337       TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION
10035                                   ;*****************************************************************************
10036  034326  005212                   TS337:  INC     (R2)            ;UPDATE TEST NUMBER
10037  034330  022712  000337                   CMP     #337,(R2)       ;SEQUENCE ERROR?
10038  034334  001006                           BNE     TS340-10        ;BR TO ERROR HALT ON SEQ ERROR
10039  034336  012706  001000                   MOV     #BUFF,SP        ;STACK POINTER SETUP
10040  034342  012767  034362  143434           MOV     #RETA5,RTRAP5   ;RETURN LOCATION
10041  034350  000100                           JMP     %0              ;ILLEGAL INSTRUCTION, SHOULD TRAP
10042  034352  012742  001031                   MOV     #1031,-(R2)     ;MOVE TO MAILBOX #  *******  1031  *******
10043  034356  005242                           INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10044  034360  000000                           HALT                    ;ILLEGAL INSTRUCTION DIDN'T TRAP,OR WRONG $TSTNM
10045  034362                           RETA5:
10046                                   ;*****************************************************************************
10047                                   ;TEST 340       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10048                                   ;*****************************************************************************
10049  034362  005212                   TS340:  INC     (R2)            ;UPDATE TEST NUMBER
10050  034364  022712  000340                   CMP     #340,(R2)       ;SEQUENCE ERROR?
10051  034370  001011                           BNE     TS341-10        ;BR TO ERROR HALT ON SEQ ERROR
10052  034372  012706  001000                   MOV     #BUFF,SP        ;STACK POINTER SETUP
10053  034376  012767  034406  143400           MOV     #RETB5,RTRAP5   ;RETURN POINTER
10054  034404  000100                           JMP     %0              ;RESERVED INSTRUCTION
10055  034406  020627  000774           RETB5:  CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
10056  034412  001404                           BEQ     TS341
10057                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10058                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10059                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10060                                                                   ;           WHICH FOLLOWS W/ 766           <====
10061  034414  012742  001032                           MOV     #1032,-(R2)     ;MOVE TO MAILBOX #  *******  1032  *******
10062  034420  005242                                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10063  034422  000000                                   HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
10064                                                                           ;  OR SEQUENCE ERROR
10065                                   ;*****************************************************************************
10066                                   ;TEST 341       TEST THAT PROPER P.C. IS SAVED
10067                                   ;*****************************************************************************
10068  034424  005212                   TS341:  INC     (R2)            ;UPDATE TEST NUMBER
10069  034426  022712  000341                   CMP     #341,(R2)       ;SEQUENCE ERROR?
10070  034432  001012                           BNE     TS342-10        ;BR TO ERROR HALT ON SEQ ERROR
10071  034434  012706  001000                   MOV     #BUFF,SP        ;STACK POINTER SETUP
10072  034440  012767  034450  143336           MOV     #RETC5,RTRAP5   ;RETURN FROM TRAP POINTER
10073  034446  000100                           JMP     %0              ;TRAP ON THIS INSTRUCTION
10074  034450  022767  034450  144316   RETC5:  CMP     #.,BUFF-4       ;CHECK FOR INCREMENTED P.C.
10075  034456  001404                           BEQ     TS342
10076                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10077                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10078                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10079                                                                   ;           WHICH FOLLOWS W/ 765           <====
10080  034460  012742  001033                           MOV     #1033,-(R2)     ;MOVE TO MAILBOX #  *******  1033  *******
10081  034464  005242                                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10082  034466  000000                                   HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
```

```
10083                                                   ;  OR SEQUENCE ERROR
10084                                   ;*****************************************************************************
10085                                   ;TEST 342       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
10086                                   ;*****************************************************************************
10087  034470  005212                  TS342:  INC     (R2)            ;UPDATE TEST NUMBER
10088  034472  022712  000342                  CMP     #342,(R2)       ;SEQUENCE ERROR?
10089  034476  001037                          BNE     TS343-10        ;BR TO ERROR HALT ON SEQ ERROR
10090  034500  012706  001000                  MOV     #BUFF,SP        ;SET UP
10091  034504  012767  034522  143272          MOV     #RETD5,RTRAP5   ;SET UP
10092  034512  005067  143260                  CLR     CC              ;CLEAR CC AND PRIORITY
10093  034516  000257                          CCC
10094  034520  000100                          JMP     %0              ;TRAP
10095  034522  026727  144250  000000  RETD5:  CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
10096  034530  001404                          BEQ     1$
10097                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10098                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10099                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10100                                                                   ;           WHICH FOLLOWS W/ 762           <====
10101  034532  012742  001034                          MOV     #1034,-(R2)     ;MOVE TO MAILBOX #  *******  1034  *******
10102  034536  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10103  034540  000000                          HALT                    ;INCORRECT STATUS
10104  034542  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
10105  034546  012767  034566  143230          MOV     #RETE5,RTRAP5   ;SET UP
10106  034554  012767  000357  143214          MOV     #357,CC         ;SET PRIORITY
10107  034562  000277                          SCC                     ;SET CC
10108  034564  000100                          JMP     %0              ;TRAP
10109  034566  026727  144204  000357  RETE5:  CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
10110  034574  001404                          BEQ     TS343
10111                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10112                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10113                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10114                                                                   ;           WHICH FOLLOWS W/ 740           <====
10115  034576  012742  001035                          MOV     #1035,-(R2)     ;MOVE TO MAILBOX #  *******  1035  *******
10116  034602  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10117  034604  000000                          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
10118                                                                   ;  OR SEQUENCE ERROR
10119                                   ;*****************************************************************************
10120                                   ;TEST 343       TEST THAT "NEW" STATUS IS CORRECT
10121                                   ;*****************************************************************************
10122  034606  005212                  TS343:  INC     (R2)            ;UPDATE TEST NUMBER
10123  034610  022712  000343                  CMP     #343,(R2)       ;SEQUENCE ERROR?
10124  034614  001106                          BNE     TS344-10        ;BR TO ERROR HALT ON SEQ ERROR
10125  034616  012706  001000                  MOV     #BUFF,SP
10126  034622  012767  034636  143154          MOV     #RETF5,RTRAP5
10127  034630  005067  143152                  CLR     RTRAP5+2        ;CLEAR FUTURE PRIORITY AND CC
10128  034634  000100                          JMP     %0
10129  034636                          RETF5:                          ;TEST FOR "C" CLEARED
10130  034636  100004                          BPL     1$
10131                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10132                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10133                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10134                                                                   ;           WHICH FOLLOWS W/ 766           <====
10135  034640  012742  001036                          MOV     #1036,-(R2)     ;MOVE TO MAILBOX #  *******  1036  *******
10136  034644  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10137  034646  000000                          HALT                    ;C NOT CLEARED
10138  034650                          1$:
```

```
10139  034650  001004                        BNE     2$
10140                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10141                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10142                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10143                                                  ;           WHICH FOLLOWS W/ 761           <====
10144  034652  012742  001037                        MOV     #1037,-(R2)     ;MOVE TO MAILBOX #  *******  1037  *******
10145  034656  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10146  034660  000000                                HALT                    ;Z NOT CLEARED
10147  034662                                2$:
10148  034662  102004                        BVC     3$
10149                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10150                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10151                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10152                                                  ;           WHICH FOLLOWS W/ 754           <====
10153  034664  012742  001040                        MOV     #1040,-(R2)     ;MOVE TO MAILBOX #  *******  1040  *******
10154  034670  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10155  034672  000000                                HALT                    ;V NOT CLEARED
10156  034674                                3$:
10157  034674  103004                        BCC     4$
10158                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10159                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10160                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10161                                                  ;           WHICH FOLLOWS W/ 747           <====
10162  034676  012742  001041                        MOV     #1041,-(R2)     ;MOVE TO MAILBOX #  *******  1041  *******
10163  034702  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10164  034704  000000                                HALT                    ;C NOT CLEARED
10165  034706  032767  000357  143062        4$:     BIT     #357,CC         ;TEST PRIORITY
10166  034714  001404                        BEQ     5$
10167                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10168                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10169                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10170                                                  ;           WHICH FOLLOWS W/ 737           <====
10171  034716  012742  001042                        MOV     #1042,-(R2)     ;MOVE TO MAILBOX #  *******  1042  *******
10172  034722  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10173  034724  000000                                HALT                    ;PRIORITY NOT ZERO
10174  034726  012706  001000                5$:     MOV     #BUFF,SP
10175  034732  012767  034750  143044                MOV     #RETG5,RTRAP5
10176  034740  012767  000357  143040                MOV     #357,RTRAP5+2   ;SET NEW "CC" AND PRIORITY
10177  034746  000100                                JMP     %0              ;TRAP HERE
10178  034750                                RETG5:
10179  034750  100404                        BMI     1$
10180                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10181                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10182                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10183                                                  ;           WHICH FOLLOWS W/ 721           <====
10184  034752  012742  001043                        MOV     #1043,-(R2)     ;MOVE TO MAILBOX #  *******  1043  *******
10185  034756  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10186  034760  000000                                HALT                    ;N NOT SET
10187  034762                                1$:
10188  034762  001404                        BEQ     2$
10189                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10190                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10191                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10192                                                  ;           WHICH FOLLOWS W/ 714           <====
10193  034764  012742  001044                        MOV     #1044,-(R2)     ;MOVE TO MAILBOX #  *******  1044  *******
10194  034770  005242                                INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
10195  034772  000000                          HALT                    ;Z NOT SET
10196  034774                          2$:
10197  034774  102404                          BVS     3$
10198                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10199                                          ;           CONDITIONAL BRANCH INST. AND   <====
10200                                          ;           REPLACE THE MOVE INSTRUCTION   <====
10201                                          ;           WHICH FOLLOWS W/ 707           <====
10202  034776  012742  001045                  MOV     #1045,-(R2)     ;MOVE TO MAILBOX #  *******  1045  *******
10203  035002  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10204  035004  000000                          HALT                    ;V NOT SET
10205  035006                          3$:
10206  035006  103404                          BCS     4$
10207                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10208                                          ;           CONDITIONAL BRANCH INST. AND   <====
10209                                          ;           REPLACE THE MOVE INSTRUCTION   <====
10210                                          ;           WHICH FOLLOWS W/ 702           <====
10211  035010  012742  001046                  MOV     #1046,-(R2)     ;MOVE TO MAILBOX #  *******  1046  *******
10212  035014  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10213  035016  000000                          HALT                    ;C NOT SET
10214  035020  016706  142752          4$:     MOV     CC,SP
10215  035024  022706  000357                  CMP     #357,SP
10216  035030  001404                          BEQ     TS344
10217                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10218                                          ;           CONDITIONAL BRANCH INST. AND   <====
10219                                          ;           REPLACE THE MOVE INSTRUCTION   <====
10220                                          ;           WHICH FOLLOWS W/ 671           <====
10221  035032  012742  001047                  MOV     #1047,-(R2)     ;MOVE TO MAILBOX #  *******  1047  *******
10222  035036  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10223  035040  000000                          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10224                                                                  ;  OR SEQUENCE ERROR
10225                                  ;*****************************************************************************
10226                                  ;TEST 344       TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUCION
10227                                  ;*****************************************************************************
10228  035042  005212                  TS344:  INC     (R2)            ;UPDATE TEST NUMBER
10229  035044  022712  000344                  CMP     #344,(R2)       ;SEQUENCE ERROR?
10230  035050  001006                          BNE     TS345-10        ;BR TO ERROR HALT ON SEQ ERROR
10231  035052  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10232  035056  012767  035076  142720          MOV     #RETH5,RTRAP5   ;RETURN LOCATION
10233  035064  004000                          JSR     %0,%0           ;RESERVED INSTRUCTION, SHOULD TRAP
10234  035066  012742  001050                  MOV     #1050,-(R2)     ;MOVE TO MAILBOX #  *******  1050  *******
10235  035072  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10236  035074  000000                          HALT                    ;DIDN'T TRAP,OR WRONG $TSTNM
10237  035076                          RETH5:
10238                                  ;*****************************************************************************
10239                                  ;TEST 345       TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10240                                  ;*****************************************************************************
10241  035076  005212                  TS345:  INC     (R2)            ;UPDATE TEST NUMBER
10242  035100  022712  000345                  CMP     #345,(R2)       ;SEQUENCE ERROR?
10243  035104  001011                          BNE     TS346-10        ;BR TO ERROR HALT ON SEQ ERROR
10244  035106  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10245  035112  012767  035122  142664          MOV     #RETJ,RTRAP5    ;RETURN POINTER
10246  035120  004000                          JSR     %0,%0           ;RESERVED INSTRUCTION
10247  035122  020627  000774          RETJ:   CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
10248  035126  001404                          BEQ     TS346
10249                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10250                                          ;           CONDITIONAL BRANCH INST. AND   <====
```

```
10251                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10252                                                   ;           WHICH FOLLOWS W/ 766          <====
10253  035130  012742  001051                 MOV     #1051,-(R2)     ;MOVE TO MAILBOX #  *******  1051  *******
10254  035134  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10255  035136  000000                          HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $TSTNM
10256                                                                   ;  OR SEQUENCE ERROR
10257                                          ;*************************************************************************
10258                                          ;TEST 346       TEST THAT PROPER P.C. IS SAVED
10259                                          ;*************************************************************************
10260  035140  005212                 TS346:   INC     (R2)            ;UPDATE TEST NUMBER
10261  035142  022712  000346                  CMP     #346,(R2)       ;SEQUENCE ERROR?
10262  035146  001012                          BNE     TS347-10        ;BR TO ERROR HALT ON SEQ ERROR
10263  035150  012706  001000                  MOV     #BUFF,SP        ;STACK POINTER SETUP
10264  035154  012767  035164  142622          MOV     #RETK,RTRAP5    ;RETURN FROM TRAP POINTER
10265  035162  004000                 INSTK:   JSR     %0,%0           ;TRAP ON THIS INSTRUCTION
10266  035164  022767  035164  143602 RETK:    CMP     #INSTK+2,BUFF-4 ;CHECK FOR INCREMENED P.C.
10267  035172  001404                          BEQ     TS347
10268                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10269                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10270                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10271                                                   ;           WHICH FOLLOWS W/ 765           <====
10272  035174  012742  001052                 MOV     #1052,-(R2)     ;MOVE TO MAILBOX #  *******  1052  *******
10273  035200  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10274  035202  000000                          HALT                    ;INCORRECT P.C.,OR WRONG $TSTNM
10275                                                                   ;  OR SEQUENCE ERROR
10276
10277                                          ;*************************************************************************
10278                                          ;TEST 347       TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
10279                                          ;*************************************************************************
10280  035204  005212                 TS347:   INC     (R2)            ;UPDATE TEST NUMBER
10281  035206  022712  000347                  CMP     #347,(R2)       ;SEQUENCE ERROR?
10282  035212  001037                          BNE     TS350-10        ;BR TO ERROR HALT ON SEQ ERROR
10283  035214  012706  001000                  MOV     #BUFF,SP        ;SET UP
10284  035220  012767  035236  142556          MOV     #RETL,RTRAP5    ;SET UP
10285  035226  005067  142544                  CLR     CC              ;CLEAR CC AND PRIORITY
10286  035232  000257                          CCC
10287  035234  004000                          JSR     %0,%0           ;TRAP
10288  035236  026727  143534  000000 RETL:    CMP     BUFF-2,#0       ;TEST THAT OLD STATUS WENT TO STACK
10289  035244  001404                          BEQ     1$
10290                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10291                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10292                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10293                                                   ;           WHICH FOLLOWS W/ 762           <====
10294  035246  012742  001053                 MOV     #1053,-(R2)     ;MOVE TO MAILBOX #  *******  1053  *******
10295  035252  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10296  035254  000000                          HALT                    ;INCORRCT STATUS
10297  035256  012706  001000          1$:     MOV     #BUFF,SP        ;SET UP
10298  035262  012767  035302  142514          MOV     #RETM,RTRAP5    ;SET UP
10299  035270  012767  000357  142500          MOV     #357,CC         ;SET PRIORITY
10300  035276  000277                          SCC                     ;SET CC
10301  035300  004000                          JSR     %0,%0           ;TRAP
10302  035302  026727  143470  000357 RETM:    CMP     BUFF-2,#357     ;COMPARES STATUS ON STACK
10303  035310  001404                          BEQ     TS350
10304                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10305                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10306                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
10307                                                   ;           WHICH FOLLOWS W/ 740          <====
10308  035312  012742  001054                  MOV     #1054,-(R2)  ;MOVE TO MAILBOX #  *******  1054  *******
10309  035316  005242                          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10310  035320  000000                          HALT                 ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
10311                                                                ;  OR SEQUENCE ERROR
10312          ;*************************************************************************************
10313          ;TEST 350       TEST THAT "NEW" STATUS IS CORRECT
10314          ;*************************************************************************************
10315  035322  005212                  TS350:  INC     (R2)         ;UPDATE TEST NUMBER
10316  035324  022712  000350                  CMP     #350,(R2)    ;SEQUENCE ERROR?
10317  035330  001105                          BNE     TS351-10     ;BR TO ERROR HALT ON SEQ ERROR
10318  035332  012706  001000                  MOV     #BUFF,SP
10319  035336  012767  035352  142440          MOV     #RETN,RTRAP5
10320  035344  005067  142436                  CLR     RTRAP5+2     ;CLEAR FUTURE PRIORITY AND CC
10321  035350  004000                          JSR     %0,%0
10322  035352                          RETN:                        ;TEST FOR "C" CLEARED
10323  035352  100004                          BPL     1$
10324                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10325                                                               ;           CONDITIONAL BRANCH INST. AND   <====
10326                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
10327                                                               ;           WHICH FOLLOWS W/ 766           <====
10328  035354  012742  001055                  MOV     #1055,-(R2)  ;MOVE TO MAILBOX #  *******  1055  *******
10329  035360  005242                          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10330  035362  000000                          HALT                 ;C NOT CLEARED
10331  035364                          1$:
10332  035364  001004                          BNE     2$
10333                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10334                                                               ;           CONDITIONAL BRANCH INST. AND   <====
10335                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
10336                                                               ;           WHICH FOLLOWS W/ 761           <====
10337  035366  012742  001056                  MOV     #1056,-(R2)  ;MOVE TO MAILBOX #  *******  1056  *******
10338  035372  005242                          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10339  035374  000000                          HALT                 ;Z NOT CLEARED
10340  035376                          2$:
10341  035376  102004                          BVC     3$
10342                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10343                                                               ;           CONDITIONAL BRANCH INST. AND   <====
10344                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
10345                                                               ;           WHICH FOLLOWS W/ 754           <====
10346  035400  012742  001057                  MOV     #1057,-(R2)  ;MOVE TO MAILBOX #  *******  1057  *******
10347  035404  005242                          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10348  035406  000000                          HALT                 ;V NOT CLEARED
10349  035410                          3$:
10350  035410  103004                          BCC     4$
10351                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10352                                                               ;           CONDITIONAL BRANCH INST. AND   <====
10353                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
10354                                                               ;           WHICH FOLLOWS W/ 747           <====
10355  035412  012742  001060                  MOV     #1060,-(R2)  ;MOVE TO MAILBOX #  *******  1060  *******
10356  035416  005242                          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10357  035420  000000                          HALT                 ;C NOT CLEARED
10358  035422  016700  142350          4$:     MOV     CC,%0        ;TEMP STORAGE
10359  035426  001404                          BEQ     5$
10360                                                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10361                                                               ;           CONDITIONAL BRANCH INST. AND   <====
10362                                                               ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
10363                                                   ;           WHICH FOLLOWS W/ 740          <====
10364  035430  012742  001061                 MOV     #1061,-(R2)  ;MOVE TO MAILBOX #  *******  1061  *******
10365  035434  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10366  035436  000000                         HALT                 ;PRIORITY NOT ZERO
10367  035440  012706  001000           5$:   MOV     #BUFF,SP
10368  035444  012767  035462  142332       MOV     #RETO,RTRAP5
10369  035452  012767  000357  142326       MOV     #357,RTRAP5+2  ;SET NEW "CC" AND PRIORITY
10370  035460  004000                         JSR     %0,%0          ;TRAP HERE
10371  035462                           RETO:
10372  035462  100404                         BMI     1$
10373                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
10374                                                   ;           CONDITIONAL BRANCH INST. AND  <====
10375                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10376                                                   ;           WHICH FOLLOWS W/ 722          <====
10377  035464  012742  001062                 MOV     #1062,-(R2)  ;MOVE TO MAILBOX #  *******  1062  *******
10378  035470  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10379  035472  000000                         HALT                 ;N NOT SET
10380  035474                           1$:
10381  035474  001404                         BEQ     2$
10382                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
10383                                                   ;           CONDITIONAL BRANCH INST. AND  <====
10384                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10385                                                   ;           WHICH FOLLOWS W/ 715          <====
10386  035476  012742  001063                 MOV     #1063,-(R2)  ;MOVE TO MAILBOX #  *******  1063  *******
10387  035502  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10388  035504  000000                         HALT                 ;Z NOT SET
10389  035506                           2$:
10390  035506  102404                         BVS     3$
10391                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
10392                                                   ;           CONDITIONAL BRANCH INST. AND  <====
10393                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10394                                                   ;           WHICH FOLLOWS W/ 710          <====
10395  035510  012742  001064                 MOV     #1064,-(R2)  ;MOVE TO MAILBOX #  *******  1064  *******
10396  035514  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10397  035516  000000                         HALT                 ;V NOT SET
10398  035520                           3$:
10399  035520  103404                         BCS     4$
10400                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
10401                                                   ;           CONDITIONAL BRANCH INST. AND  <====
10402                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10403                                                   ;           WHICH FOLLOWS W/ 703          <====
10404  035522  012742  001065                 MOV     #1065,-(R2)  ;MOVE TO MAILBOX #  *******  1065  *******
10405  035526  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10406  035530  000000                         HALT                 ;C NOT SET
10407  035532  016700  142240           4$:   MOV     CC,%0
10408  035536  022700  000357                 CMP     #357,%0
10409  035542  001404                         BEQ     TS351
10410                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
10411                                                   ;           CONDITIONAL BRANCH INST. AND  <====
10412                                                   ;           REPLACE THE MOVE INSTRUCTION  <====
10413                                                   ;           WHICH FOLLOWS W/ 672          <====
10414  035544  012742  001066                 MOV     #1066,-(R2)  ;MOVE TO MAILBOX #  *******  1066  *******
10415  035550  005242                         INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
10416  035552  000000                         HALT                 ;PRIORITY WAS CHANGED,OR WRONG $TSTNM
10417                                                   ;  OR SEQUENCE ERROR
10418
```

```
10419                                   ;*****************************************************************************
10420                                   ;TEST 351       TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS
10421                                   ;*****************************************************************************
10422  035554  005212                  TS351:  INC     (R2)            ;UPDATE TEST NUMBER
10423  035556  022712  000351                  CMP     #351,(R2)       ;SEQUENCE ERROR?
10424  035562  001006                          BNE     TS352-10        ;BR TO ERROR HALT ON SEQ ERROR
10425  035564  012706  000150                  MOV     #150,%6         ;R6 = 150
10426  035570  012767  035610  142206        MOV     #TDEC1,4        ;STACK OVERFLOW TRAP POINTER
10427  035576  005746                          TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
10428  035600  012742  001067                MOV     #1067,-(R2)     ;MOVE TO MAILBOX #  *******  1067  *******
10429  035604  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10430  035606  000000                          HALT                    ;SHOULD HAVE TRAPPED,OR WRONG $TSTNM
10431  035610                          TDEC1:
10432
10433                                   ;*****************************************************************************
10434                                   ;TEST 352       TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
10435                                   ;*****************************************************************************
10436  035610  005212                  TS352:  INC     (R2)            ;UPDATE TEST NUMBER
10437  035612  022712  000352                  CMP     #352,(R2)       ;SEQUENCE ERROR?
10438  035616  001011                          BNE     TS353-10        ;BR TO ERROR HALT ON SEQ ERROR
10439  035620  012706  000150                  MOV     #150,%6         ;R6 = 150
10440  035624  012767  035634  142152        MOV     #TDEC2,4        ;TRAP POINTER
10441  035632  005746                          TST     -(6)            ;WITH R6 = 150 SHOULD TRAP
10442  035634  020627  000142          TDEC2:  CMP     %6,#142         ;DID R6 DECREMENT
10443  035640  001404                          BEQ     TS353
10444                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10445                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10446                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10447                                                                  ;           WHICH FOLLOWS W/ 766           <====
10448  035642  012742  001070                MOV     #1070,-(R2)     ;MOVE TO MAILBOX #  *******  1070  *******
10449  035646  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10450  035650  000000                          HALT                    ;R6 NOT = 142,OR WRONG $TSTNM
10451                                                                  ;  OR SEQUENCE ERROR
10452
10453                                   ;*****************************************************************************
10454                                   ;TEST 353       TEST DIFFERENT TYPES OF OVERFLOW
10455                                   ;*****************************************************************************
10456  035652  005212                  TS353:  INC     (R2)            ;UPDATE TEST NUMBER
10457  035654  022712  000353                  CMP     #353,(R2)       ;SEQUENCE ERROR?
10458  035660  001041                          BNE     TS354-10        ;BR TO ERROR HALT ON SEQ ERROR
10459  035662  012706  000150                  MOV     #150,%6
10460  035666  005067  142254                  CLR     146             ;STATUS WORD OF LOC 10
10461  035672  012767  035702  142104        MOV     #TDEC3,4        ;RETURN TO LOC 4
10462  035700  005246                          INC     -(6)
10463  035702  005767  142240          TDEC3:  TST     146
10464  035706  001004                          BNE     1$
10465                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10466                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10467                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10468                                                                  ;           WHICH FOLLOWS W/ 764           <====
10469  035710  012742  001071                MOV     #1071,-(R2)     ;MOVE TO MAILBOX #  *******  1071  *******
10470  035714  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10471  035716  000000                          HALT                    ;INCREMENT OPERATION NOT INHIBITED
10472  035720  012705  001000          1$:     MOV     #1000,%5
10473  035724  012706  000400                  MOV     #400,%6
10474  035730  012767  035750  142046        MOV     #TDEC4,4
```

```
10475  035736  124645                          CMPB    -(6),-(5)
10476  035740  012742  001072                  MOV     #1072,-(R2)     ;MOVE TO MAILBOX #  *******  1072  *******
10477  035744  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10478  035746  000000                          HALT                    ;STACK = 400 AND DECREMENTED, SHOULD TRAP
10479  035750  012706  000400          TDEC4:  MOV     #400,%6
10480  035754  012767  035774  142022        MOV     #TDEC7,4
10481  035762  134546                          BITB    -(5),-(6)
10482  035764                          TDEC6:
10483  035764  012742  001073                  MOV     #1073,-(R2)     ;MOVE TO MAILBOX #  *******  1073  *******
10484  035770  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10485  035772  000000                          HALT                    ;NO STACK OVERFLOW,OR WRONG $TSTNM
10486  035774                          TDEC7:
10487
10488                                   ;*****************************************************************************
10489                                   ;TEST 354       TEST THAT AN 77 CAUSES AN OVERFLOW TRAP
10490                                   ;*****************************************************************************
10491  035774  005212                  TS354:  INC     (R2)            ;UPDATE TEST NUMBER
10492  035776  022712  000354                  CMP     #354,(R2)       ;SEQUENCE ERROR?
10493  036002  001011                          BNE     VDEC2           ;BR TO ERROR HALT ON SEQ ERROR
10494  036004  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10495  036010  012767  036026  141772        MOV     #VDEC2,10       ;SET UP 77 VECTOR
10496  036016  012767  036036  141760        MOV     #VDEC1,4        ;SET UP OVERFLOW VECTOR
10497  036024  000077                          77                      ;THIS TRAP SHOULD CAUSE OVERFLOW
10498  036026                          VDEC2:
10499  036026  012742  001074                  MOV     #1074,-(R2)     ;MOVE TO MAILBOX #  *******  1074  *******
10500  036032  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10501  036034  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10502  036036  012767  000012  141744  VDEC1:  MOV     #10+2,10
10503                                   ;*****************************************************************************
10504                                   ;TEST 355       TEST THAT AN IOT CAUSES AN OVERFLOW TRAP
10505                                   ;*****************************************************************************
10506  036044  005212                  TS355:  INC     (R2)            ;UPDATE TEST NUMBER
10507  036046  022712  000355                  CMP     #355,(R2)       ;SEQUENCE ERROR?
10508  036052  001011                          BNE     VDEC4           ;BR TO ERROR HALT ON SEQ ERROR
10509  036054  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10510  036060  012767  036076  141732        MOV     #VDEC4,20       ;SET UP IOT VECTOR
10511  036066  012767  036106  141710        MOV     #VDEC3,4        ;SET UP OVERFLOW VECTOR
10512  036074  000004                          IOT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10513  036076                          VDEC4:
10514  036076  012742  001075                  MOV     #1075,-(R2)     ;MOVE TO MAILBOX #  *******  1075  *******
10515  036102  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10516  036104  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10517  036106  012767  000022  141704  VDEC3:  MOV     #20+2,20
10518                                   ;*****************************************************************************
10519                                   ;TEST 356       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
10520                                   ;*****************************************************************************
10521  036114  005212                  TS356:  INC     (R2)            ;UPDATE TEST NUMBER
10522  036116  022712  000356                  CMP     #356,(R2)       ;SEQUENCE ERROR?
10523  036122  001011                          BNE     VDEC6           ;BR TO ERROR HALT ON SEQ ERROR
10524  036124  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10525  036130  012767  036146  141672        MOV     #VDEC6,30       ;SET UP EMT VECTOR
10526  036136  012767  036156  141640        MOV     #VDEC5,4        ;SET UP OVERFLOW VECTOR
10527  036144  104000                          EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10528  036146                          VDEC6:
10529  036146  012742  001076                  MOV     #1076,-(R2)     ;MOVE TO MAILBOX #  *******  1076  *******
10530  036152  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
10531  036154  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10532  036156  012767  000032  141644  VDEC5:  MOV     #30+2,30
10533                                  ;*****************************************************************************
10534                                  ;TEST 357       TEST THAT AN TRAP CAUSES AN OVERFLOW TRAP
10535                                  ;*****************************************************************************
10536  036164  005212                  TS357:  INC     (R2)            ;UPDATE TEST NUMBER
10537  036166  022712  000357                  CMP     #357,(R2)       ;SEQUENCE ERROR?
10538  036172  001011                          BNE     VDEC8           ;BR TO ERROR HALT ON SEQ ERROR
10539  036174  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10540  036200  012767  036216  141626          MOV     #VDEC8,34       ;SET UP TRAP VECTOR
10541  036206  012767  036226  141570          MOV     #VDEC7,4        ;SET UP OVERFLOW VECTOR
10542  036214  104400                          TRAP                    ;THIS TRAP SHOULD CAUSE OVERFLOW
10543  036216                          VDEC8:
10544  036216  012742  001077                  MOV     #1077,-(R2)     ;MOVE TO MAILBOX #  *******  1077  *******
10545  036222  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10546  036224  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10547  036226  012767  000036  141600  VDEC7:  MOV     #34+2,34
10548                                  ;*****************************************************************************
10549                                  ;TEST 360       TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
10550                                  ;*****************************************************************************
10551  036234  005212                  TS360:  INC     (R2)            ;UPDATE TEST NUMBER
10552  036236  022712  000360                  CMP     #360,(R2)       ;SEQUENCE ERROR?
10553  036242  001011                          BNE     VDEC10          ;BR TO ERROR HALT ON SEQ ERROR
10554  036244  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10555  036250  012767  036266  141536          MOV     #VDEC10,14      ;SET UP TRT VECTOR
10556  036256  012767  036276  141520          MOV     #VDEC9,4        ;SET UP OVERFLOW VECTOR
10557  036264  000003                          TRT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10558  036266                          VDEC10:
10559  036266  012742  001100                  MOV     #1100,-(R2)     ;MOVE TO MAILBOX #  *******  1100  *******
10560  036272  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10561  036274  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10562  036276  012767  000016  141510  VDEC9:  MOV     #14+2,14
10563                                  ;*****************************************************************************
10564                                  ;TEST 361       TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
10565                                  ;*****************************************************************************
10566  036304  005212                  TS361:  INC     (R2)            ;UPDATE TEST NUMBER
10567  036306  022712  000361                  CMP     #361,(R2)       ;SEQUENCE ERROR?
10568  036312  001011                          BNE     VDEC11          ;BR TO ERROR HALT ON SEQ ERROR
10569  036314  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10570  036320  012767  036336  141456          MOV     #VDEC11,4       ;SET UP ILLA VECTOR
10571  036326  012767  036346  141450          MOV     #VDEC12,4       ;SET UP OVERFLOW VECTOR
10572  036334  004700                          ILLA                    ;THIS TRAP SHOULD CAUSE OVERFLOW
10573  036336                          VDEC11:
10574  036336  012742  001101                  MOV     #1101,-(R2)     ;MOVE TO MAILBOX #  *******  1101  *******
10575  036342  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10576  036344  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10577  036346  012767  000006  141430  VDEC12: MOV     #4+2,4
10578  036354  020627  000370                  CMP     %6,#370         ;STACK PUSHED FOUR WORDS?
10579  036360  001404                          BEQ     TS362
10580                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10581                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10582                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10583                                                                  ;           WHICH FOLLOWS W/ 754           <====
10584  036362  012742  001102                  MOV     #1102,-(R2)     ;MOVE TO MAILBOX #  *******  1102  *******
10585  036366  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10586  036370  000000                          HALT                    ;TRAP OVERFLOW DID NOT OCCUR
```

```
                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 362       TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
                                ;*****************************************************************************
036372  005212                  TS362:  INC     (R2)            ;UPDATE TEST NUMBER
036374  022712  000362                  CMP     #362,(R2)       ;SEQUENCE ERROR?
036400  001011                          BNE     VDEC13          ;BR TO ERROR HALT ON SEQ ERROR
036402  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
036406  012767  036424  141370          MOV     #VDEC13,4       ;SET UP ILLB VECTOR
036414  012767  036434  141362          MOV     #VDEC14,4       ;SET UP OVERFLOW VECTOR
036422  000100                          ILLB                    ;THIS TRAP SHOULD CAUSE OVERFLOW
036424                          VDEC13:
036424  012742  001103                  MOV     #1103,-(R2)     ;MOVE TO MAILBOX #  *******  1103  *******
036430  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
036432  000000                          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
036434  012767  000006  141342  VDEC14: MOV     #4+2,4

                                ;*****************************************************************************
                                ;TEST 363       TEST FOR FALSE OVERFLOW TRAP
                                ;*****************************************************************************
036442  005212                  TS363:  INC     (R2)            ;UPDATE TEST NUMBER
036444  022712  000363                  CMP     #363,(R2)       ;SEQUENCE ERROR?
036450  001023                          BNE     FOVER           ;BR TO ERROR HALT ON SEQ ERROR

036452  012767  036520  141324          MOV     #FOVER,4        ;SET UP OVERFLOW POINTER
036460  012706  001002                  MOV     #1002,%6
036464  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
036466  012706  002002                  MOV     #2002,%6
036472  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
036474  012706  004002                  MOV     #4002,%6
036500  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
036502  012706  010002                  MOV     #10002,%6
036506  005746                          TST     -(6)
036510  012706  020000                  MOV     #20000,%6       ;SHOULD NOT OVERFLOW
036514  005746                          TST     -(6)
036516  000404                          BR      STP

                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 754           <====
036520                          FOVER:
036520  012742  001104                  MOV     #1104,-(R2)     ;MOVE TO MAILBOX #  *******  1104  *******
036524  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
036526  000000                          HALT                    ;IT OVERFLOWED,OR WRONG $TSTNM
```

```
10631  036530  012767  000006  141246  STP:    MOV     #6,4
10632  036536  005067  141244          CLR     6
10633
10634
10635
10636  036542  005212                  TS364:  INC     (R2)            ;UPDATE TEST NUMBER
10637  036544  022712  000364                  CMP     #364,(R2)       ;SEQUENCE ERROR?
10638  036550  001013                          BNE     TS365-10        ;BR TO ERROR HALT ON SEQ ERROR
10639  036552  012706  001000                  MOV     #BUFF,SP
10640  036556  012767  036610  141230          MOV     #RETAT,RTRAP4   ;SET UP TO TRAP TO 14
10641  036564  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10642  036570  012746  036576                  MOV     #.+6,-(SP)      ;PUSH PC
10643  036574  000002                          RTI                     ;SET T BIT
10644  036576  000240                          NOP                     ;TRAP HERE
10645  036600  012742  001105                  MOV     #1105,-(R2)     ;MOVE TO MAILBOX #  *******  1105  *******
10646  036604  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10647  036606  000000                          HALT                    ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10648  036610                          RETAT:
10649
10650
10651
10652  036610  005212                  TS365:  INC     (R2)            ;UPDATE TEST NUMBER
10653  036612  022712  000365                  CMP     #365,(R2)       ;SEQUENCE ERROR?
10654  036616  001022                          BNE     TS366-10        ;BR TO ERROR HALT ON SEQ ERROR
10655  036620  012706  001000                  MOV     #BUFF,SP
10656  036624  012767  036656  141162          MOV     #RETBT,RTRAP4
10657  036632  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10658  036636  012746  036644                  MOV     #.+6,-(SP)      ;PUSH PC
10659  036642  000002                          RTI                     ;SET T BIT
10660  036644  000240                          NOP                     ;TRAP HERE
10661  036646  012742  001106                  MOV     #1106,-(R2)     ;MOVE TO MAILBOX #  *******  1106  *******
10662  036652  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10663  036654  000000                          HALT                    ;TRACE BIT DID NOT TRAP!
10664  036656  020627  000774          RETBT:  CMP     SP,#BUFF-4
10665  036662  001404                          BEQ     TS366
10666                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10667                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10668                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10669                                                                  ;           WHICH FOLLOWS W/ 755           <====
10670  036664  012742  001107                  MOV     #1107,-(R2)     ;MOVE TO MAILBOX #  *******  1107  *******
10671  036670  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10672  036672  000000                          HALT                    ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10673                                                                  ;  OR SEQUENCE ERROR
10674
10675
10676
10677  036674  005212                  TS366:  INC     (R2)            ;UPDATE TEST NUMBER
10678  036676  022712  000366                  CMP     #366,(R2)       ;SEQUENCE ERROR?
10679  036702  001016                          BNE     TS367-10        ;BR TO ERROR HALT ON SEQ ERROR
10680  036704  012706  001000                  MOV     #BUFF,SP
10681  036710  012767  036730  141076          MOV     #RETCT,RTRAP4
10682  036716  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10683  036722  012746  036730                  MOV     #.+6,-(SP)      ;PUSH PC
10684  036726  000002                          RTI                     ;SET T BIT
10685                                                                  ;TRAP HERE
10686  036730  022767  036730  142036  RETCT:  CMP     #.,BUFF-4
```

```
10687  036736  001404                          BEQ     TS367
10688                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10689                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10690                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10691                                                   ;           WHICH FOLLOWS W/ 761           <====
10692  036740  012742  001110                        MOV     #1110,-(R2)     ;MOVE TO MAILBOX #  *******  1110  *******
10693  036744  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10694  036746  000000                          HALT                    ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
10695                                                                   ;  OR SEQUENCE ERROR
10696
10697
10698                                   ;*****************************************************************************************
10699                                   ;TEST 367       TEST  THAT RTT POPS T- BIT
10700                                   ;*****************************************************************************************
10701  036750  005212                  TS367:  INC     (R2)            ;UPDATE TEST NUMBER
10702  036752  022712  000367                  CMP     #367,(R2)       ;SEQUENCE ERROR?
10703  036756  001015                          BNE     TS370-10        ;BR TO ERROR HALT ON SEQ ERROR
10704
10705  036760  012706  001000                  MOV     #BUFF,SP
10706  036764  005001                          CLR     R1              ;CLEAR R1
10707  036766  012746  000020                  MOV     #20,-(SP)
10708  036772  012746  037006                  MOV     #RTT1,-(SP)
10709  036776  012767  037022  141010          MOV     #RTT2,14
10710  037004  000006                          RTT
10711  037006  000240                  RTT1:   NOP
10712  037010  001404                          BEQ     TS370
10713                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10714                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10715                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10716                                                   ;           WHICH FOLLOWS W/ 762           <====
10717  037012  012742  001111                        MOV     #1111,-(R2)     ;MOVE TO MAILBOX #  *******  1111  *******
10718  037016  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10719  037020  000000                          HALT                    ;T-BIT DID NOT TRAP,OR WRONG $TESTN
10720                                                                   ;  OR SEQUENCE ERROR
10721
10722  037022                          RTT2:
10723                                   ;*****************************************************************************************
10724                                   ;TEST 370       TEST  THAT RTT ALLOWS ONE INST. BEFORE TRAP
10725                                   ;*****************************************************************************************
10726  037022  005212                  TS370:  INC     (R2)            ;UPDATE TEST NUMBER
10727  037024  022712  000370                  CMP     #370,(R2)       ;SEQUENCE ERROR?
10728  037030  001030                          BNE     TS371-10        ;BR TO ERROR HALT ON SEQ ERROR
10729  037032  012705  177777                  MOV     #177777,%5
10730  037036  012706  001000          RTT5:   MOV     #BUFF,SP
10731  037042  012746  000020                  MOV     #20,-(SP)
10732  037046  012746  037064                  MOV     #RTT3,-(SP)
10733  037052  012767  037102  140734          MOV     #RTT4,14
10734  037060  005001                          CLR     R1              ;CLEAR R0
10735  037062  000006                          RTT                     ;SET T-BIT
10736  037064  005201                  RTT3:   INC     R1
10737  037066  005205                          INC     %5
10738  037070  001762                          BEQ     RTT5            ;DO THIS TEST NO MORE THAN 2 TIMES
10739  037072  012742  001112                  MOV     #1112,-(R2)     ;MOVE TO MAILBOX #  *******  1112  *******
10740  037076  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10741  037100  000000                          HALT                    ;DID NOT TRAP
10742  037102  005301                  RTT4:   DEC     R1              ;SEE IF RTT ALLOWS 1 INST.
```

```
10743  037104  001406                          BEQ     RTT6
10744  037106  005205                          INC     %5              ;DO THIS TEST NO MORE THAN TWO TIMES
10745  037110  001752                          BEQ     RTT5
10746                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10747                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10748                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10749                                                                  ;           WHICH FOLLOWS W/ 747           <====
10750  037112  012742  001113                  MOV     #1113,-(R2)     ;MOVE TO MAILBOX #  *******  1113  *******
10751  037116  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10752  037120  000000                          HALT                    ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
10753  037122                          RTT6:
10754                                  ;*******************************************************************************
10755                                  ;TEST 371       TEST  THAT RTI DOES NOT ALLOW 1 INST.
10756                                  ;*******************************************************************************
10757  037122  005212                  TS371:  INC     (R2)            ;UPDATE TEST NUMBER
10758  037124  022712  000371                  CMP     #371,(R2)       ;SEQUENCE ERROR?
10759  037130  001022                          BNE     TS372-10        ;BR TO ERROR HALT ON SEQ ERROR
10760  037132  012706  001000                  MOV     #BUFF,SP
10761  037136  012746  000020                  MOV     #20,-(SP)
10762  037142  012746  037160                  MOV     #RTI1,-(SP)
10763  037146  012767  037172  140640          MOV     #RTI2,14
10764  037154  005001                          CLR     R1
10765  037156  000002                          RTI                     ;SET T-BIT
10766  037160  005201                  RTI1:   INC     R1              ;RTI SHOULD NOT ALLOW THIS
10767  037162  012742  001114                  MOV     #1114,-(R2)     ;MOVE TO MAILBOX #  *******  1114  *******
10768  037166  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10769  037170  000000                          HALT                    ;T- BIT DID NOT CAUSE TRAP
10770  037172  005701                  RTI2:   TST     R1
10771                                                                  ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
10772  037174  001404                          BEQ     TS372
10773                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10774                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10775                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10776                                                                  ;           WHICH FOLLOWS W/ 755           <====
10777  037176  012742  001115                  MOV     #1115,-(R2)     ;MOVE TO MAILBOX #  *******  1115  *******
10778  037202  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10779  037204  000000                          HALT                    ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
10780                                                                  ;  OR SEQUENCE ERROR
10781
10782                                  ;*******************************************************************************
10783                                  ;TEST 372       TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
10784                                  ;*******************************************************************************
10785  037206  005212                  TS372:  INC     (R2)            ;UPDATE TEST NUMBER
10786  037210  022712  000372                  CMP     #372,(R2)       ;SEQUENCE ERROR?
10787  037214  001026                          BNE     BR70            ;BR TO ERROR HALT ON SEQ ERROR
10788
10789  037216  012706  001000                  MOV     #BUFF,%6
10790  037222  012767  037262  140564          MOV     #TRACE,14       ;TRACE TRAP
10791  037230  005027  000016                  CLR     #16
10792  037234  005027  000022                  CLR     #22
10793  037240  012767  037302  140552          MOV     #TONT1,20       ;IOT TRAP
10794  037246  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10795  037252  012746  037260                  MOV     #.+6,-(SP)      ;PUSH PC
10796  037256  000006                          RTT
10797  037260  000004                          IOT                     ;TRAP, NEW CC HAVE TRACE RESET
10798  037262                          TRACE:
```

```
10799  037262  012742  001116                  MOV     #1116,-(R2)     ;MOVE TO MAILBOX #  *******  1116  *******
10800  037266  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10801  037270  000000                          HALT                    ;TRACE TRAP WAS NOT INHIBITED
10802  037272                          BR70:
10803  037272  012742  001117                  MOV     #1117,-(R2)     ;MOVE TO MAILBOX #  *******  1117  *******
10804  037276  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10805  037300  000000                          HALT                    ;WRONG TSTNM,OR WRONG $TSTNM
10806  037302  012767  000016  140504  TONT1:  MOV     #16,14
10807  037310  012767  000022  140502          MOV     #22,20
10808                                  ;*************************************************************************************
10809                                  ;TEST 373       TEST THAT THE TRACE BIT IS SAVED IN THE STACK
10810                                  ;*************************************************************************************
10811  037316  005212                  TS373:  INC     (R2)            ;UPDATE TEST NUMBER
10812  037320  022712  000373                  CMP     #373,(R2)       ;SEQUENCE ERROR?
10813  037324  001020                          BNE     STP3            ;BR TO ERROR HALT ON SEQ ERROR
10814  037326  012706  001000                  MOV     #BUFF,%6        ;SET UP STACK POINTER
10815  037332  012767  037356  140454          MOV     #TRC1,14        ;TRACE TRAP RETURN
10816  037340  005067  140452                  CLR     16
10817  037344  012746  000020                  MOV     #20,-(SP)       ;SET THE T BIT
10818  037350  012746  037356                  MOV     #TRC1,-(SP)
10819  037354  000002                          RTI
10820  037356  036727  141414  000020  TRC1:   BIT     BUFF-2,#20      ;CHECK FOR T BIT ON STACK
10821  037364  001004                          BNE     STP3D
10822                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10823                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
10824                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
10825                                                                  ;           WHICH FOLLOWS W/ 757           <====
10826  037366                          STP3:
10827  037366  012742  001120                  MOV     #1120,-(R2)     ;MOVE TO MAILBOX #  *******  1120  *******
10828  037372  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10829  037374  000000                          HALT                    ;T BIT NOT SAVED ON THE STACK,OR WRONG $TSTNM
10830  037376  012767  000016  140410  STP3D:  MOV     #16,14
10831
10832                                  ;*************************************************************************
10833                                  ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL
10834                                  ;ADDRESS TRAPS TO LOCATION 4.  THIS WILL RUN ON 30K SYSTEM.  BUT IF
10835                                  ;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED
10836                                  ;AT, SINCE IT MAY HAVE I/O DEVICES.  IF SWR BIT 1=1, THEN THAT AREA IS
10837                                  ;CHECKED.  (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP).  LOC 160000
10838                                  ;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION
10839                                  ;177700 (THE UNIBUS ADDRESS FOR R0 ON OLDER SYSTEMS) IS USED FOR FORCING
10840                                  ;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.
10841                                  ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
10842                                  ;*************************************************************************************
10843                                  ;TEST 374       TEST NON-EXISTENT ADDRESS TRAPS
10844                                  ;*************************************************************************************
10845  037404  005212                  TS374:  INC     (R2)            ;UPDATE TEST NUMBER
10846  037406  022712  000374                  CMP     #374,(R2)       ;SEQUENCE ERROR?
10847  037412  001150                          BNE     TS375-10        ;BR TO ERROR HALT ON SEQ ERROR
10848  037414  042737  010000  037464          BIC     #10000,@#HICORE ;SET HIGHT CORE LIMIT TO 160000
10849  037422  032737  000002  000322          BIT     #2,@#$SWREG     ;CHECK IF BIT 1 IS SET
10850  037430  001403                          BEQ     1$              ;BRANCH IF IT IS, LEAVE LIMIT=160000
10851  037432  052737  010000  037464          BIS     #10000,@#HICORE ;SET UPPER CORE LIMIT TO 30K (170000)
10852  037440  005000                  1$:     CLR     R0              ;
10853  037442  005067  140340                  CLR     6               ;
10854  037446  012767  037554  140330          MOV     #ATRAP,4        ;SET UP ADDRESS TRAP ENTRANCE
```

```
10855  037454  012706  001000              MOV     #BUFF,SP        ;SET STACK POINTER
10856  037460  105720              NOR:    TSTB    (0)+            ;IF OUTSIDE OF CORE, TRAP TO 4
10857  037462  020027                      CMP     R0,(PC)+        ;IS POINTER INSIDE 28K (30K) CORE
10858  037464  160000              HICORE: .WORD   160000          ;MAY BE CHANGED TO 170000 IF 30K
10859  037466  103774                      BLO     NOR             ;TEST THE REST OF CORE
10860  037470  012737  037512  000004      MOV     #ROTRAP,@#4     ;SET UP NEW VECTOR POINTER
10861  037476  105737  177700              TSTB    @#177700        ;SHOULD CAUSE A TRAP
10862  037502                      TRPADR:
10863  037502  012742  001121              MOV     #1121,-(R2)     ;MOVE TO MAILBOX #  *******  1121  *******
10864  037506  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10865  037510  000000                      HALT                    ;SHOULD HAVE TRAPED
10866                              ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
10867  037512  106767  140260    ROTRAP: MFPS    STATUS
10868  037516  005767  140254            TST     STATUS          ;TEST PSW
10869  037522  001404                    BEQ     1$
10870                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10871                                                           ;           CONDITIONAL BRANCH INST. AND   <====
10872                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
10873                                                           ;           WHICH FOLLOWS W/ 733           <====
10874  037524  012742  001122              MOV     #1122,-(R2)     ;MOVE TO MAILBOX #  *******  1122  *******
10875  037530  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10876  037532  000000                      HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
10877  037534  026727  141234  037502  1$: CMP     BUFF-4,#TRPADR  ;TEST OLD PC AT STACK
10878  037542  001453                    BEQ     TRAPB
10879                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10880                                                           ;           CONDITIONAL BRANCH INST. AND   <====
10881                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
10882                                                           ;           WHICH FOLLOWS W/ 723           <====
10883  037544  012742  001123              MOV     #1123,-(R2)     ;MOVE TO MAILBOX #  *******  1123  *******
10884  037550  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10885  037552  000000                      HALT                    ;OLD PC WAS NOT SAVED
10886                              ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
10887  037554  005300              ATRAP:  DEC     R0              ;
10888  037556  010067  000032              MOV     R0,CORH         ;MOVE THE FIRST NXM LOCATION IN CORH
10889                              ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
10890  037562  013700  037464              MOV     @#HICORE,R0     ;SET UP THE HIGHEST MEM LOCATION
10891  037566  005300                      DEC     R0              ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
10892  037570  000402                      BR      NOSUB           ;DON'T SUBTRACT 1K FIRST TIME
10893  037572  162700  001000      CTRAP:  SUB     #1000,R0        ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
10894                                                           ;TO SPEED UP TESTING
10895  037576  012767  037630  140200  NOSUB: MOV     #BTRAP,4        ;SET UP THE VECTOR
10896  037604  012706  001000              MOV     #BUFF,SP
10897  037610  005710                      TST     (R0)            ;DOES THIS MEMORY EXIST?
10898                                                           ;IF NXM, TRAP TO BTRAP
10899  037612  020027              DTRAP1: CMP     R0,(PC)+        ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
10900                                                           ;TRAP TO ATRAP
10901  037614  000000              CORH:   .WORD   0
10902  037616  101425                      BLOS    TRAPB
10903                                                           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10904                                                           ;           CONDITIONAL BRANCH INST. AND   <====
10905                                                           ;           REPLACE THE MOVE INSTRUCTION   <====
10906                                                           ;           WHICH FOLLOWS W/ 675           <====
10907  037620  012742  001124              MOV     #1124,-(R2)     ;MOVE TO MAILBOX #  *******  1124  *******
10908  037624  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10909  037626  000000                      HALT                    ;CONTENTS OF R0 SHOULD BE LESS THAN OR EQUAL  TO CORH
10910                                                           ;IF THIS COMPARISON FAILS IT MEANS
```

```
10911                                                   ;THAT SOME LEGAL ADDRESS TRAPPED, OR
10912                                                   ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
10913  037630  106767  140142          BTRAP:  MFPS    STATUS
10914  037634  005767  140136                  TST     STATUS
10915  037640  001404                          BEQ     1$
10916                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10917                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10918                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10919                                                   ;           WHICH FOLLOWS W/ 664           <====
10920  037642  012742  001125                          MOV     #1125,-(R2)     ;MOVE TO MAILBOX #  *******  1125  *******
10921  037646  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10922  037650  000000                                  HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
10923  037652  026727  141116  037612  1$:     CMP     BUFF-4,#DTRAP1  ;CHECK IF TRAP PC IS OK
10924  037660  001744                          BEQ     CTRAP
10925                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
10926                                                   ;           CONDITIONAL BRANCH INST. AND   <====
10927                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
10928                                                   ;           WHICH FOLLOWS W/ 654           <====
10929  037662                          AUTO1:
10930  037662  012742  001126                          MOV     #1126,-(R2)     ;MOVE TO MAILBOX #  *******  1126  *******
10931  037666  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10932  037670  000000                                  HALT                    ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10933  037672  012767  000006  140104  TRAPB:  MOV     #6,4            ;RESET TRAP CATCHER
10934  037700  005067  140102                  CLR     6               ;RESET TRAP CATCHER
10935
10936
10937                                   ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10938  037704  005067  000020                  CLR     PROFTE
10939  037710  012706  001000                  MOV     #BUFF,SP        ;SET UP THE STACK POINTER
10940  037714  012767  037732  140062          MOV     #DL11W,4        ;SET UP THE TRAP VECTOR
10941  037722  005767  137636                  TST     TPS             ;TEST THE PUNCH STATUS REGISTER
10942  037726  000403                          BR      DL11W1
10943  037730  000000                  PROFTE: 000000
10944  037732  005267  177772          DL11W:  INC     PROFTE          ;INCR IF NO DL11W
10945  037736  012767  000006  140040  DL11W1: MOV     #6,4
10946
10947  037744                          SKP104:
10948                                   ;*****************************************************************************
10949                                   ;TEST 375       TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
10950                                   ;*****************************************************************************
10951  037744  005212                  TS375:  INC     (R2)            ;UPDATE TEST NUMBER
10952  037746  022712  000375                  CMP     #375,(R2)       ;SEQUENCE ERROR?
10953  037752  001037                          BNE     TDEC8           ;BR TO ERROR HALT ON SEQ ERROR
10954  037754  005767  177750                  TST     PROFTE
10955  037760  001047                          BNE     R7TRX
10956  037762  122767  000001  140330          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
10957  037770  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
10958  037772  005767  140310                  TST     $PASS                   ;IS THIS THE FIRST PASS?
10959  037776  001040                          BNE     R7TRX                   ;IF NOT FIRST PASS, SKIP TEST
10960  040000                          2$:
10961  040000  000005                          RESET                           ;
10962  040002  012767  000340  137766          MOV     #340,STATUS     ;LOCK OUT INTERRUPT
10963  040010  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10964  040014  012767  040062  137762          MOV     #TDEC77,4       ;SET UP OVERFLOW TRAP
10965  040022  012767  040052  140034          MOV     #TDEC8,64       ;SET UP INTERRUPT VECTOR
10966  040030  012767  000100  137526          MOV     #100,TTCSR      ;SET INTERRUPT ENABLE
```

```
10967  040036  005067  137734          CLR     STATUS          ;ALLOW INTERRUPT TO OCCUR
10968  040042  012742  001127          MOV     #1127,-(R2)     ;MOVE TO MAILBOX #  *******  1127  *******
10969  040046  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10970  040050  000000                  HALT                    ;NO INTERRUPT OCCURRED
10971  040052                  TDEC8:
10972  040052  012742  001130          MOV     #1130,-(R2)     ;MOVE TO MAILBOX #  *******  1130  *******
10973  040056  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10974  040060  000000                  HALT                    ;OVERFLOW TRAP DID NOT OCCUR OR WRONG $TSTNM
10975  040062  005067  137476  TDEC77: CLR     TTCSR           ;CLEAR INTERRUPT ENABLE
10976  040066  012767  000006  137710  MOV     #6,4
10977  040074  005067  137706          CLR     6
10978  040100                  R7TRX:
10979                          ;******************************************************************************
10980                          ;TEST 376       TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
10981                          ;******************************************************************************
10982  040100  005212          TS376:  INC     (R2)            ;UPDATE TEST NUMBER
10983  040102  022712  000376          CMP     #376,(R2)       ;SEQUENCE ERROR?
10984  040106  001045                  BNE     BR71            ;BR TO ERROR HALT ON SEQ ERROR
10985  040110  005767  177614          TST     PROFTE
10986  040114  001053                  BNE     NODL
10987  040116  122767  000001  140174  CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
10988  040124  001003                  BNE     2$                      ;IF NOT, DO THIS TEST
10989  040126  005767  140154          TST     $PASS                   ;IS THIS THE FIRST PASS?
10990  040132  001044                  BNE     NODL                    ;IF NOT FIRST PASS, SKIP TEST
10991  040134                  2$:
10992  040134  012706  001000          MOV     #BUFF,%6
10993  040140  012767  000340  137630  MOV     #340,STATUS     ;SET TO A HIGH PRIORITY LEVEL
10994  040146  012767  040212  137710  MOV     #TR0,64
10995  040154  012767  000100  137402  MOV     #100,TTCSR      ;INTERRUPT FOR TTY PUNCH/PRINTER
10996  040162  012767  040222  137644  MOV     #BR71,34        ;TRAP VECTOR
10997  040170  012767  040232  137666  MOV     #TR2,64         ;TTY VECTOR
10998  040176  012767  000340  137632  MOV     #340,36         ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
10999  040204  005067  137566          CLR     STATUS          ;SHOULD INTERRUPT AT END OF CLR INST
11000  040210  104400                  TRAP                    ;TTY INTERRUPT SHOULD OVERRIDE TRAP
11001  040212                  TR0:
11002  040212  012742  001131          MOV     #1131,-(R2)     ;MOVE TO MAILBOX #  *******  1131  *******
11003  040216  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11004  040220  000000                  HALT                    ;TTY SHOULDN'T HAVE INTERRUPTED
11005  040222                  BR71:
11006  040222  012742  001132          MOV     #1132,-(R2)     ;MOVE TO MAILBOX #  *******  1132  *******
11007  040226  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11008  040230  000000                  HALT                    ;TRAP OCCURRED FIRST,OR WRONG $TSTNM
11009  040232  005067  137600  TR2:    CLR     36
11010  040236  042767  000100  137320  BIC     #100,TTCSR
11011  040244                  NODL:
11012                          ;******************************************************************************
11013                          ;TEST 377       TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS
11014                          ;******************************************************************************
11015  040244  005212          TS377:  INC     (R2)            ;UPDATE TEST NUMBER
11016  040246  022712  000377          CMP     #377,(R2)       ;SEQUENCE ERROR?
11017  040252  001043                  BNE     TR5             ;BR TO ERROR HALT ON SEQ ERROR
11018  040254  005767  177450          TST     PROFTE
11019  040260  001063                  BNE     NODL1
11020  040262  122767  000001  140030  CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11021  040270  001003                  BNE     2$                      ;IF NOT, DO THIS TEST
11022  040272  005767  140010          TST     $PASS                   ;IS THIS THE FIRST PASS?
```

```
11023  040276  001054                          BNE     NODL1                   ;IF NOT FIRST PASS, SKIP TEST
11024  040300                          2$:
11025  040300  012706  001000                  MOV     #BUFF,%6
11026  040304  012767  000340  137464          MOV     #340,STATUS
11027  040312  012767  000100  137244          MOV     #100,TTCSR
11028  040320  012767  040360  137506          MOV     #TR3,34         ;TRAP
11029  040326  012767  040372  137530          MOV     #TR4,64         ;TTY OUTPUT
11030  040334  012767  000340  137524          MOV     #340,66         ;TTY OUTPUT PRIORITY
11031  040342  012767  040362  137450          MOV     #TR5,20         ;IOT
11032  040350  012767  000340  137444          MOV     #340,22         ;IOT PRIORITY
11033  040356  104400                          TRAP                    ;THE ACT OF TRAPPING LOWER PRIORITY
11034  040360  000004                  TR3:    IOT                     ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
11035  040362                          TR5:
11036  040362  012742  001133                  MOV     #1133,-(R2)     ;MOVE TO MAILBOX #  *******  1133  *******
11037  040366  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11038  040370  000000                          HALT                    ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TSTNM
11039  040372  005067  137424          TR4:    CLR     22              ;CLR IOT PRIORITY
11040  040376  005067  137464                  CLR     66
11041  040402  012767  000036  137424          MOV     #36,34
11042  040410  012767  000066  137446          MOV     #66,64
11043  040416  012767  000022  137374          MOV     #22,20
11044  040424  005067  137134                  CLR     TTCSR
11045  040430                          NODL1:
11046
11047                                   ;*****************************************************************************
11048                                   ;TEST 400       TEST THAT "RESET" GOES TO OUTSIDE WORLD
11049                                   ;*****************************************************************************
11050  040430  005212                  TS400:  INC     (R2)            ;UPDATE TEST NUMBER
11051  040432  022712  000400                  CMP     #400,(R2)       ;SEQUENCE ERROR?
11052  040436  001035                          BNE     TS401-10        ;BR TO ERROR HALT ON SEQ ERROR
11053  040440  005767  177264                  TST     PROFTE
11054  040444  001036                          BNE     NODL2
11055  040446  122767  000001  137644          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11056  040454  001003                          BNE     2$                      ;IF NOT, DO THIS TEST
11057  040456  005767  137624                  TST     $PASS                   ;IS THIS THE FIRST PASS?
11058  040462  001027                          BNE     NODL2                   ;IF NOT FIRST PASS, SKIP TEST
11059  040464                          2$:
11060  040464  012767  000100  137072          MOV     #100,TTCSR      ;SET INTERRUPT ENABLE
11061  040472  012767  000100  137060          MOV     #100,TRCSR      ;SET INTERRUPT ENABLE
11062  040500  000005                          RESET                   ;SHOULD CLEAR INTERRUPT ENABLE
11063  040502  032767  000100  137054          BIT     #100,TTCSR      ;TEST FOR CLEAR
11064  040510  001404                          BEQ     1$
11065                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11066                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11067                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11068                                                                   ;           WHICH FOLLOWS W/ 752           <====
11069  040512  012742  001134                  MOV     #1134,-(R2)     ;MOVE TO MAILBOX #  *******  1134  *******
11070  040516  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11071  040520  000000                          HALT                    ;RESET FAILED TO CLEAR TTCSR
11072  040522  032767  000100  137030  1$:     BIT     #100,TRCSR      ;TEST FOR CLEAR
11073  040530  001404                          BEQ     TS401
11074                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11075                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11076                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11077                                                                   ;           WHICH FOLLOWS W/ 742           <====
11078  040532  012742  001135                  MOV     #1135,-(R2)     ;MOVE TO MAILBOX #  *******  1135  *******
```

```
11079  040536  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11080  040540  000000                          HALT                    ;RESET FAILED TO CLEAR TRCSR,OR WRONG $TSTNM
11081                                                                   ;  OR SEQUENCE ERROR
11082  040542                                  NODL2:
11083          ;*****************************************************************************************
11084          ;TEST 401       TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
11085          ;*****************************************************************************************
11086  040542  005212                          TS401:  INC     (R2)            ;UPDATE TEST NUMBER
11087  040544  022712  000401                         CMP     #401,(R2)       ;SEQUENCE ERROR?
11088  040550  001023                                 BNE     RESET3          ;BR TO ERROR HALT ON SEQ ERROR
11089  040552  122767  000001  137540                 CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11090  040560  001003                                 BNE     2$                      ;IF NOT, DO THIS TEST
11091  040562  005767  137520                         TST     $PASS                   ;IS THIS THE FIRST PASS?
11092  040566  001027                                 BNE     SKTST2                  ;IF NOT FIRST PASS, SKIP TEST
11093  040570                                  2$:
11094  040570  012706  001000                         MOV     #BUFF,%6        ;SET STACK
11095  040574  012767  040630  137212                 MOV     #RESET2,14      ;SET UP TRACE VECTOR
11096  040602  012746  000020                         MOV     #20,-(R6)       ;SET THE T-BIT ON STACK
11097  040606  012746  040614                         MOV     #1$,-(R6)       ;MOVE NEW PC ON STACK
11098  040612  000006                                 RTT
11099  040614  000005                          1$:    RESET                   ;SHOULD HAVE NO EFFECT
11100  040616  000005                                 RESET                   ;NO EFFECT
11101  040620                                  RESET3:
11102  040620  012742  001136                         MOV     #1136,-(R2)     ;MOVE TO MAILBOX #  *******  1136  *******
11103  040624  005242                                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11104  040626  000000                                 HALT                    ;TRACE TRAP FAILED,OR WRONG $TSTNM
11105  040630  005067  137142                 RESET2: CLR     STATUS          ;CLEAR TRACK
11106  040634  005067  137156                         CLR     16              ;TRACE STATUS
11107  040640  012767  000016  137146                 MOV     #16,14
11108  040646                                  SKTST2:
11109          ;*****************************************************************************************
11110          ;TEST 402       TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
11111          ;*****************************************************************************************
11112  040646  005212                          TS402:  INC     (R2)            ;UPDATE TEST NUMBER
11113  040650  022712  000402                         CMP     #402,(R2)       ;SEQUENCE ERROR?
11114  040654  001057                                 BNE     TTY11           ;BR TO ERROR HALT ON SEQ ERROR
11115  040656  005767  177046                         TST     PROFTE
11116  040662  001062                                 BNE     NODL3
11117  040664  122767  000001  137426                 CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11118  040672  001003                                 BNE     2$                      ;IF NOT, DO THIS TEST
11119  040674  005767  137406                         TST     $PASS                   ;IS THIS THE FIRST PASS?
11120  040700  001053                                 BNE     NODL3                   ;IF NOT FIRST PASS, SKIP TEST
11121  040702                                  2$:
11122  040702  000005                                 RESET
11123  040704  012706  001000                         MOV     #BUFF,%6        ;SET UP STACK
11124  040710  012767  040734  137146                 MOV     #TTY3,64        ;INTERRUPT VECTOR
11125  040716  005067  137054                         CLR     STATUS          ;DROP PROCESSOR PRIORITY
11126  040722  012767  000357  137136                 MOV     #357,66         ;HIGH PRIORITY ON INTERRUPT
11127  040730  005167  136630                         COM     TTCSR           ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
11128  040734  026727  137036  000357  TTY3:  CMP     STATUS,#357
11129  040742  001404                                 BEQ     1$
11130                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11131                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11132                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11133                                                                   ;           WHICH FOLLOWS W/ 744           <====
11134  040744  012742  001137                         MOV     #1137,-(R2)     ;MOVE TO MAILBOX #  *******  1137  *******
```

```
11135  040750  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11136  040752  000000                          HALT                    ;INTERRUPT DID NOT POP CORRECT STATUS
11137  040754  000005                  1$:     RESET                   ;CLR NTERRUPT ENABLE
11138  040756  012706  001000                  MOV     #BUFF,%6        ;STACK SET UP
11139  040762  012767  041006  137074          MOV     #TTY4,64        ;INTERRUPT VECTOR
11140  040770  005067  137072                  CLR     66              ;CLR NEW STATUS
11141  040774  012767  000157  136774          MOV     #157,STATUS     ;PROCESSOR STATUS
11142  041002  005167  136556                  COM     TTCSR           ;SET INTERRUPT ENABLE
11143  041006  005767  136764          TTY4:   TST     STATUS
11144  041012  001404                          BEQ     TTT37
11145                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11146                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11147                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11148                                                                  ;           WHICH FOLLOWS W/ 720           <====
11149  041014                          TTY11:
11150  041014  012742  001140                  MOV     #1140,-(R2)     ;MOVE TO MAILBOX #  *******  1140  *******
11151  041020  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11152  041022  000000                          HALT                    ;INCORRECT STATUS,OR WRONG $TSTNM
11153  041024  005067  136534          TTT37:  CLR     TTCSR
11154  041030                          NODL3:
11155
11156                                  ;*******************************************************************************
11157                                  ;TEST 403       TEST THE 'WAIT' INSTRUCTION
11158                                  ;*******************************************************************************
11159  041030  005212                  TS403:  INC     (R2)            ;UPDATE TEST NUMBER
11160  041032  022712  000403                  CMP     #403,(R2)       ;SEQUENCE ERROR?
11161  041036  001062                          BNE     STP4            ;BR TO ERROR HALT ON SEQ ERROR
11162  041040  122767  000001  137252          CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
11163  041046  001003                          BNE     1$                      ;IF NOT, DO THIS TEST
11164  041050  005767  137232                  TST     $PASS                   ;IS THIS THE FIRST PASS?
11165  041054  001057                          BNE     STP4E                   ;IF NOT FIRST PASS, SKIP TEST
11166  041056                          1$:
11167  041056  042767  000100  136500          BIC     #100,TPS        ;CLEAR INTERRUPT ENABLE
11168  041064  012706  001000                  MOV     #BUFF,SP        ;SET UP THE STACK
11169  041070  012767  041156  136766          MOV     #WATE,64        ;SET UP THE INTERRUPT VECTOR
11170  041076  005067  136764                  CLR     66
11171  041102  105767  136456          WATE1:  TSTB    TPS             ;WAIT FOR READY
11172  041106  100375                          BPL     WATE1           ;TO BE UP
11173  041110  012767  000015  136450          MOV     #15,TPB         ;DO A CARRIAGE RETURN
11174  041116  105767  136442          WATE2:  TSTB    TPS             ;WAIT FOR READY TO COME UP
11175  041122  100375                          BPL     WATE2
11176  041124  012767  000015  136434          MOV     #15,TPB         ;DO ANOTHER CARRIAGE RETURN
11177  041132  052767  000100  136424          BIS     #100,TPS        ;SET THE INTERRUPT ENABLE
11178  041140  005067  136632                  CLR     STATUS          ;CLEAR THE PSW
11179  041144  000001                  WATE3:  WAIT                    ;WAIT FOR THE INTERRUPT
11180  041146  012742  001141                  MOV     #1141,-(R2)     ;MOVE TO MAILBOX #  *******  1141  *******
11181  041152  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11182  041154  000000                          HALT                    ;WAIT INSTRUCTION DID NOT LOOP
11183  041156  005767  136614          WATE:   TST     STATUS  ;IS THE PSW CORRECT?
11184  041162  001404                          BEQ     1$
11185                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11186                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11187                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11188                                                                  ;           WHICH FOLLOWS W/ 725           <====
11189  041164  012742  001142                  MOV     #1142,-(R2)     ;MOVE TO MAILBOX #  *******  1142  *******
11190  041170  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
11191  041172  000000                          HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
11192  041174  026727  137574  041146  1$:     CMP     BUFF-4,#WATE3+2 ;IS THE OLD PC SAVED
11193  041202  001404                          BEQ     STP4E
11194                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11195                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11196                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11197                                                                   ;           WHICH FOLLOWS W/ 715           <====
11198  041204                                  STP4:
11199  041204  012742  001143                          MOV     #1143,-(R2)     ;MOVE TO MAILBOX #  *******  1143  *******
11200  041210  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11201  041212  000000                                  HALT                    ;OLD PC WAS NOT SAVED OR WRONG $TESTN
11202  041214                                  STP4E:
11203                                          ;*************************************************************************************
11204                                          ;TEST 404       TEST THAT USING REGISTER ADDR (177700) CAUSES TIME OUT.
11205                                          ;*************************************************************************************
11206  041214  005212                          TS404:  INC     (R2)            ;UPDATE TEST NUMBER
11207  041216  022712  000404                          CMP     #404,(R2)       ;SEQUENCE ERROR?
11208  041222  001017                                  BNE     TS405-10        ;BR TO ERROR HALT ON SEQ ERROR
11209                                                   ;
11210                                                   ;REGISTER ADDRESS (177700-177717) CAUSE TIME OUT WHEN USED
11211                                                   ;AS PROGRAM ADDRESS BY THE CPU.
11212                                                   ;
11213  041224  012706  001000                          MOV      #BUFF,SP       ;SET  STACK POINTER
11214  041230  012737  041252  000004                  MOV      #RETR1,@#RTRAP5;SET TRAP RETURN ADDR
11215  041236  005737  177700                  PCN1:   TST      @#177700       ;BAD ADDR REFERENCE, TRAP TO 4
11216  041242  012742  001144                          MOV     #1144,-(R2)     ;MOVE TO MAILBOX #  *******  1144  *******
11217  041246  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11218  041250  000000                                  HALT                    ;REFERENCING 177700 DID NOT CAUSE TIME OUT
11219  041252  022767  041242  137514  RETR1:  CMP     #PCN1+4,BUFF-4  ;PROPER PC STORED ON STACK?
11220  041260  001404                          BEQ     TS405
11221                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11222                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11223                                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11224                                                                   ;           WHICH FOLLOWS W/ 760           <====
11225  041262  012742  001145                          MOV     #1145,-(R2)     ;MOVE TO MAILBOX #  *******  1145  *******
11226  041266  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11227  041270  000000                                  HALT                    ;OLD PC WAS NOT SAVED IN STACK
11228                                                                           ;  OR SEQUENCE ERROR
11229
11230                                          ;*************************************************************
11231                                          ;ODD ADDRESS USED BY A 'WORD" INSTRUCTION SHOULD NOT
11232                                          ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
11233                                          ;*************************************************************************************
11234                                          ;TEST 405       TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
11235                                          ;*************************************************************************************
11236  041272  005212                          TS405:  INC     (R2)            ;UPDATE TEST NUMBER
11237  041274  022712  000405                          CMP     #405,(R2)       ;SEQUENCE ERROR?
11238  041300  001013                                  BNE     TS406-10        ;BR TO ERROR HALT ON SEQ ERROR
11239                                                   ;
11240  041302  012737  041330  000004                  MOV     #RETR2,@#RTRAP5 ;SET TRAP RETURN ADDR
11241  041310  005037  000000                          CLR     @#0             ;PUT ALL 0 IN LOC 0
11242  041314  005337  000001                          DEC     @#1             ;DECREMENT ODD ADDRESS, SHOULD NOT TRAP
11243  041320  022737  177777  000000                  CMP     #-1,@#0         ;WORD LOC 0 HAS ALL ONES?
11244  041326  001404                                  BEQ     TS406
11245                                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11246                                                                   ;           CONDITIONAL BRANCH INST. AND   <====
```

```
11247                                                   ;          REPLACE THE MOVE INSTRUCTION   <====
11248                                                   ;          WHICH FOLLOWS W/ 764           <====
11249  041330                                  RETR2:
11250  041330  012742  001146                         MOV     #1146,-(R2)     ;MOVE TO MAILBOX #  *******  1146  *******
11251  041334  005242                                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11252  041336  000000                                 HALT                    ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP
11253                                                                          ;  OR SEQUENCE ERROR
11254
11255                                           ;*************************************************************
11256                                           ;
11257                                           ;USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
11258                                           ;THE REGISTER IN USE WILL BE INCREMENTED.
11259                                           ;
11260                                           ;*************************************************************************************
11261                                           ;TEST 406       TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
11262                                           ;*************************************************************************************
11263  041340  005212                          TS406:  INC     (R2)            ;UPDATE TEST NUMBER
11264  041342  022712  000406                         CMP     #406,(R2)       ;SEQUENCE ERROR?
11265  041346  001016                                 BNE     TS407-10        ;BR TO ERROR HALT ON SEQ ERROR
11266  041350  012737  041376  000004                 MOV     #RETR3,@#RTRAP5 ;SET TRAP RETURN ADDR
11267  041356  012700  177700                         MOV     #177700,R0      ;STORES BAD MEMORY REFERENCE
11268  041362  012720  001234                         MOV     #1234,(R0)+     ;BAD ADDR REFERENCE, TRAP TO LOC 4
11269  041366  012742  001147                         MOV     #1147,-(R2)     ;MOVE TO MAILBOX #  *******  1147  *******
11270  041372  005242                                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11271  041374  000000                                 HALT                    ;ADDRESSING 177700 DID NOT CAUSE TRAP
11272  041376  022700  177702                 RETR3:. CMP     #177702,R0      ;WAS R0 INCREMENTED?
11273  041402  001404                                 BEQ     TS407
11274                                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11275                                                                          ;           CONDITIONAL BRANCH INST. AND   <====
11276                                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
11277                                                                          ;           WHICH FOLLOWS W/ 761           <====
11278  041404  012742  001150                         MOV     #1150,-(R2)     ;MOVE TO MAILBOX #  *******  1150  *******
11279  041410  005242                                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11280  041412  000000                                 HALT                    ;R0 WAS NOT INCREMENTED
11281                                                                          ;  OR SEQUENCE ERROR
11282                                           ;*************************************************************
11283                                           ;
11284                                           ;AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
11285                                           ;TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
11286                                           ;WAS BAD, A BOUBLE BUS ERROR OCCURED.  THE STACK POINTER WOULD
11287                                           ;THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
11288                                           ;LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
11289                                           ;EXECUTION.
11290                                           ;
11291                                           ;*************************************************************************************
11292                                           ;TEST 407       TEST FOR DOUBLE BUS ERROR.
11293                                           ;*************************************************************************************
11294  041414  005212                          TS407:  INC     (R2)            ;UPDATE TEST NUMBER
11295  041416  022712  000407                         CMP     #407,(R2)       ;SEQUENCE ERROR?
11296  041422  001054                                 BNE     TS410-10        ;BR TO ERROR HALT ON SEQ ERROR
11297  041424  012737  041502  000004                 MOV     #DBE1,@#RTRAP5  ;SET TRAP RETURN ADDR
11298  041432  012737  000340  000006                 MOV     #340,@#6        ;SET UP PS
11299  041440  012737  041472  000010                 MOV     #DBE2,@#RTRAP   ;SET TRAP RETURN ADDR
11300  041446  012737  000340  000012                 MOV     #340,@#12       ;SET UP PS
11301  041454  012706  177700                         MOV     #177700,SP      ;SET ILLEGAL SP
11302  041460  000077                          DBE:   TRAPA                   ;ILLEGAL INSTRUCTION
```

```
11303  041462  012742  001151                MOV     #1151,-(R2)     ;MOVE TO MAILBOX #  *******  1151  *******
11304  041466  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11305  041470  000000                        HALT                    ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
11306  041472                        DBE2:
11307  041472  012742  001152                MOV     #1152,-(R2)     ;MOVE TO MAILBOX #  *******  1152  *******
11308  041476  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11309  041500  000000                        HALT                    ;TRAP TO WRONG LOCATION
11310  041502  022737  041462  000000  DBE1: CMP     #DBE+2,@#0      ;OLD PC GOT SAVED?
11311  041510  001404                        BEQ     DBE3
11312                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11313                                                                 ;           CONDITIONAL BRANCH INST. AND   <====
11314                                                                 ;           REPLACE THE MOVE INSTRUCTION   <====
11315                                                                 ;           WHICH FOLLOWS W/ 744           <====
11316  041512  012742  001153                MOV     #1153,-(R2)     ;MOVE TO MAILBOX #  *******  1153  *******
11317  041516  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11318  041520  000000                        HALT                    ;OLD PC DID NOT GET SAVEDD
11319  041522  022737  000340  000002  DBE3: CMP     #340,@#2        ;CORRECT PS SAVED?
11320  041530  001404                        BEQ     DBE4
11321                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11322                                                                 ;           CONDITIONAL BRANCH INST. AND   <====
11323                                                                 ;           REPLACE THE MOVE INSTRUCTION   <====
11324                                                                 ;           WHICH FOLLOWS W/ 734           <====
11325  041532  012742  001154                MOV     #1154,-(R2)     ;MOVE TO MAILBOX #  *******  1154  *******
11326  041536  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11327  041540  000000                        HALT                    ;CORRECT PS DID NOT GET SAVE
11328  041542  022706  000000        DBE4:   CMP     #0,SP           ;SP POINTS TO LOC 0?
11329  041546  001404                        BEQ     DBE5
11330                                                                 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11331                                                                 ;           CONDITIONAL BRANCH INST. AND   <====
11332                                                                 ;           REPLACE THE MOVE INSTRUCTION   <====
11333                                                                 ;           WHICH FOLLOWS W/ 725           <====
11334  041550  012742  001155                MOV     #1155,-(R2)     ;MOVE TO MAILBOX #  *******  1155  *******
11335  041554  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11336  041556  000000                        HALT                    ;SP IS NOT POINTING TO LOC 0
11337  041560  012706  001000        DBE5:   MOV     #STBOT,SP       ;RESET SP
11338
11339
11340                                ;*************************************************************************************
11341                                ;TEST 410       TEST MFPT
11342                                ;*************************************************************************************
11343  041564  005212                TS410:  INC     (R2)            ;UPDATE TEST NUMBER
11344  041566  022712  000410                CMP     #410,(R2)       ;SEQUENCE ERROR?
11345  041572  001023                        BNE     TS411-10        ;BR TO ERROR HALT ON SEQ ERROR
11346                                ;THIS TESTS THE MFPT INSTRUCTION- MOVE FROM PROCESSOR TYPE
11347                                ;UPON EXECUTION, R0 WILL RECEIVE THE PROCESSOR MODEL CODE
11348                                ;WHICH IS "000003" FOR F11.
11349          000007                MFPT=000007
11350  041574  012706  001000                MOV     #STBOT,R6       ;INIT. SP
11351  041600  013746  000010                MOV     @#10,-(SP)      ;SAVE TRAP VECTOR
11352  041604  012737  041634  000010        MOV     #1$,@#10        ;SET UP ILLEGAL INSTRUCTION TRAP
11353  041612  010046                        MOV     R0,-(SP)        ;SAVE R0
11354  041614  000007                        MFPT                    ;GET PROCESSOR MODEL
11355  041616  010037  041644                MOV     R0,@#CPUTYP     ;STORE IT
11356  041622  012600                        MOV     (SP)+,R0        ;RESTORE R0
11357  041624  022737  000003  041644        CMP     #3,@#CPUTYP     ;CHECK MODEL TYPE
11358  041632  001405                        BEQ     XXT
```

```
11359                                                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11360                                                   ;           CONDITIONAL BRANCH INST. AND   <====
11361                                                   ;           REPLACE THE MOVE INSTRUCTION   <====
11362                                                   ;           WHICH FOLLOWS W/ 757           <====
11363  041634                                   1$:
11364  041634  012742  001156                          MOV     #1156,-(R2)     ;MOVE TO MAILBOX #  *******  1156  *******
11365  041640  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11366  041642  000000                                  HALT                    ;ILLEGAL INSTR TRAP OR WRONG MODEL TYPE
11367
11368  041644  000000                           CPUTYP: .WORD   0
11369  041646                                   XXT:
11370  041646  012637  000010                          MOV     (SP)+,@#10      ;RESTORE TRAP VECTOR
11371                                           ;*****************************************************************
11372                                           ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
11373                                           ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
11374                                           ;CONTROL-CHIP.  ON THE KDF11-A, THIS INCLUDES: FIS(CTL3), CIS AND WCS
11375                                           ;INSTRUCTIONS.  A CTLERR TRAPS TO LOCATION 10.
11376                                           ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
11377                                           ;*************************************************************************
11378                                           ;TEST 411       TEST CTLERR SERVICE ROUTINE
11379                                           ;*************************************************************************
11380  041652  005212                           TS411:  INC     (R2)            ;UPDATE TEST NUMBER
11381  041654  022712  000411                          CMP     #411,(R2)       ;SEQUENCE ERROR?
11382  041660  001022                                  BNE     TS412-10        ;BR TO ERROR HALT ON SEQ ERROR
11383
11384  041662  012706  001000                          MOV     #STBOT,R6       ;INIT STACK POINTER
11385  041666  012737  041706  000010                  MOV     #1$,@#10                ;SET UP RETURN ADDR FROM TRAP
11386  041674  012737  000340  000012                  MOV     #340,@#12       ;SET TRAP PRIORITY=7
11387  041702  075006                                  FADD    R6              ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
11388  041704  000000                                  HALT                    ;DID NOT TRAP..CHECK CSEL LINE
11389  041706  012737  041720  000010  1$:             MOV     #2$,@#10                ;SET UP RETURN ADDR FROM TRAP
11390  041714  076000                                  76000                   ;EXECUTE CIS INSTR..SHOULD TRAP
11391  041716  000000                                  HALT                    ;DID NOT TRAP
11392  041720  012737  041732  000010  2$:             MOV     #3$,@#10                ;SET UP RETURN ADDR FROM TRAP
11393  041726  076700                                  76700                   ;EXECUTE WCS INSTR..SHOULD CAUSE CTLERR AND TRAP
11394  041730  000000                                  HALT                    ;DID NOT TRAP
11395  041732  012706  001000          3$:             MOV     #STBOT,R6       ;RE-INIT STACK POINTER
11396
11397                                           ;*************************************************************************
11398                                           ;TEST 412       TEST THAT ALL RESERVED INSTRUCTIONS TRAP
11399                                           ;*************************************************************************
11400  041736  005212                           TS412:  INC     (R2)            ;UPDATE TEST NUMBER
11401  041740  022712  000412                          CMP     #412,(R2)       ;SEQUENCE ERROR?
11402  041744  001112                                  BNE     RET4            ;BR TO ERROR HALT ON SEQ ERROR
11403  041746  042767  000100  135610                  BIC     #100,TPS
11404  041754  012737  042002  000244                  MOV     #TRAP244,@#244  ; SET UP TO SEE IF
11405  041762  013767  000010  000024                  MOV     @#10,TENSAVE    ;  THIS PROCESSOR HAS THE
11406  041770  012737  042012  000010                  MOV     #TRAP10,@#10    ;  FLOATING POINT OPTION
11407  041776  170007                                  .WORD   170007          ; AN ILLEGAL FPP INSTRUCTION
11408  042000  000406                                  BR      AROUND          ; THE FOLLOWING
11409  042002                                   TRAP244:                       ; IF FPP IN--
11410  042002  013767  042234  000234                  MOV     @#FPP,FINISH    ;  RESET END OF TABLE POINTER
11411  042010  000002                                  RTI                     ; AND RETURN
11412  042012                                   TRAP10:                        ; LEAVE THE TABLE ALONE
11413  042012  000002                                  RTI                     ; AND RETURN
11414  042014  000000                           TENSAVE:.WORD   0              ; A PLACE TO STORE CONTENTS OF 10
```

```
11415
11416  042016                                  AROUND:                         ;  CONTINUATION POINT
11417  042016  012737  000246  000244                  MOV     #246,@#244      ;  RESTORE THE TRAP VECTOR
11418  042024  016737  177764  000010                  MOV     TENSAVE,@#10    ; RESTORE THE ILLEGAL INST. VECTOR
11419  042032  012703  042214                          MOV     #TABLE,TAB      ;TABLE POINTER
11420  042036  012305                          GIN1:   MOV     (TAB)+,FIRST    ;FIRST OR CURRENT INSTRUCTION
11421  042040  012301                                  MOV     (TAB)+,LAST     ;LAST INSTRUCTION OR GROUP
11422  042042  020567  000176                          CMP     FIRST,FINISH    ;TESTED ALL
11423  042046  001415                                  BEQ     GIN3            ;YES BRANCH
11424  042050  010567  000172                          MOV     FIRST,INST      ;SET UP INST
11425  042054  005267  000166                  GIN2:   INC     INST
11426  042060  012767  042126  135722                  MOV     #RET,10         ;SET UP RETURN FROM TRAP
11427  042066  012706  001000                          MOV     #BUFF,SP        ;SET UP STACK POINTER
11428  042072  005067  135700                          CLR     CC              ;CLEAR PRIORITY
11429  042076  000167  000144                          JMP     INST            ;EXECUTE RESERVED INSTRUCTION
11430  042102  012737  062350  000034  GIN3:   MOV     #$TRAP,@#34     ;TRAP VECTOR FOR TRAP CALL
11431  042110  012737  000340  000036                  MOV     #340,@#36       ;LEVEL 7
11432  042116  012700  000370                          MOV     #370,R0         ;RESET RESERVED AREA 370-402
11433  042122  000167  000234                          JMP     THRPRT          ;JUMP TO EIS TEST
11434
11435                                          ;TRAPPING SHOULD SEND YOU HERE
11436  042126  020627  000774                  RET:    CMP     SP,#BUFF-4      ;TEST DECREMENT OF SP
11437  042132  001404                                  BEQ     RET1
11438  042134  012742  001157                          MOV     #1157,-(R2)     ;MOVE TO MAILBOX #  *******  1157  *******
11439  042140  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11440  042142  000000                                  HALT                    ;WRONG DECREMENT
11441  042144  026727  136624  042250  RET1:   CMP     BUFF-4,#INST+2  ;LOC OF INST UNINCREMENTED
11442  042152  001404                                  BEQ     RET2
11443  042154  012742  001160                          MOV     #1160,-(R2)     ;MOVE TO MAILBOX #  *******  1160  *******
11444  042160  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11445  042162  000000                                  HALT                    ;INST INC ON TRAP
11446  042164  005767  136606          RET2:   TST     BUFF-2
11447  042170  001404                                  BEQ     RET3
11448                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
11449                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
11450                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
11451                                                                  ;           WHICH FOLLOWS W/ 665           <====
11452  042172                                  RET4:
11453  042172  012742  001161                          MOV     #1161,-(R2)     ;MOVE TO MAILBOX #  *******  1161  *******
11454  042176  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
11455  042200  000000                                  HALT                    ;CONDITION CODES SET ON TRAP OR WRONG $TSTNM
11456  042202  026701  000040          RET3:   CMP     INST,LAST
11457  042206  001713                                  BEQ     GIN1            ;SET UP NEW GROUP
11458  042210  000167  177640                          JMP     GIN2            ;FINISH OLD GROUP
11459                                                                  ;END OF INSTRUCTION GROUP
11460  042214  000007                  TABLE:  7                       ;END OF OPERATE
11461  042216  000077                          77
11462  042220  000207                          207                     ;RTS,RT1,JMP
11463  042222  000227                          227
11464  042224  006777                          6777
11465  042226  007777                          7777
11466  042230  075037                          075037
11467  042232  076777                          76777
11468  042234  167777                  FPP:    167777                  ;  START OF THE FPP INSTRUCTIONS
11469  042236  177700                          177700
11470  042240  177716                          177716
```

```
11471  042242  177777                    177777
11472  042244  042244          FINISH:   .                       ;END FLAG
11473  042246  000000          INST:     HALT                    ;WILL CONTINUE RESERVED INST
11474  042250  000000                    HALT                    ;SHOULD TRAP TO LOC 10
11475  042252  000000                    HALT                    ;LOC 10 SHOULD SEND YOU TO
11476  042254  000000                    HALT                    ;RET
11477  042256  000000                    HALT
11478                            .SBTTL ** STARTING OF EIS TEST **
11479           000000           .REPT  0
11480
11481
11482                            .PAGE
11483
11484                                   PART THREE:             EIS INSTRUCTION TESTS
11485
11486
11487                                   ABSTRACT
11488
11489                                   THIS PROGRAM TESTS THE F11 EXTENDED INSTRUCTION SET
11490                                   <ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT-
11491                                   LEAST ONCE WITH EACH INSTRUCTION.
11492
11493
11494
11495                                   SWITCH SETTINGS
11496
11497                                   IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM
11498                                   AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE
11499                                   SWITCH REGISTER.  THE USER SHOULD SET THIS LOCATION BEFORE
11500                                   STARTING THE PROGRAM.
11501
11502
11503                                   BIT #   OCTAL VALUE             FUNCTION
11504
11505                                   15      100000..... HALT ON ERROR
11506                                   13      020000..... INHIBIT ERROR PRINTOUT
11507
11508                                   AN 8 BIT BYTE $ENVM [I.E. LOCATION 321] HAS BEEN USED TO DEFINE
11509                                   THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING
11510                                   BIT 5 OF BYTE $ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN
11511                                   LOCATION 320
11512
11513                            .ENDR
11514
11515
11516                            .ABS
11517                            .NLIST  MD,MC,CND
11518                            .LIST   ME
11519
11520
```

```
11521
11522
11523                   000000          DUMMY=  0
11524                   000001          ERRNM=  1
11525                   000051          F=      51
11526                   000176          N=      176
11527                   001000          SW09=   1000
11528                   002000          SW10=   2000
11529                   004000          SW11=   4000
11530                   010000          SW12=   10000
11531
11532   042260                          COUNT:
11533                   042262                  .=COUNT+2
11534   042262                          PSWORD:
11535                   042264                  .=PSWORD+2
11536   042264                          TEMP1:
11537                   042266                  .=TEMP1+2
11538   042266                          TEMP2:
11539                   042270                  .=TEMP2+2
11540   042270                          TEMP3:
11541                   042272                  .=TEMP3+2
11542   042272                          TEMP4:
11543                   042274                  .=TEMP4+2
11544   042274  000000                  TEMP5:  .WORD
11545   042276  000000                  TEMP6:  .WORD
11546   042300     000                  TYPCNT: .BYTE
11547   042301     000                  $TPCNT: .BYTE
11548   042302  000007                  S0:     7
11549   042304  177771                  S1:     -7
11550   042306  042304                  S2:     S1
11551   042310  177772                  S3:     -6
11552   042312  177777                  S4:     -1
11553   042314  040000                  S5:     40000
11554   042316  042314                  S6:     S5
11555   042320  040000                  S7:     40000
11556   042322  177776                  S8:     -2
11557   042324  000002                  S9:     2
11558   042326  042324                  S10:    S9
11559   042330  000002                  S11:    2
11560   042332  177570                  SWR:    177570
11561   042334  177570                  DISPLAY:        177570
11562   042336  000064                  TTYOUT: 64
11563   042340  177566                  $TPB:   177566
11564   042342  177564                  $TPS:   177564
11565   042344  005015  020040  000040  $CRLF:  .ASCIZ  <15><12>/   /
11566   042352  006412  047520  042527  POWER:  .ASCIZ  <12><15>/POWER/
11567   042360  000122
11568
11569
11570
11571
11572
11573
11574
11575
11576
```

```
11577
```

```
11578  042362                          THRPRT:
11579
11580  042362  012705  000304                  MOV     #$TESTN,R5      ;MAKE R5 POINT TO THE LOCATION $TESTN
11581  042366  005037  042260                  CLR     @#COUNT         ;CLEAR THE COUNTER
11582  042372  012715  000001                  MOV     #1,(R5)         ;INITIALIZE TEST NUMBER
11583  042376  012706  001000                  MOV     #STBOT,SP       ;** STACK AT STBOT **
11584  042402  013746  000004                  MOV     @#4,-(SP)       ;SAVE ERROR VECTOR
11585  042406  013746  000006                  MOV     @#6,-(SP)       ;
11586  042412  012767  042426  135364          MOV     #1$,4           ;SET UP TIMEOUT VECTOR
11587  042420  005777  177706                  TST     @SWR            ;TRY TO REFERENCE HARDWRE SWR
11588  042424  000407                          BR      3$              ;BRANCH IF NO TIMEOUT TRAP OCCURS
11589  042426  012767  000176  177676  1$:     MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
11590  042434  012767  000174  177672          MOV     #DISPREG,DISPLAY        ;POINT TO SOFTWARE DISPLAY REG
11591  042442  022626                          CMP     (SP)+,(SP)+     ;RESTORE STACK
11592  042444  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR VECTOR
11593  042450  012637  000004                  MOV     (SP)+,@#4       ;
11594  042454  106427  000000                  MTPS    #0              ;PLACE #0 IN PSW
11595  042460  132737  000001  000320          BITB    #1,@#$ENV       ;ARE WE UNDER APT ?
11596  042466  001403                          BEQ     2$              ;IF NOT THEN GO TO 2$
11597  042470  012767  000322  177634          MOV     #$SWREG,SWR     ;USE APT SWITCH REGISTER
11598  042476  012737  000001  042264  2$:     MOV     #1,@#TEMP1      ;TEMP1=1
11599  042504  005037  042266                  CLR     @#TEMP2         ;TEMP2=0
11600  042510  012737  000001  042270          MOV     #1,@#TEMP3      ;TEMP3=1
11601  042516  005037  042272                  CLR     @#TEMP4         ;TEMP4=0
11602
11603
```

```
11604
11605                                   ;*********************************
11606                                   ;     ASH INSTRUCTION TESTS
11607                                   ;*********************************
11608
11609
11610
11611
11612                                           ;*************
11613                                           ; TESTS 1-36
11614                                           ;*************
11615
11616
11617  042522  010767  135444          ASTART: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11618  042526  013700  042264                  MOV     @#TEMP1,%0      ;LOAD R0 WITH THE CONTENTS OF TEMP1
11619  042532  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11620  042540  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11621  042542  013701  042266                  MOV     @#TEMP2,R1      ;OTHERWISE EXECUTE THE INSTRUCTION
11622                                                                  ;IN MODE 0 USING R1
11623  042546  072001                          ASH     R1,R0
11624  042550  000402                          BR      4$
11625  042552  072067  177510          2$:     ASH     TEMP2,%0        ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
11626  042556  106737  042262          4$:     MFPS    @#PSWORD        ;SAVE PS
11627  042562  123737  042272  042262          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11628  042570  001403                          BEQ     .+10
11629  042572  004767  016044                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11630                                                                  ;THE PS IS NOT EQUAL TO 0
11631  042576  000001                          1               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11632                                                          ;BY  (013746  000172  000207)
11633
11634  042600  005237  042260                  INC     @#COUNT         ;INCREMENT THE COUNTER
11635  042604  023700  042270                  CMP     @#TEMP3,%0      ;IS THE RESULT IN R0 EQUAL TO TEMP3?
11636  042610  001403                          BEQ     .+10
11637  042612                          6$:
11638  042612  004767  016024                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11639                                                                  ;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
11640  042616  000002                          2               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11641                                                          ;BY  (013746  000172  000207)
11642
11643  042620  021537  042260                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE
11644                                                                  ;COUNTER?
11645  042624  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11646  042626  005215                          INC     (R5)
11647  042630  010767  135336                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11648  042634  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11649                                                                  ;BY 14. AND RIGHT BY 14.?
11650  042640  002011                          BGE     8$
11651  042642  005237  042266                  INC     @#TEMP2
11652  042646  006367  177416                  ASL     TEMP3           ;SHIFT TEMP3 LEFT.
11653  042652  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11654  042656  001004                          BNE     REGR1
11655  042660  000167  001010                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11656  042664  004767  001032          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11657  042670  010767  135276          REGR1:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11658  042674  013701  042264                  MOV     @#TEMP1,%1      ;LOAD R1 WITH THE CONTENTS OF TEMP1
11659  042700  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
```

```
11660  042706  001004                        BNE     2$                      ;IF NOT THEN GO TO 2$
11661  042710  013702  042266                MOV     @#TEMP2,R2              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11662  042714  072102                        ASH     R2,R1                   ;USING R1
11663  042716  000402                        BR      4$
11664  042720  072167  177342          2$:   ASH     TEMP2,%1                ;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
11665  042724  106737  042262          4$:   MFPS    @#PSWORD                ;SAVE PS
11666  042730  123737  042272  042262        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11667  042736  001403                        BEQ     .+10
11668  042740  004767  015676                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11669                                                                          ;THE PS IS NOT EQUAL TO 0
11670  042744  000003                        3                       ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11671                                                                  ;BY  (013746  000172  000207)
11672
11673  042746  005237  042260                INC     @#COUNT                 ;INCREMENT THE COUNTER
11674  042752  023701  042270                CMP     @#TEMP3,%1              ;IS THE RESULT IN R1 EQUAL TO TEMP3?
11675  042756  001403                        BEQ     .+10
11676  042760                          6$:
11677  042760  004767  015656                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11678                                                                          ;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
11679  042764  000004                        4                       ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11680                                                                  ;BY  (013746  000172  000207)
11681
11682  042766  021537  042260                CMP     (R5),@#COUNT            ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11683  042772  001372                        BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
11684  042774  005215                        INC     (R5)
11685  042776  010767  135170                MOV     PC,LPADR                ;STORE ERROR LOOP ADDRESS
11686  043002  021527  000037                CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11687                                                                          ;BY 14. AND RIGHT BY 14.?
11688  043006  002011                        BGE     8$
11689  043010  005237  042266                INC     @#TEMP2
11690  043014  006367  177250                ASL     TEMP3                   ;SHIFT TEMP3 LEFT
11691  043020  021527  000020                CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11692  043024  001004                        BNE     REGR2
11693  043026  000167  000642                JMP     NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11694  043032  004767  000664          8$:   JSR     PC,TST37                ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11695  043036  010767  135130          REGR2: MOV    PC,LPADR                ;STORE ERROR LOOP ADDRESS
11696  043042  013702  042264                MOV     @#TEMP1,%2              ;LOAD R2 WITH THE CONTENTS OF TEMP1
11697  043046  032737  000001  000306        BIT     #1,@#$PASS              ;IS IT AN EVEN PASS ?
11698  043054  001004                        BNE     2$                      ;IF NOT THEN GO TO 2$
11699  043056  013703  042266                MOV     @#TEMP2,R3              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11700  043062  072203                        ASH     R3,R2                   ;USING R2
11701  043064  000402                        BR      4$
11702  043066  072267  177174          2$:   ASH     TEMP2,%2                ;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
11703  043072  106737  042262          4$:   MFPS    @#PSWORD                ;SAVE PS
11704  043076  123737  042272  042262        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11705  043104  001403                        BEQ     .+10
11706  043106  004767  015530                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11707                                                                          ;THE PS IS NOT EQUAL TO 0
11708  043112  000005                        5                       ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11709                                                                  ;BY  (013746  000172  000207)
11710
11711  043114  005237  042260                INC     @#COUNT
11712  043120  023702  042270                CMP     @#TEMP3,%2              ;IS THE RESULT IN R2 EQUAL TO TEMP3?
11713  043124  001403                        BEQ     .+10
11714  043126                          6$:
11715  043126  004767  015510                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```
11716
11717  043132  000006                          6               ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
                                                               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11718                                                          ;BY  (013746  000172  000207)
11719
11720  043134  021537  042260                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11721  043140  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11722  043142  005215                          INC     (R5)
11723  043144  010767  135022                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11724  043150  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11725                                                                  ;LEFT BY 14, AND RIGHT BY 14.?
11726  043154  002011                          BGE     8$
11727  043156  005237  042266                  INC     @#TEMP2
11728  043162  006367  177102                  ASL     TEMP3           ;SHIFTED TEMP3 LEFT
11729  043166  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11730  043172  001004                          BNE     REGR3
11731  043174  000167  000474                  JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11732  043200  004767  000516          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11733  043204  010767  134762          REGR3:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11734  043210  013703  042264                  MOV     @#TEMP1,%3      ;LOAD R3 WITH THE CONTENTS OF TEMP1
11735  043214  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11736  043222  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11737  043224  013704  042266                  MOV     @#TEMP2,R4      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11738  043230  072304                          ASH     R4,R3           ;USING R3
11739  043232  000402                          BR      4$
11740  043234  072367  177026          2$:     ASH     TEMP2,%3        ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
11741  043240  106737  042262          4$:     MFPS    @#PSWORD        ;SAVE PS
11742  043244  123737  042272  042262          CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11743  043252  001403                          BEQ     .+10
11744  043254  004767  015362                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11745                                                                  ;THE PS IS NOT EQUAL TO 0.
11746  043260  000007                          7               ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11747                                                          ;BY  (013746  000172  000207)
11748
11749  043262  005237  042260                  INC     @#COUNT
11750  043266  023703  042270                  CMP     @#TEMP3,%3      ;IS THE RESULT IN R3 EQUAL TO TEMP3?
11751  043272  001403                          BEQ     .+10
11752  043274                          6$:
11753  043274  004767  015342                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11754                                                                  ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
11755  043300  000010                          10              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11756                                                          ;BY  (013746  000172  000207)
11757
11758  043302  021537  042260                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11759  043306  001372                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11760  043310  005215                          INC     (R5)
11761  043312  010767  134654                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11762  043316  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11763                                                                  ;LEFT BY 14, AND RIGHT BY 14.?
11764  043322  002010                          BGE     8$
11765  043324  005237  042266                  INC     @#TEMP2
11766  043330  006367  176734                  ASL     TEMP3           ;SHIFT TEMP3 LEFT?
11767  043334  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11768  043340  001003                          BNE     REGR4
11769  043342  000554                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11770  043344  004767  000352          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11771  043350  010767  134616          REGR4:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
```

```
11772  043354  013704  042264                  MOV     @#TEMP1,%4      ;LOAD R4 WITH THE CONTENTS OF TEMP1
11773  043360  010501                          MOV     R5,R1           ;SAVE R5
11774  043362  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11775  043370  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11776  043372  013705  042266                  MOV     @#TEMP2,R5      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11777  043376  072405                          ASH     R5,R4           ;USING R4
11778  043400  000402                          BR      4$
11779  043402  072467  176660          2$:     ASH     TEMP2,%4        ;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
11780  043406  106737  042262          4$:     MFPS    @#PSWORD        ;SAVE PS
11781  043412  123737  042272  042262          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11782  043420  001403                          BEQ     .+10
11783  043422  004767  015214                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11784                                                                  ;THE PS IS NOT EQUAL TO 0
11785  043426  000011                          11              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11786                                                          ;BY  (013746  000172  000207)
11787
11788  043430  005237  042260                  INC     @#COUNT
11789  043434  023704  042270                  CMP     @#TEMP3,%4      ;IS THE RESULT IN R4 EQUAL TO TEMP3?
11790  043440  001403                          BEQ     .+10
11791  043442                          6$:
11792  043442  004767  015174                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11793                                                                  ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
11794  043446  000012                          12              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11795                                                          ;BY  (013746  000172  000207)
11796
11797  043450  010105                          MOV     R1,R5           ;RESTORE R5
11798  043452  021537  042260                  CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11799  043456  001371                          BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11800  043460  005215                          INC     (R5)
11801  043462  010767  134504                  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11802  043466  021527  000037                  CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN
11803                                                                  ;SHIFTED LEFT BY 14. AND RIGHT BY 14.?
11804  043472  002010                          BGE     8$
11805  043474  005237  042266                  INC     @#TEMP2
11806  043500  006367  176564                  ASL     TEMP3           ;SHIFT TEMP3 LEFT
11807  043504  021527  000020                  CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
11808  043510  001003                          BNE     REGR5
11809  043512  000470                          BR      NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11810  043514  004767  000202          8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11811  043520  010767  134446          REGR5:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11812  043524  010501                          MOV     R5,R1           ;SAVE R5
11813  043526  013705  042264                  MOV     @#TEMP1,%5      ;LOAD R5 WITH THE CONTENTS OF TEMP1
11814  043532  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11815  043540  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
11816  043542  013700  042266                  MOV     @#TEMP2,R0      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11817  043546  072500                          ASH     R0,R5           ;USING R5
11818  043550  000402                          BR      4$
11819  043552  072567  176510          2$:     ASH     TEMP2,%5        ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
11820  043556  106737  042262          4$:     MFPS    @#PSWORD        ;SAVE PS
11821  043562  123737  042272  042262          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11822  043570  001403                          BEQ     .+10
11823  043572  004767  015044                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11824                                                                  ;THE PS IS NOT EQUAL TO 0.
11825  043576  000013                          13              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11826                                                          ;BY  (013746  000172  000207)
11827
```

```
11828  043600  005237  042260          INC     @#COUNT
11829  043604  023705  042270          CMP     @#TEMP3,%5      ;IS THE RESULT IN R5 EQUAL TO TEMP3?
11830  043610  001403                  BEQ     .+10
11831  043612                  6$:
11832  043612  004767  015024          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11833                                                  ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE
11834  043616  000014                  14              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11835                                                  ;BY  (013746  000172  000207)
11836
11837  043620  021137  042260          CMP     (R1),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11838  043624  001372                  BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11839  043626  010105                  MOV     R1,R5           ;RESTORE R5
11840  043630  005215                  INC     (R5)
11841  043632  010767  134334          MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11842  043636  021527  000037          CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11843                                                          ;LEFT BY 14. AND RIGHT BY 14.?
11844  043642  002010                  BGE     8$              ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11845  043644  005237  042266          INC     @#TEMP2
11846  043650  006367  176414          ASL     TEMP3           ;SHIFT TEMP3 LEFT
11847  043654  021527  000020          CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11848  043660  001405                  BEQ     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11849  043662  000402                  BR      10$
11850  043664  004767  000032          8$:     JSR     PC,TST37
11851  043670  000167  176626          10$:    JMP     ASTART          ;GO BACK TO START
11852  043674  012737  040000  042264  NEGAT:  MOV     #40000,@#TEMP1  ;TEMP1=40000
11853  043702  012737  177762  042266          MOV     #177762,@#TEMP2 ;TEMP2=177762
11854  043710  012737  000001  042270          MOV     #1,@#TEMP3      ;TEMP3=1
11855  043716  000167  176600                  JMP     ASTART
11856  043722  021527  000037          TST37:  CMP     (R5),#37        ;IS IT TEST 37?
11857  043726  001013                  BNE     TST40           ;IF NOT THEN TRY TEST 40
11858  043730  005037  042264          CLR     @#TEMP1         ;0
11859  043734  012737  000020  042266  MOV     #16.,@#TEMP2    ;SHIFTED BY 16
11860  043742  005037  042270          CLR     @#TEMP3         ;IS=0
11861  043746  012737  000004  042272  MOV     #4,@#TEMP4      ;AND PS=4
11862  043754  000207                  RTS     PC
11863  043756  021527  000040          TST40:  CMP     (R5),#40        ;IS IT TEST 40?
11864  043762  001003                  BNE     TST41           ;IF NOT THEN TRY TEST 41
11865  043764  005037  042266          CLR     @#TEMP2         ;0 SHIFTED BY 0=0 AND PS=4
11866  043770  000207                  RTS     PC
11867  043772  021527  000041          TST41:  CMP     (R5),#41        ;IS IT TEST 41?
11868  043776  001004                  BNE     TST42           ;IF NOT THEN TRY TEST 42
11869  044000  012737  177760  042266  MOV     #-16.,@#TEMP2   ;0 SHIFTED BY -16.=0 AND PS=4
11870  044006  000207                  RTS     PC
11871  044010  021527  000042          TST42:  CMP     (R5),#42        ;IS IT TEST 42?
11872  044014  001013                  BNE     TST43           ;IF NOT THEN TRY TEST 43
11873  044016  012737  100000  042264  MOV     #100000,@#TEMP1 ;100000
11874  044024  005237  042266          INC     @#TEMP2         ;SHIFTED BY -15
11875  044030  005337  042270          DEC     @#TEMP3         ;IS=-1
11876  044034  012737  000010  042272  MOV     #10,@#TEMP4             ;AND PS=10
11877  044042  000207                  RTS     PC
11878  044044  021527  000043          TST43:  CMP     (R5),#43        ;IS IT TEST 43?
11879  044050  001012                  BNE     TST44           ;IF NOT THEN IF NOT THEN TRY TEST 44
11880  044052  012737  125252  042264  MOV     #125252,@#TEMP1 ;125252
11881  044060  012737  177777  042266  MOV     #-1,@#TEMP2     ;SHIFTED BY -1
11882  044066  012737  152525  042270  MOV     #152525,@#TEMP3 ;IS=152525 AND PS=10
11883  044074  000207                  RTS     PC
```

```
11884  044076  021527  000044          TST44:  CMP     (R5),#44        ;IS IT TEST 44?
11885  044102  001012                          BNE     TST45           ;IF NOT THEN TRY TEST 45
11886  044104  012737  000001  042266          MOV     #1,@#TEMP2      ;125252 SHIFTED BY 1
11887  044112  012737  052524  042270          MOV     #52524,@#TEMP3  ;IS=52524
11888  044120  012737  000003  042272          MOV     #3,@#TEMP4      ;AND PS=3
11889  044126  000207                          RTS     PC
11890  044130  021527  000045          TST45:  CMP     (R5),#45        ;IS IT TEST 45?
11891  044134  001012                          BNE     TST46           ;IF NOT THEN TRY TEST 46
11892  044136  012737  177776  042266          MOV     #-2,@#TEMP2     ;125252 SHIFTED BY -2
11893  044144  012737  165252  042270          MOV     #165252,@#TEMP3 ;IS=165252
11894  044152  012737  000011  042272          MOV     #11,@#TEMP4     ;AND PS=11
11895  044160  000207                          RTS     PC
11896  044162  021527  000046          TST46:  CMP     (R5),#46        ;IS IT TEST 46?
11897  044166  001014                          BNE     TST47           ;IF NOT THEN TRY TEST 47
11898  044170  012737  177777  042264          MOV     #-1,@#TEMP1     ;-1
11899  044176  012737  000020  042266          MOV     #16.,@#TEMP2    ;SHIFTED BY 15.
11900  044204  005037  042270                  CLR     @#TEMP3         ;IS=0
11901  044210  012737  000007  042272          MOV     #7,@#TEMP4      ;AND PS=7
11902  044216  000207                          RTS     PC
11903  044220  021527  000047          TST47:  CMP     (R5),#47        ;IS IT TEST 47?
11904  044224  001011                          BNE     TST50           ;IF NOT THEN TRY TEST 50
11905  044226  005337  042266                  DEC     @#TEMP2         ;-1 SHIFTED BY 15
11906  044232  012737  100000  042270          MOV     #100000,@#TEMP3 ;IS=100000
11907  044240  012737  000011  042272          MOV     #11,@#TEMP4     ;AND PS=11
11908  044246  000207                          RTS     PC
11909  044250  021527  000050          TST50:  CMP     (R5),#50        ;IS IT TEST 50
11910  044254  001007                          BNE     ENT51           ;IF NOT THEN TRY TEST 51
11911  044256  012737  137777  042264          MOV     #137777,@#TEMP1 ;137777 SHIFTED BY 15. IS=100000
11912  044264  012737  000013  042272          MOV     #13,@#TEMP4     ;AND PS=13
11913  044272  000207                          RTS     PC
11914  044274  021527  000051          ENT51:  CMP     (R5),#51        ;IS IT ENTERING TEST 51?
11915  044300  001403                          BEQ     .+10
11916  044302  004767  014334                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11917                                                                  ;TEST NUMBER GOOFED
11918  044306  000015                          15              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11919                                                          ;BY  (013746  000172  000207)
11920
11921
11922  044310  005726                          TST     (SP)+           ;RESTORE STACK POINTER
11923  044312  012704  177771                  MOV     #-7,%4
11924  044316  012702  042304                  MOV     #S1,%2
11925  044322  012703  042306                  MOV     #S2,%3
```

```
11926                                   ;*****************************************************************************
11927                                   ;TEST:51   11/34   ASH     125252 SHIFTED BY #5 = 52500  PS = 3
11928                                   ;*****************************************************************************
11929
11930  044326  010767  133640          TST51:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11931  044332  012701  125252                  MOV     #125252,%1      ;LOAD R1 WITH 125252
11932  044336  072127  000005                  ASH     #5,%1           ;SHIFT R1 BY #5
11933  044342  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
11934  044346  122737  000003  042262          CMPB    #3,@#PSWORD     ;IS THE PS 3?
11935  044354  001403                          BEQ     .+10
11936  044356  004767  014260                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11937                                                                  ;THE PS IS NOT EQUAL TO 3
11938  044362  000016                                  16              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11939                                                                  ;BY  (013746  000172  000207)
11940
11941  044364  022701  052500                  CMP     #52500,%1       ;IS THE RESULT 52500?
11942  044370  001403                          BEQ     .+10
11943  044372                          1$:
11944  044372  004767  014244                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11945                                                                  ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
11946  044376  000017                                  17              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11947                                                                  ;BY  (013746  000172  000207)
11948
11949  044400  021527  000051                  CMP     (R5),#51                ;IS $TESTN = #51
11950  044404  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11951  044406  005215                          INC     (R5)
11952
11953
11954
11955                                   ;*****************************************************************************
11956                                   ;TEST:52   11/34   ASH     125252 SHIFTED BY @S2 = 177525  PS = 10
11957                                   ;*****************************************************************************
11958
11959  044410  010767  133556          TST52:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11960  044414  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
11961  044420  072077  175662                  ASH     @S2,%0          ;SHIFT R0 BY @S2
11962  044424  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
11963  044430  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
11964  044436  001403                          BEQ     .+10
11965  044440  004767  014176                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11966                                                                  ;THE PS IS NOT EQUAL TO 10
11967  044444  000020                                  20              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11968                                                                  ;BY  (013746  000172  000207)
11969
11970  044446  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
11971  044452  001403                          BEQ     .+10
11972  044454                          1$:
11973  044454  004767  014162                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11974                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
11975  044460  000021                                  21              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11976                                                                  ;BY  (013746  000172  000207)
11977
11978  044462  021527  000052                  CMP     (R5),#52                ;IS $TESTN = #52
11979  044466  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11980  044470  005215                          INC     (R5)
11981
```

```
11982
11983
```

```
11984                                   ;*****************************************************************************
11985                                   ;TEST:53   11/34   ASH     125252 SHIFTED BY @#S1 = 177525  PS = 10
11986                                   ;*****************************************************************************
11987
11988  044472  010767  133474          TST53:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
11989  044476  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
11990  044502  072037  042304                  ASH     @#S1,%0         ;SHIFT R0 BY @#S1
11991  044506  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
11992  044512  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
11993  044520  001403                          BEQ     .+10
11994  044522  004767  014114                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11995                                                                  ;THE PS IS NOT EQUAL TO 10
11996  044526  000022                          22              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
11997                                                          ;BY  (013746  000172  000207)
11998
11999  044530  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12000  044534  001403                          BEQ     .+10
12001  044536                          1$:
12002  044536  004767  014100                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12003                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12004  044542  000023                          23              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12005                                                          ;BY  (013746  000172  000207)
12006
12007  044544  021527  000053                  CMP     (R5),#53                ;IS $TESTN = #53
12008  044550  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12009  044552  005215                          INC     (R5)
12010
12011
12012
12013                                   ;*****************************************************************************
12014                                   ;TEST:54   11/34   ASH     125252 SHIFTED BY (2) = 177525  PS = 10
12015                                   ;*****************************************************************************
12016
12017  044554  010767  133412          TST54:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12018  044560  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12019  044564  072012                          ASH     (2),%0          ;SHIFT R0 BY (2)
12020  044566  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12021  044572  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12022  044600  001403                          BEQ     .+10
12023  044602  004767  014034                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12024                                                                  ;THE PS IS NOT EQUAL TO 10
12025  044606  000024                          24              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12026                                                          ;BY  (013746  000172  000207)
12027
12028  044610  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12029  044614  001403                          BEQ     .+10
12030  044616                          1$:
12031  044616  004767  014020                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12032                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12033  044622  000025                          25              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12034                                                          ;BY  (013746  000172  000207)
12035
12036  044624  021527  000054                  CMP     (R5),#54                ;IS $TESTN = #54
12037  044630  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12038  044632  005215                          INC     (R5)
12039
```

```
12040
12041
```

```
12042                                   ;*****************************************************************************
12043                                   ;TEST:55   11/34   ASH     125252 SHIFTED BY (2)+ = 177525  PS = 10
12044                                   ;*****************************************************************************
12045
12046  044634  010767  133332          TST55:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12047  044640  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12048  044644  072022                          ASH     (2)+,%0         ;SHIFT R0 BY (2)+
12049  044646  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12050  044652  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12051  044660  001403                          BEQ     .+10
12052  044662  004767  013754                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12053                                                                  ;THE PS IS NOT EQUAL TO 10
12054  044666  000026                                  26              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12055                                                                  ;BY  (013746  000172  000207)
12056
12057  044670  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12058  044674  001403                          BEQ     .+10
12059  044676                          1$:
12060  044676  004767  013740                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12061                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12062  044702  000027                                  27              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12063                                                                  ;BY  (013746  000172  000207)
12064
12065  044704  021527  000055                  CMP     (R5),#55                ;IS $TESTN = #55
12066  044710  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12067  044712  005215                          INC     (R5)
12068
12069
12070
12071                                   ;*****************************************************************************
12072                                   ;TEST:56   11/34   ASH     125252 SHIFTED BY -(2) = 177525  PS = 10
12073                                   ;*****************************************************************************
12074
12075  044714  010767  133252          TST56:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12076  044720  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12077  044724  072042                          ASH     -(2),%0         ;SHIFT R0 BY -(2)
12078  044726  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12079  044732  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12080  044740  001403                          BEQ     .+10
12081  044742  004767  013674                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12082                                                                  ;THE PS IS NOT EQUAL TO 10
12083  044746  000030                                  30              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12084                                                                  ;BY  (013746  000172  000207)
12085
12086  044750  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12087  044754  001403                          BEQ     .+10
12088  044756                          1$:
12089  044756  004767  013660                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12090                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12091  044762  000031                                  31              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12092                                                                  ;BY  (013746  000172  000207)
12093
12094  044764  021527  000056                  CMP     (R5),#56                ;IS $TESTN = #56
12095  044770  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12096  044772  005215                          INC     (R5)
12097
```

```
12098
12099
```

```
12100                                   ;*****************************************************************************
12101                                   ;TEST:57   11/34   ASH     125252 SHIFTED BY 2(3) = 177252  PS = 11
12102                                   ;*****************************************************************************
12103
12104  044774  010767  133172          TST57:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12105  045000  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12106  045004  072063  000002                  ASH     2(3),%0         ;SHIFT R0 BY 2(3)
12107  045010  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12108  045014  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12109  045022  001403                          BEQ     .+10
12110  045024  004767  013612                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12111                                                                  ;THE PS IS NOT EQUAL TO 11
12112  045030  000032                          32              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12113                                                          ;BY  (013746  000172  000207)
12114
12115  045032  022700  177252                  CMP     #177252,%0      ;IS THE RESULT 177252?
12116  045036  001403                          BEQ     .+10
12117  045040                          1$:
12118  045040  004767  013576                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12119                                                                  ;R0 IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
12120  045044  000033                          33              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12121                                                          ;BY  (013746  000172  000207)
12122
12123  045046  021527  000057                  CMP     (R5),#57                ;IS $TESTN = #57
12124  045052  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12125  045054  005215                          INC     (R5)
12126
12127
12128
12129                                   ;*****************************************************************************
12130                                   ;TEST:60   11/34   ASH     125252 SHIFTED BY @(3) = 177525  PS = 10
12131                                   ;*****************************************************************************
12132
12133  045056  010767  133110          TST60:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12134  045062  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12135  045066  072073  000000                  ASH     @(3),%0         ;SHIFT R0 BY @(3)
12136  045072  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12137  045076  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12138  045104  001403                          BEQ     .+10
12139  045106  004767  013530                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12140                                                                  ;THE PS IS NOT EQUAL TO 10
12141  045112  000034                          34              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12142                                                          ;BY  (013746  000172  000207)
12143
12144  045114  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12145  045120  001403                          BEQ     .+10
12146  045122                          1$:
12147  045122  004767  013514                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12148                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12149  045126  000035                          35              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12150                                                          ;BY  (013746  000172  000207)
12151
12152  045130  021527  000060                  CMP     (R5),#60                ;IS $TESTN = #60
12153  045134  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12154  045136  005215                          INC     (R5)
12155
```

```
12156
12157
```

```
12158                                   ;*****************************************************************************
12159                                   ;TEST:61   11/34   ASH     125252 SHIFTED BY @(3)+ = 177525  PS = 10
12160                                   ;*****************************************************************************
12161
12162  045140  010767  133026          TST61:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12163  045144  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12164  045150  072033                          ASH     @(3)+,%0        ;SHIFT R0 BY @(3)+
12165  045152  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12166  045156  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12167  045164  001403                          BEQ     .+10
12168  045166  004767  013450                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12169                                                                  ;THE PS IS NOT EQUAL TO 10
12170  045172  000036                          36              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12171                                                          ;BY  (013746  000172  000207)
12172
12173  045174  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12174  045200  001403                          BEQ     .+10
12175  045202                          1$:
12176  045202  004767  013434                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12177                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12178  045206  000037                          37              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12179                                                          ;BY  (013746  000172  000207)
12180
12181  045210  021527  000061                  CMP     (R5),#61                ;IS $TESTN = #61
12182  045214  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12183  045216  005215                          INC     (R5)
12184
12185
12186
12187                                   ;*****************************************************************************
12188                                   ;TEST:62   11/34   ASH     125252 SHIFTED BY @-(3) = 177525  PS = 10
12189                                   ;*****************************************************************************
12190
12191  045220  010767  132746          TST62:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12192  045224  012700  125252                  MOV     #125252,%0      ;LOAD R0 WITH 125252
12193  045230  072053                          ASH     @-(3),%0        ;SHIFT R0 BY @-(3)
12194  045232  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12195  045236  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12196  045244  001403                          BEQ     .+10
12197  045246  004767  013370                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12198                                                                  ;THE PS IS NOT EQUAL TO 10
12199  045252  000040                          40              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12200                                                          ;BY  (013746  000172  000207)
12201
12202  045254  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12203  045260  001403                          BEQ     .+10
12204  045262                          1$:
12205  045262  004767  013354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12206                                                                  ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12207  045266  000041                          41              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12208                                                          ;BY  (013746  000172  000207)
12209
12210  045270  021527  000062                  CMP     (R5),#62                ;IS $TESTN = #62
12211  045274  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12212  045276  005215                          INC     (R5)
12213
```

```
12214
12215
```

```
12216                                   ;**********************************
12217                                   ;     ASHC INSTRUCTION TESTS
12218                                   ;**********************************
12219
12220
12221
12222                                           ;*************
12223                                           ;TESTS 63-157
12224                                           ;*************
12225
12226
12227
12228  045300  010767  132666                   MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12229  045304  012737  000062  042260           MOV     #62,@#COUNT
12230  045312  005037  042264                   CLR     @#TEMP1         ;TEMP1=0
12231  045316  012737  000001  042266           MOV     #1,@#TEMP2      ;TEMP2=1
12232  045324  005037  042270                   CLR     @#TEMP3         ;TEMP3=0
12233  045330  005037  042272                   CLR     @#TEMP4         ;TEMP4=0
12234  045334  012737  000001  042274           MOV     #1,@#TEMP5      ;TEMP5=1
12235  045342  005037  042276                   CLR     @#TEMP6         ;0 1 SHIFTED BY 0=0 1, PS=0
12236
12237  045346  010502                   REG01:  MOV     R5,R2           ;SAVE R5
12238  045350  013700  042264                   MOV     @#TEMP1,%0      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
12239  045354  013701  042266                   MOV     @#TEMP2,%0!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
12240  045360  000241                           CLC
12241  045362  032737  000001  000306           BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12242  045370  001004                           BNE     2$              ;IF NOT THEN GO TO 2$
12243  045372  013705  042270                   MOV     @#TEMP3,R5      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12244  045376  073005                           ASHC    R5,R0           ;USING R0
12245  045400  000402                           BR      4$
12246  045402  073067  174662           2$:     ASHC    TEMP3,%0        ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
12247  045406  106737  042262           4$:     MFPS    @#PSWORD        ;SAVE PS
12248  045412  123737  042276  042262           CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12249  045420  001403                           BEQ     .+10
12250  045422  004767  013214                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12251                                                           ;WRONG PS
12252  045426  000042                           42              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12253                                                           ;BY  (013746  000172  000207)
12254
12255  045430  005237  042260                   INC     @#COUNT
12256  045434  023700  042272                   CMP     @#TEMP4,%0      ;IS THE RESULT IN R0 SAME AS TEMP4?
12257  045440  001403                           BEQ     .+10
12258  045442  004767  013174                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12259                                                           ;WRONG RESULT IN R0
12260  045446  000043                           43              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12261                                                           ;BY  (013746  000172  000207)
12262
12263  045450  023701  042274                   CMP     @#TEMP5,%1      ;IS THE RESULT IN R1 SAME AS TEMP5?
12264  045454  001403                           BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12265                                                           ;AND PS=TEMP6
12266  045456  004767  013160                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12267                                                           ;WRONG RESULT IN R1
12268  045462  000044                           44              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12269                                                           ;BY  (013746  000172  000207)
12270
12271  045464  010205                           MOV     R2,R5           ;RESTORE R5
```

```
12272  045466  021537  042260          CMP     (R5),@#COUNT    ;IS TEST NUMBER=COUNTER?
12273  045472  001403                  BEQ     .+10
12274  045474  004767  013142          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12275                                                  ;NO
12276  045500  000045                  45              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12277                                                  ;BY  (013746  000172  000207)
12278
12279  045502  005215                  INC     (R5)
12280  045504  021527  000160          CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12281  045510  002014                  BGE     6$              ;YES
12282  045512  005237  042270          INC     @#TEMP3
12283  045516  000241                  CLC
12284  045520  006137  042274          ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12285  045524  006137  042272          ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP4 IN TEMP5
12286  045530  021527  000121          CMP     (R5),#121       ;IS IT TEST 121?
12287  045534  001004                  BNE     REGR23
12288  045536  004467  000414          JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12289  045542  004767  000444  6$:     JSR     %7,TST160
12290  045546  010767  132420  REGR23: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12291  045552  013702  042264          MOV     @#TEMP1,%2      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
12292  045556  013703  042266          MOV     @#TEMP2,%2+1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
12293  045562  000241                  CLC
12294  045564  032737  000001  000306  BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12295  045572  001004                  BNE     2$              ;IF NOT THEN GO TO 2$
12296  045574  013704  042270          MOV     @#TEMP3,R4      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12297  045600  073204                  ASHC    R4,R2           ;USING R2
12298  045602  000402                  BR      4$
12299  045604  073267  174460  2$:     ASHC    TEMP3,%2        ;ASHC REGISTER 2 BY THE CONTETNS OF TEMP3
12300  045610  106737  042262  4$:     MFPS    @#PSWORD        ;SAVE PS
12301  045614  123737  042276  042262  CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12302  045622  001403                  BEQ     .+10
12303  045624  004767  013012          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12304                                                  ;WRONG PS
12305  045630  000046                  46              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12306                                                  ;BY  (013746  000172  000207)
12307
12308  045632  005237  042260          INC     @#COUNT
12309  045636  023702  042272          CMP     @#TEMP4,%2      ;IS THE RESULT IN R2 SAME AS TEMP4?
12310  045642  001403                  BEQ     .+10
12311  045644  004767  012772          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12312                                                  ;WRONG RESULT IN R2
12313  045650  000047                  47              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12314                                                  ;BY  (013746  000172  000207)
12315
12316  045652  023703  042274          CMP     @#TEMP5,%3      ;IS THE RESULT IN R3 SAME AS TEMP5?
12317  045656  001403                  BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12318                                                  ;AND PS=TEMP6
12319  045660  004767  012756          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12320                                                  ;WRONG RESULT IN R1
12321  045664  000050                  50              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12322                                                  ;BY  (013746  000172  000207)
12323
12324  045666  021537  042260          CMP     (R5),@#COUNT    ;IS TEST NUMBER=COUNTER?
12325  045672  001403                  BEQ     .+10
12326  045674  004767  012742          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12327                                                  ;NO
```

```
12328  045700  000051                          51              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12329                                                           ;BY  (013746  000172  000207)
12330
12331  045702  005215                          INC     (R5)
12332  045704  021527  000160                  CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12333  045710  002014                          BGE     6$              ;YES
12334  045712  005237  042270                  INC     @#TEMP3
12335  045716  000241                          CLC
12336  045720  006137  042274                  ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12337  045724  006137  042272                  ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
12338  045730  021527  000121                  CMP     (R5),#121       ;IS IT TEST 121?
12339  045734  001004                          BNE     REG45
12340  045736  004467  000214                  JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12341  045742  004767  000244          6$:     JSR     %7,TST160
12342  045746  010767  132220          REG45:  MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12343  045752  010501                          MOV     R5,R1           ;SAVE R5
12344  045754  013704  042264                  MOV     @#TEMP1,%4      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
12345  045760  013705  042266                  MOV     @#TEMP2,%4!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
12346  045764  000241                          CLC
12347  045766  032737  000001  000306          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
12348  045774  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
12349  045776  013700  042270                  MOV     @#TEMP3,R0      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12350  046002  073400                          ASHC    R0,R4           ;USING R4
12351  046004  000402                          BR      4$
12352  046006  073467  174256          2$:     ASHC    TEMP3,%4        ;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
12353  046012  106737  042262          4$:     MFPS    @#PSWORD        ;SAVE PS
12354  046016  123737  042276  042262          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12355  046024  001403                          BEQ     .+10
12356  046026  004767  012610                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12357                                                           ;WRONG PS
12358  046032  000052                          52              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12359                                                           ;BY  (013746  000172  000207)
12360
12361  046034  005237  042260                  INC     @#COUNT
12362  046040  023704  042272                  CMP     @#TEMP4,%4      ;IS THE RESULT IN R4 SAME AS TEMP4?
12363  046044  001403                          BEQ     .+10
12364  046046  004767  012570                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12365                                                           ;WRONG RESULT IN R4
12366  046052  000053                          53              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12367                                                           ;BY  (013746  000172  000207)
12368
12369  046054  023705  042274                  CMP     @#TEMP5,%5      ;IS THE RESULT IN R5 SAME AS TEMP5?
12370  046060  001403                          BEQ     .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12371                                                           ;AND PS=TEMP6
12372  046062  004767  012554                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12373                                                           ;WRONG RESULT IN R5
12374  046066  000054                          54              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12375                                                           ;BY  (013746  000172  000207)
12376
12377  046070  021137  042260                  CMP     (R1),@#COUNT    ;IS TEST NUMBER=COUNTER?
12378  046074  001403                          BEQ     .+10
12379  046076  004767  012540                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12380                                                           ;NO
12381  046102  000055                          55              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12382                                                           ;BY  (013746  000172  000207)
12383
```

```
12384  046104  010105                        MOV     R1,R5           ;RESTORE R5
12385  046106  005215                        INC     (R5)
12386  046110  021527  000160                CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12387  046114  002014                        BGE     6$              ;YES
12388  046116  005237  042270                INC     @#TEMP3
12389  046122  000241                        CLC
12390  046124  006137  042274                ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
12391  046130  006137  042272                ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
12392  046134  021527  000121                CMP     (R5),#121       ;IS IT TEST 121?
12393  046140  001004                        BNE     8$
12394  046142  004467  000010                JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12395  046146  004767  000040        6$:     JSR     %7,TST160
12396  046152  000167  177170        8$:     JMP     REG01
12397  046156  022424                RITSH:  CMP     (R4)+,(R4)+     ;MAKE R4 POINT TO THE NEXT REG TAG
12398  046160  012737  040000  042264        MOV     #40000,@#TEMP1  ;TEMP1=4000
12399  046166  005037  042266                CLR     @#TEMP2         ;TEMP2=0
12400  046172  012737  177742  042270        MOV     #-30.,@#TEMP3   ;TEMP3=-30
12401  046200  005037  042272                CLR     @#TEMP4         ;TEMP4=0
12402  046204  005237  042274                INC     @#TEMP5         ;TEMP5=1
12403  046210  000204                        RTS     R4
12404  046212  021527  000160        TST160: CMP     (R5),#160       ;IS IT TEST 160
12405  046216  001010                        BNE     TST161          ;IF NOT THEN TRY TEST 161
12406  046220  005037  042264                CLR     @#TEMP1         ;0 0 SHIFTED BY 0
12407  046224  005037  042272                CLR     @#TEMP4         ;IS EQUAL TO 0 0
12408  046230  012737  000004  042276        MOV     #4,@#TEMP6      ;AND PS=4
12409  046236  000207                        RTS     %7
12410  046240  021527  000161        TST161: CMP     (R5),#161       ;IS IT TEST 161
12411  046244  001004                        BNE     TST162
12412  046246  012737  177746  042270        MOV     #-32,@#TEMP3    ;0 0 SHIFTED BY -32=0 0, PS=4
12413  046254  000207                        RTS     %7
12414  046256  021527  000162        TST162: CMP     (R5),#162       ;IS IT TEST 162
12415  046262  001004                        BNE     TST163          ;IF NOT THEN TRY TEST 163
12416  046264  012737  000032  042270        MOV     #32,@#TEMP3     ;0 0 SHIFTED BY 32=0 0, PS=4
12417  046272  000207                        RTS     %7
12418  046274  021527  000163        TST163: CMP     (R5),#163       ;IS IT TEST 163?
12419  046300  001016                        BNE     TST164          ;IF NOT THEN TRY TEST 164
12420  046302  012737  052525  042264        MOV     #52525,@#TEMP1  ;52525 0
12421  046310  012737  177760  042270        MOV     #-16.,@#TEMP3   ;SHIFTED BY -16.
12422  046316  005037  042272                CLR     @#TEMP4
12423  046322  012737  052525  042274        MOV     #52525,@#TEMP5  ;IS EQUAL TO 0 52525
12424  046330  005037  042276                CLR     @#TEMP6         ;AND PS = 0
12425  046334  000207                        RTS     %7
12426  046336  021527  000164        TST164: CMP     (R5),#164       ;IS IT TEST 164?
12427  046342  001014                        BNE     TST165          ;IF NOT THEN TRY TEST 165
12428  046344  012737  125252  042264        MOV     #125252,@#TEMP1 ;125252 0 SHIFTED BY -16.
12429  046352  005337  042272                DEC     @#TEMP4
12430  046356  012737  125252  042274        MOV     #125252,@#TEMP5 ;IS EQUAL TO -1 125252
12431  046364  012737  000010  042276        MOV     #10,@#TEMP6     ;AND PS=10
12432  046372  000207                        RTS     %7
12433  046374  021527  000165        TST165: CMP     (R5),#165       ;IS IT TEST 165?
12434  046400  001007                        BNE     TST166          ;IF NOT THEN TRY TEST 166
12435  046402  012737  177777  042264        MOV     #-1,@#TEMP1     ;-1 0 SHIFTED BY -16
12436  046410  012737  177777  042274        MOV     #-1,@#TEMP5     ;IS EQUAL TO -1 -1, AND PS=10
12437  046416  000207                        RTS     %7
12438  046420  021527  000166        TST166: CMP     (R5),#166       ;IS IT TEST 166?
12439  046424  001011                        BNE     TST167          ;IF NOT THEN TRY TEST 167
```

```
12440  046426  012737  100000  042264          MOV     #100000,@#TEMP1 ;100000 0
12441  046434  012737  177740  042270          MOV     #-32.,@#TEMP3   ;SHIFTED BY -32 IS EQUAL TO -1 -1
12442  046442  005237  042276                  INC     @#TEMP6         ;AND PS=11
12443  046446  000207                          RTS     %7
12444  046450  021527  000167          TST167: CMP     (R5),#167       ;IS IT TEST 167?
12445  046454  001014                          BNE     TST170          ;IF NOT THEN TRY TEST 170
12446  046456  005037  042264                  CLR     @#TEMP1
12447  046462  005337  042266                  DEC     @#TEMP2         ;0 -1
12448  046466  012737  000020  042270          MOV     #16.,@#TEMP3    ;SHIFTED BY 16.
12449  046474  005037  042274                  CLR     @#TEMP5         ;IS EQUAL TO -1 0
12450  046500  005237  042276                  INC     @#TEMP6         ;AND PS=12
12451  046504  000207                          RTS     %7
12452  046506  021527  000170          TST170: CMP     (R5),#170       ;IS IT TEST 170?
12453  046512  001007                          BNE     TST171          ;IF NOT THEN TRY TEST 171
12454  046514  012737  125252  042266          MOV     #125252,@#TEMP2 ;0 125252 SHIFTED BY 16
12455  046522  012737  125252  042272          MOV     #125252,@#TEMP4 ;IS EQUAL TO 125252 0, AND PS=12
12456  046530  000207                          RTS     %7
12457  046532  021527  000171          TST171: CMP     (R5),#171       ;IS IT TEST 171?
12458  046536  001010                          BNE     TST172          ;IF NOT THEN TRY TEST 172
12459  046540  005337  042270                  DEC     @#TEMP3         ;0 125252 SHIFTED BY 15
12460  046544  012737  052525  042272          MOV     #52525,@#TEMP4  ;IS EQUAL TO 52525 0
12461  046552  005037  042276                  CLR     @#TEMP6         ;AND PS=0
12462  046556  000207                          RTS     %7
12463  046560  021527  000172          TST172: CMP     (R5),#172       ;IS IT TEST 172?
12464  046564  001006                          BNE     TST173          ;IF NOT THEN TRY TEST 173
12465  046566  012737  052525  042266          MOV     #52525,@#TEMP2  ;0 52525
12466  046574  005237  042270                  INC     @#TEMP3         ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
12467  046600  000207                          RTS     %7
12468  046602  021527  000173          TST173: CMP     (R5),#173       ;IS IT TEST 173?
12469  046606  001014                          BNE     TST174          ;IF NOT THEN TRY TEST 174
12470  046610  012737  177777  042266          MOV     #-1,@#TEMP2     ;0 -1
12471  046616  005337  042270                  DEC     @#TEMP3         ;SHIFTED BY 15.
12472  046622  012737  077777  042272          MOV     #77777,@#TEMP4
12473  046630  012737  100000  042274          MOV     #100000,@#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
12474  046636  000207                          RTS     %7
12475  046640  021527  000174          TST174: CMP     (R5),#174       ;IS IT TEST 174?
12476  046644  001013                          BNE     TST175          ;IF NOT THEN TRY TEST 175
12477  046646  012737  100000  042264          MOV     #100000,@#TEMP1
12478  046654  005337  042266                  DEC     @#TEMP2         ;100000 -2 SHIFTED BY 15.
12479  046660  005037  042274                  CLR     @#TEMP5         ;IS EQUAL TO 77777 0
12480  046664  012737  000002  042276          MOV     #2,@#TEMP6      ;AND PS=2
12481  046672  000207                          RTS     %7
12482  046674  021527  000175          TST175: CMP     (R5),#175       ;IS IT TEST 175?
12483  046700  001015                          BNE     ENT176          ;IF NOT THEN TRY TEST 176
12484  046702  012737  177777  042264          MOV     #-1,@#TEMP1
12485  046710  005037  042266                  CLR     @#TEMP2         ;-1 0
12486  046714  005237  042270                  INC     @#TEMP3         ;SHIFTED BY 16.
12487  046720  005037  042272                  CLR     @#TEMP4         ;IS EQUAL TO 0 0
12488  046724  012737  000007  042276          MOV     #7,@#TEMP6      ;AND PS=7
12489  046732  000207                          RTS     %7
12490  046734  021527  000176          ENT176: CMP     (R5),#176       ;IS THE PROGRM ENTERING TEST 176?
12491  046740  001403                          BEQ     .+10
12492  046742  004767  011674                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12493                                                                  ;TEST NUMBER GOOFED
12494  046746  000056                          56              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12495                                                          ;BY  (013746  000172  000207)
```

```
12496
12497
12498  046750  005726                    TST     (SP)+           ;RESTORE STACK POINTER
12499
```

```
12500                                   ;*****************************************************************
12501                                   ;TEST:176       1 SHIFTED BY 8. = 400  PS = 0
12502                                   ;*****************************************************************
12503
12504  046752  010767  131214          TST176: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12505  046756  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12506  046762  012701  000001                  MOV     #1,%1!1         ;LOAD R1!1 WITH 1
12507  046766  000241                          CLC
12508  046770  073127  000010                  ASHC    #8.,%1          ;SHIFT R1,R1!1 BY 8.
12509  046774  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12510  047000  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12511  047006  001403                          BEQ     .+10
12512  047010  004767  011626                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12513                                                                  ;THE PS IS NOT EQUAL TO 0
12514  047014  000057                          57              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12515                                                          ;BY  (013746  000172  000207)
12516
12517  047016  022701  000400                  CMP     #400,%1         ;IS THE RESULT 400?
12518  047022  001403                          BEQ     .+10
12519  047024  004767  011612                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12520                                                                  ;R1 IS NOT EQUAL TO 400
12521  047030  000060                          60              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12522                                                          ;BY  (013746  000172  000207)
12523
12524  047032  021527  000176                  CMP     (R5),#176               ;IS $TESTN = #176?
12525  047036  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12526  047040  004767  011576                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12527                                                                  ;TEST IS IN WRONG SEQUENCE
12528  047044  000061                          61              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12529                                                          ;BY  (013746  000172  000207)
12530
12531  047046  005215                          INC     (R5)
12532
12533
12534                                   ;*****************************************************************
12535                                   ;TEST:177       -1 SHIFTED BY 15. = 100000  PS = 11
12536                                   ;*****************************************************************
12537
12538  047050  010767  131116          TST177: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12539  047054  012703  000000                  MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
12540  047060  012703  177777                  MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
12541  047064  000241                          CLC
12542  047066  073327  000017                  ASHC    #15.,%3         ;SHIFT R3,R3!1 BY 15.
12543  047072  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12544  047076  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12545  047104  001403                          BEQ     .+10
12546  047106  004767  011530                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12547                                                                  ;THE PS IS NOT EQUAL TO 11
12548  047112  000062                          62              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12549                                                          ;BY  (013746  000172  000207)
12550
12551  047114  022703  100000                  CMP     #100000,%3      ;IS THE RESULT 100000?
12552  047120  001403                          BEQ     .+10
12553  047122  004767  011514                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12554                                                                  ;R3 IS NOT EQUAL TO 100000
12555  047126  000063                          63              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
```

```
12556                                                   ;BY  (013740  000172  000207)
12557
12558  047130  021527  000177          CMP     (R5),#177               ;IS $TESTN = #177?
12559  047134  001403                  BEQ     .+10            ;IF NOT THEN GO TO HLT
12560  047136  004767  011500          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12561                                                   ;TEST IS IN WRONG SEQUENCE
12562  047142  000064                  64              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12563                                                   ;BY  (013746  000172  000207)
12564
12565  047144  005215                  INC     (R5)
12566
12567
```

```
12568                                   ;*****************************************************************
12569                                   ;TEST:200       52525 SHIFTED BY 0 = 52525  PS = 0
12570                                   ;*****************************************************************
12571
12572  047146  010767  131020          TST200: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12573  047152  010501                          MOV     R5,R1           ;SAVE R5
12574  047154  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
12575  047160  012705  052525                  MOV     #52525,%5!1     ;LOAD R5!1 WITH 52525
12576  047164  000241                          CLC
12577  047166  073527  000000                  ASHC    #0,%5           ;SHIFT R5,R5!1 BY 0
12578  047172  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12579  047176  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12580  047204  001403                          BEQ     .+10
12581  047206  004767  011430                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12582                                                                  ;THE PS IS NOT EQUAL TO 0
12583  047212  000065                          65              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12584                                                          ;BY  (013746  000172  000207)
12585
12586  047214  022705  052525                  CMP     #52525,%5       ;IS THE RESULT 52525?
12587  047220  001403                          BEQ     .+10
12588  047222  004767  011414                  JSR     PC,$HL1;SEEN AN ERROR, GO TO THE HALT ROUTINE
12589                                                                  ;R5 IS NOT EQUAL TO 52525
12590  047226  000066                          66              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12591                                                          ;BY  (013746  000172  000207)
12592
12593  047230  010105                          MOV     R1,R5           ;RESTORE R5
12594  047232  021527  000200                  CMP     (R5),#200               ;IS $TESTN = #200?
12595  047236  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12596  047240  004767  011376                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12597                                                                  ;TEST IS IN WRONG SEQUENCE
12598  047244  000067                          67              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12599                                                          ;BY  (013746  000172  000207)
12600
12601  047246  005215                          INC     (R5)
12602
12603
12604                                   ;*****************************************************************
12605                                   ;TEST:201       20010 SHIFTED BY -13. = 101  PS = 0
12606                                   ;*****************************************************************
12607
12608  047250  010767  130716          TST201: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12609  047254  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12610  047260  012701  020010                  MOV     #20010,%1!1     ;LOAD R1!1 WITH 20010
12611  047264  000241                          CLC
12612  047266  073127  177763                  ASHC    #-13.,%1        ;SHIFT R1,R1!1 BY -13.
12613  047272  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12614  047276  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS THE PS 0?
12615  047304  001403                          BEQ     .+10
12616  047306  004767  011330                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12617                                                                  ;THE PS IS NOT EQUAL TO 0
12618  047312  000070                          70              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12619                                                          ;BY  (013746  000172  000207)
12620
12621  047314  022701  000101                  CMP     #101,%1         ;IS THE RESULT 101?
12622  047320  001403                          BEQ     .+10
12623  047322  004767  011314                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```
12624
12625  047326  000071                          71              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12626                                                          ;BY  (013746  000172  000207)
12627
12628  047330  021527  000201                  CMP     (R5),#201               ;IS $TESTN = #201?
12629  047334  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12630  047336  004767  011300                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12631                                                                  ;TEST IS IN WRONG SEQUENCE
12632  047342  000072                          72              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12633                                                          ;BY  (013746  000172  000207)
12634
12635  047344  005215                          INC     (R5)
12636
12637
```

```
12638                                   ;*****************************************************************************
12639                                   ;TEST:202       -1 SHIFTED BY 16. = 0  PS = 11
12640                                   ;*****************************************************************************
12641
12642  047346  010767  130620          TST202: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12643  047352  012703  000000                  MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
12644  047356  012703  177777                  MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
12645  047362  000241                          CLC
12646  047364  073327  000020                  ASHC    #16.,%3         ;SHIFT R3,R3!1 BY 16.
12647  047370  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12648  047374  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12649  047402  001403                          BEQ     .+10
12650  047404  004767  011232                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12651                                                                  ;THE PS IS NOT EQUAL TO 11
12652  047410  000073                          73              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12653                                                          ;BY  (013746  000172  000207)
12654
12655  047412  022703  000000                  CMP     #0,%3           ;IS THE RESULT 0?
12656  047416  001403                          BEQ     .+10
12657  047420  004767  011216                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12658                                                                  ;R3 IS NOT EQUAL TO 0
12659  047424  000074                          74              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12660                                                          ;BY  (013746  000172  000207)
12661
12662  047426  021527  000202                  CMP     (R5),#202               ;IS $TESTN = #202?
12663  047432  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12664  047434  004767  011202                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12665                                                                  ;TEST IS IN WRONG SEQUENCE
12666  047440  000075                          75              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12667                                                          ;BY  (013746  000172  000207)
12668
12669  047442  005215                          INC     (R5)
12670
12671
12672                                   ;*****************************************************************************
12673                                   ;TEST:203       1 SHIFTED BY -1 = 100000  PS = 1
12674                                   ;*****************************************************************************
12675
12676  047444  010767  130522          TST203: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12677  047450  010501                          MOV     R5,R1           ;SAVE R5
12678  047452  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
12679  047456  012705  000001                  MOV     #1,%5!1         ;LOAD R5!1 WITH 1
12680  047462  000241                          CLC
12681  047464  073527  177777                  ASHC    #-1,%5          ;SHIFT R5,R5!1 BY -1
12682  047470  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12683  047474  122737  000001  042262          CMPB    #1,@#PSWORD     ;IS THE PS 1?
12684  047502  001403                          BEQ     .+10
12685  047504  004767  011132                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12686                                                                  ;THE PS IS NOT EQUAL TO 1
12687  047510  000076                          76              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12688                                                          ;BY  (013746  000172  000207)
12689
12690  047512  022705  100000                  CMP     #100000,%5      ;IS THE RESULT 100000?
12691  047516  001403                          BEQ     .+10
12692  047520  004767  011116                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12693                                                                  ;R5 IS NOT EQUAL TO 100000
```

```
12694  047524  000077                    77              ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12695                                                    ;BY  (013746  000172  000207)
12696
12697  047526  010105                    MOV     R1,R5           ;RESTORE R5
12698  047530  021527  000203            CMP     (R5),#203               ;IS $TESTN = #203?
12699  047534  001403                    BEQ     .+10            ;IF NOT THEN GO TO HLT
12700  047536  004767  011100            JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12701                                                    ;TEST IS IN WRONG SEQUENCE
12702  047542  000100                    100             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12703                                                    ;BY  (013746  000172  000207)
12704
12705  047544  005215                    INC     (R5)
12706
12707
```

```
12708                                   ;*****************************************************************************
12709                                   ;TEST:204       125252 SHIFTED BY -16. = 125252  PS = 11
12710                                   ;*****************************************************************************
12711
12712  047546  010767  130420          TST204: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12713  047552  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
12714  047556  012701  125252                  MOV     #125252,%1!1    ;LOAD R1!1 WITH 125252
12715  047562  000241                          CLC
12716  047564  073127  177760                  ASHC    #-16.,%1        ;SHIFT R1,R1!1 BY -16.
12717  047570  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12718  047574  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS THE PS 11?
12719  047602  001403                          BEQ     .+10
12720  047604  004767  011032                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12721                                                                  ;THE PS IS NOT EQUAL TO 11
12722  047610  000101                          101             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12723                                                          ;BY  (013746  000172  000207)
12724
12725  047612  022701  125252                  CMP     #125252,%1      ;IS THE RESULT 125252?
12726  047616  001403                          BEQ     .+10
12727  047620  004767  011016                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12728                                                                  ;R1 IS NOT EQUAL TO 125252
12729  047624  000102                          102             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12730                                                          ;BY  (013746  000172  000207)
12731
12732  047626  021527  000204                  CMP     (R5),#204               ;IS $TESTN = #204?
12733  047632  001403                          BEQ     .+10            ;IF NOT THEN GO TO HLT
12734  047634  004767  011002                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12735                                                                  ;TEST IS IN WRONG SEQUENCE
12736  047640  000103                          103             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12737                                                          ;BY  (013746  000172  000207)
12738
12739  047642  005215                          INC     (R5)
12740
12741
12742                                   ;*****************************************************************************
12743                                   ;TEST:205       125252 125252 SHIFTED BY 21. = 52500 000000  PS = 3
12744                                   ;*****************************************************************************
12745
12746  047644  010767  130322          TST205: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12747  047650  012702  125252                  MOV     #125252,%2      ;LOAD R2 WITH 125252
12748  047654  012703  125252                  MOV     #125252,%2!1    ;LOAD R2!1 WITH 125252
12749  047660  000241                          CLC
12750  047662  073227  000025                  ASHC    #21.,%2         ;SHIFT R2,R2!1 BY 21.
12751  047666  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12752  047672  122737  000003  042262          CMPB    #3,@#PSWORD     ;IS THE PS 3?
12753  047700  001403                          BEQ     .+10
12754  047702  004767  010734                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12755                                                                  ;THE PS IS NOT EQUAL TO 3
12756  047706  000104                          104             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12757                                                          ;BY  (013746  000172  000207)
12758
12759  047710  022702  052500                  CMP     #52500,%2       ;IS THE RESULT 52500?
12760  047714  001403                          BEQ     .+10
12761  047716  004767  010720                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12762                                                                  ;R2 IS NOT EQUAL TO 52500
12763  047722  000105                          105             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
```

```
12764                                                   ;BY  (013746  000172  000207)
12765
12766  047724  022703  000000          CMP     #000000,%2!1    ;IS THE RESULT 000000?
12767  047730  001403                  BEQ     .+10
12768  047732  004767  010704          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12769                                                           ;R2!1 IS NOT EQUAL TO 000000
12770  047736  000106                  106             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12771                                                   ;BY  (013746  000172  000207)
12772
12773  047740  021527  000205          CMP     (R5),#205               ;IS $TESTN = #205?
12774  047744  001403                  BEQ     .+10            ;IF NOT THEN GO TO HLT
12775  047746  004767  010670          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12776                                                           ;TEST IS IN WRONG SEQUENCE
12777  047752  000107                  107             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12778                                                   ;BY  (013746  000172  000207)
12779
12780  047754  005215                  INC     (R5)
12781
12782
12783
12784  047756  012702  177771          MOV     #-7,%2
12785  047762  012703  042304          MOV     #S1,%3
12786  047766  012704  042306          MOV     #S2,%4
12787
```

```
12788                                   ;*************************************************************************
12789                                   ;TEST:206       125252 125252 SHIFTED BY S1 = 177525 52525  PS = 10
12790                                   ;*************************************************************************
12791
12792  047772  010767  130174          TST206: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12793  047776  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12794  050002  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12795  050006  000241                          CLC
12796  050010  073067  172270                  ASHC    S1,%0           ;SHIFT R0,R0!1 BY S1
12797  050014  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12798  050020  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12799  050026  001403                          BEQ     .+10
12800  050030  004767  010606                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12801                                                                  ;THE PS IS NOT EQUAL TO 10
12802  050034  000110                          110             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12803                                                          ;BY  (013746  000172  000207)
12804
12805  050036  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12806  050042  001403                          BEQ     .+10
12807  050044  004767  010572                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12808                                                                  ;R0 IS NOT EQUAL TO 177525
12809  050050  000111                          111             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12810                                                          ;BY  (013746  000172  000207)
12811
12812  050052  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12813  050056  001403                          BEQ     .+10
12814  050060                          1$:
12815  050060  004767  010556                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12816                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12817  050064  000112                          112             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12818                                                          ;BY  (013746  000172  000207)
12819
12820  050066  021527  000206          ,       CMP     (R5),#206               ;IS THE $TESTN = #206?
12821  050072  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12822  050074  005215                          INC     (R5)
12823
12824
12825                                   ;*************************************************************************
12826                                   ;TEST:207       125252 125252 SHIFTED BY @S2 = 177525 52525  PS = 10
12827                                   ;*************************************************************************
12828
12829  050076  010767  130070          TST207: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12830  050102  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12831  050106  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12832  050112  000241                          CLC
12833  050114  073077  172166                  ASHC    @S2,%0          ;SHIFT R0,R0!1 BY @S2
12834  050120  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12835  050124  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12836  050132  001403                          BEQ     .+10
12837  050134  004767  010502                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12838                                                                  ;THE PS IS NOT EQUAL TO 10
12839  050140  000113                          113             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12840                                                          ;BY  (013746  000172  000207)
12841
12842  050142  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12843  050146  001403                          BEQ     .+10
```

```
12844  050150  004767  010466          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12845                                                ;R0 IS NOT EQUAL TO 177525
12846  050154  000114                  114             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12847                                                ;BY  (013746  000172  000207)
12848
12849  050156  022701  052525          CMP     #52525,%0!1     ;IS THE RESULT 52525?
12850  050162  001403                  BEQ     .+10
12851  050164                  1$:
12852  050164  004767  010452          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12853                                                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12854  050170  000115                  115             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12855                                                ;BY  (013746  000172  000207)
12856
12857  050172  021527  000207          CMP     (R5),#207               ;IS THE $TESTN = #207?
12858  050176  001372                  BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12859  050200  005215                  INC     (R5)
12860
12861
```

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

        050202  010767  127764           TST210: MOV    PC,LPADR        ;STORE ERROR LOOP ADDRESS
        050206  012700  125252                   MOV    #125252,%0              ;LOAD R0 WITH 125252
        050212  012701  125252                   MOV    #125252,%0!1    ;LOAD R0!1 WITH 125252
        050216  000241                           CLC
        050220  073037  042304                   ASHC   @#S1,%0         ;SHIFT R0,R0!1 BY @#S1
        050224  106737  042262                   MFPS   @#PSWORD        ;SAVE PS
        050230  122737  000010  042262           CMPB   #10,@#PSWORD    ;IS THE PS 10?
        050236  001403                           BEQ    .+10
        050240  004767  010376                   JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;THE PS IS NOT EQUAL TO 10
        050244  000116                           116           ;TO SCOPE, REPLACE LAST:  BEQ   .+10 (001403)
                                                               ;BY  (013746  000172  000207)

        050246  022700  177525                   CMP    #177525,%0      ;IS THE RESULT 177525?
        050252  001403                           BEQ    .+10
        050254  004767  010362                   JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;R0 IS NOT EQUAL TO 177525
        050260  000117                           117           ;TO SCOPE, REPLACE LAST:  BEQ   .+10 (001403)
                                                               ;BY  (013746  000172  000207)

        050262  022701  052525                   CMP    #52525,%0!1     ;IS THE RESULT 52525?
        050266  001403                           BEQ    .+10
        050270                           1$:
        050270  004767  010346                   JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
        050274  000120                           120           ;TO SCOPE, REPLACE LAST:  BEQ   .+10 (001403)
                                                               ;BY  (013746  000172  000207)

        050276  021527  000210                   CMP    (R5),#210               ;IS THE $TESTN = #210?
        050302  001372                           BNE    1$              ;IF NOT THEN GO TO HLT ABOVE
        050304  005215                           INC    (R5)



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

        050306  010767  127660           TST211: MOV    PC,LPADR        ;STORE ERROR LOOP ADDRESS
        050312  012700  125252                   MOV    #125252,%0              ;LOAD R0 WITH 125252
        050316  012701  125252                   MOV    #125252,%0!1    ;LOAD R0!1 WITH 125252
        050322  000241                           CLC
        050324  073013                           ASHC   (3),%0          ;SHIFT R0,R0!1 BY (3)
        050326  106737  042262                   MFPS   @#PSWORD        ;SAVE PS
        050332  122737  000010  042262           CMPB   #10,@#PSWORD    ;IS THE PS 10?
        050340  001403                           BEQ    .+10
        050342  004767  010274                   JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;THE PS IS NOT EQUAL TO 10
        050346  000121                           121           ;TO SCOPE, REPLACE LAST:  BEQ   .+10 (001403)
                                                               ;BY  (013746  000172  000207)

        050350  022700  177525                   CMP    #177525,%0      ;IS THE RESULT 177525?
        050354  001403                           BEQ    .+10
```

```
12918  050356  004767  010260          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12919                                                ;R0 IS NOT EQUAL TO 177525
12920  050362  000122                  122             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12921                                                  ;BY  (013746  000172  000207)
12922
12923  050364  022701  052525          CMP     #52525,%0!1     ;IS THE RESULT 52525?
12924  050370  001403                  BEQ     .+10
12925  050372                  1$:
12926  050372  004767  010244          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12927                                                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12928  050376  000123                  123             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12929                                                  ;BY  (013746  000172  000207)
12930
12931  050400  021527  000211          CMP     (R5),#211               ;IS THE $TESTN = #211?
12932  050404  001372                  BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12933  050406  005215                  INC     (R5)
12934
12935
```

```
12936                                   ;*****************************************************************************
12937                                   ;TEST:212       125252 125252 SHIFTED BY (3)+ = 177525 52525  PS = 10
12938                                   ;*****************************************************************************
12939
12940  050410  010767  127556          TST212: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12941  050414  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12942  050420  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12943  050424  000241                          CLC
12944  050426  073023                          ASHC    (3)+,%0         ;SHIFT R0,R0!1 BY (3)+
12945  050430  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12946  050434  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12947  050442  001403                          BEQ     .+10
12948  050444  004767  010172                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12949                                                                  ;THE PS IS NOT EQUAL TO 10
12950  050450  000124                                  124             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12951                                                                  ;BY  (013746  000172  000207)
12952
12953  050452  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12954  050456  001403                          BEQ     .+10
12955  050460  004767  010156                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12956                                                                  ;R0 IS NOT EQUAL TO 177525
12957  050464  000125                                  125             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12958                                                                  ;BY  (013746  000172  000207)
12959
12960  050466  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
12961  050472  001403                          BEQ     .+10
12962  050474                          1$:
12963  050474  004767  010142                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12964                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12965  050500  000126                                  126             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12966                                                                  ;BY  (013746  000172  000207)
12967
12968  050502  021527  000212                  CMP     (R5),#212               ;IS THE $TESTN = #212?
12969  050506  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12970  050510  005215                          INC     (R5)
12971
12972
12973                                   ;*****************************************************************************
12974                                   ;TEST:213       125252 125252 SHIFTED BY -(3) = 177525 52525  PS = 10
12975                                   ;*****************************************************************************
12976
12977  050512  010767  127454          TST213: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
12978  050516  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
12979  050522  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
12980  050526  000241                          CLC
12981  050530  073043                          ASHC    -(3),%0         ;SHIFT R0,R0!1 BY -(3)
12982  050532  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
12983  050536  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
12984  050544  001403                          BEQ     .+10
12985  050546  004767  010070                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12986                                                                  ;THE PS IS NOT EQUAL TO 10
12987  050552  000127                                  127             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12988                                                                  ;BY  (013746  000172  000207)
12989
12990  050554  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
12991  050560  001403                          BEQ     .+10
```

```
                                                H  5
12992  050562  004767  010054                        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12993                                                                 ;R0 IS NOT EQUAL TO 177525
12994  050566  000130                                130             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
12995                                                                 ;BY  (013746  000172  000207)
12996
12997  050570  022701  052525                        CMP     #52525,%0!1     ;IS THE RESULT 52525?
12998  050574  001403                                BEQ     .+10
12999  050576                                1$:
13000  050576  004767  010040                        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13001                                                                 ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13002  050602  000131                                131             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13003                                                                 ;BY  (013746  000172  000207)
13004
13005  050604  021527  000213                        CMP     (R5),#213               ;IS THE $TESTN = #213?
13006  050610  001372                                BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13007  050612  005215                                INC     (R5)
13008
13009
```

```
13010                                   ;*****************************************************************************
13011                                   ;TEST:214       125252 125252 SHIFTED BY 2(4) = 177252 125252  PS = 11
13012                                   ;*****************************************************************************
13013
13014  050614  010767  127352          TST214: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13015  050620  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13016  050624  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13017  050630  000241                          CLC
13018  050632  073064  000002                  ASHC    2(4),%0         ;SHIFT R0,R0!1 BY 2(4)
13019  050636  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13020  050642  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS THE PS 11?
13021  050650  001403                          BEQ     .+10
13022  050652  004767  007764                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13023                                                                  ;THE PS IS NOT EQUAL TO 11
13024  050656  000132                          132             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13025                                                           ;BY  (013746  000172  000207)
13026
13027  050660  022700  177252                  CMP     #177252,%0      ;IS THE RESULT 177252?
13028  050664  001403                          BEQ     .+10
13029  050666  004767  007750                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13030                                                                  ;R0 IS NOT EQUAL TO 177252
13031  050672  000133                          133             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13032                                                           ;BY  (013746  000172  000207)
13033
13034  050674  022701  125252                  CMP     #125252,%0!1    ;IS THE RESULT 125252?
13035  050700  001403                          BEQ     .+10
13036  050702                          1$:
13037  050702  004767  007734                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13038                                                                  ;R0!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
13039  050706  000134                          134             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13040                                                           ;BY  (013746  000172  000207)
13041
13042  050710  021527  000214                  CMP     (R5),#214               ;IS THE $TESTN = #214?
13043  050714  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13044  050716  005215                          INC     (R5)
13045
13046
13047                                   ;*****************************************************************************
13048                                   ;TEST:215       125252 125252 SHIFTED BY @(4) = 177525 52525  PS = 10
13049                                   ;*****************************************************************************
13050
13051  050720  010767  127246          TST215: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13052  050724  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13053  050730  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13054  050734  000241                          CLC
13055  050736  073074  000000                  ASHC    @(4),%0         ;SHIFT R0,R0!1 BY @(4)
13056  050742  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13057  050746  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13058  050754  001403                          BEQ     .+10
13059  050756  004767  007660                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13060                                                                  ;THE PS IS NOT EQUAL TO 10
13061  050762  000135                          135             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13062                                                           ;BY  (013746  000172  000207)
13063
13064  050764  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13065  050770  001403                          BEQ     .+10
```

```
13066  050772  004767  007644          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13067                                                ;R0 IS NOT EQUAL TO 177525
13068  050776  000136                  136             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13069                                                  ;BY  (013746  000172  000207)
13070
13071  051000  022701  052525          CMP     #52525,%0!1     ;IS THE RESULT 52525?
13072  051004  001403                  BEQ     .+10
13073  051006                       1$:
13074  051006  004767  007630          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13075                                               ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13076  051012  000137                  137             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13077                                                  ;BY  (013746  000172  000207)
13078
13079  051014  021527  000215          CMP     (R5),#215               ;IS THE $TESTN = #215?
13080  051020  001372                  BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13081  051022  005215                  INC     (R5)
13082
13083
```

```
13084
13085                                   ;*************************************************************************
13086                                   ;TEST:216       125252 125252 SHIFTED BY @(4)+ = 177525 52525  PS = 10
13087                                   ;*************************************************************************
13088  051024  010767  127142          TST216: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13089  051030  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13090  051034  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13091  051040  000241                          CLC
13092  051042  073034                          ASHC    @(4)+,%0                ;SHIFT R0,R0!1 BY @(4)+
13093  051044  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13094  051050  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13095  051056  001403                          BEQ     .+10
13096  051060  004767  007556                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13097                                                                  ;THE PS IS NOT EQUAL TO 10
13098  051064  000140                          140             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13099                                                          ;BY  (013746  000172  000207)
13100
13101  051066  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13102  051072  001403                          BEQ     .+10
13103  051074  004767  007542                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13104                                                                  ;R0 IS NOT EQUAL TO 177525
13105  051100  000141                          141             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13106                                                          ;BY  (013746  000172  000207)
13107
13108  051102  022701  052525                  CMP     #52525,%0!1     ;IS THE RESULT 52525?
13109  051106  001403                          BEQ     .+10
13110  051110                          1$:
13111  051110  004767  007526                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13112                                                                  ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13113  051114  000142                          142             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13114                                                          ;BY  (013746  000172  000207)
13115
13116  051116  021527  000216                  CMP     (R5),#216               ;IS THE $TESTN = #216?
13117  051122  001372                          BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13118  051124  005215                          INC     (R5)
13119
13120
13121                                   ;*************************************************************************
13122                                   ;TEST:217       125252 125252 SHIFTED BY @-(4) = 177525 52525  PS = 10
13123                                   ;*************************************************************************
13124
13125  051126  010767  127040          TST217: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13126  051132  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
13127  051136  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
13128  051142  000241                          CLC
13129  051144  073054                          ASHC    @-(4),%0                ;SHIFT R0,R0!1 BY @-(4)
13130  051146  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13131  051152  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS THE PS 10?
13132  051160  001403                          BEQ     .+10
13133  051162  004767  007454                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13134                                                                  ;THE PS IS NOT EQUAL TO 10
13135  051166  000143                          143             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13136                                                          ;BY  (013746  000172  000207)
13137
13138  051170  022700  177525                  CMP     #177525,%0      ;IS THE RESULT 177525?
13139  051174  001403                          BEQ     .+10
```

```
13140  051176  004767  007440                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13141                                                         ;R0 IS NOT EQUAL TO 177525
13142  051202  000144                        144             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13143                                                         ;BY  (013746  000172  000207)
13144
13145  051204  022701  052525                CMP     #52525,%0!1     ;IS THE RESULT 52525?
13146  051210  001403                        BEQ     .+10
13147  051212                              1$:
13148  051212  004767  007424                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13149                                                         ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13150  051216  000145                        145             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13151                                                         ;BY  (013746  000172  000207)
13152
13153  051220  021527  000217                CMP     (R5),#217               ;IS THE $TESTN = #217?
13154  051224  001372                        BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
13155  051226  005215                        INC     (R5)
13156
13157
13158
13159
13160
13161
13162
13163
13164
```

```
13165                                   ;*********************************
13166                                   ;     MUL INSTRUCTION TESTS
13167                                   ;*********************************
13168
13169
13170
13171
13172                                   ;*************************************************************************
13173                                   ;TEST:220       MUL     1 * #0 = 0 0        PS = 4
13174                                   ;*************************************************************************
13175
13176  051230  010767  126736          TST220: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13177  051234  012700  000001                  MOV     #1,%0           ;LOAD MULTIPLICAND WITH 1
13178  051240  070027  000000                  MUL     #0,%0           ;MULTIPLY 1 * #0
13179  051244  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13180  051250  122737  000004  042262          CMPB    #4,@#PSWORD     ;IS PS = 4
13181  051256  001403                          BEQ     .+10
13182  051260  004767  007356                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13183                                                                  ;PS IS WRONG
13184  051264  000146                          146             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13185                                                           ;BY  (013746  000172  000207)
13186
13187  051266  022700  000000                  CMP     #0,%0           ;IS HIGH ORDER = 0
13188  051272  001403                          BEQ     .+10
13189  051274  004767  007342                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13190                                                                  ;HIGH ORDER IS WRONG
13191  051300  000147                          147             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13192                                                           ;BY  (013746  000172  000207)
13193
13194  051302  022701  000000                  CMP     #0,%0!1         ;IS LOW ORDER = 0
13195  051306  001403                          BEQ     .+10
13196  051310                          1$:
13197  051310  004767  007326                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13198                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13199  051314  000150                          150             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13200                                                           ;BY  (013746  000172  000207)
13201
13202  051316  021527  000220                  CMP     (R5),#220
13203  051322  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13204  051324  005215                          INC     (R5)
13205
13206
```

```
13207                                   ;*****************************************************************************
13208                                   ;TEST:221       MUL     -1 * #1 = -1 -1         PS = 10
13209                                   ;*****************************************************************************
13210
13211  051326  010767  126640          TST221: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13212  051332  012700  177777                  MOV     #-1,%0          ;LOAD MULTIPLICAND WITH -1
13213  051336  070027  000001                  MUL     #1,%0           ;MULTIPLY -1 * #1
13214  051342  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13215  051346  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
13216  051354  001403                          BEQ     .+10
13217  051356  004767  007260                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13218                                                                  ;PS IS WRONG
13219  051362  000151                                  151             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13220                                                                  ;BY  (013746  000172  000207)
13221
13222  051364  022700  177777                  CMP     #-1,%0          ;IS HIGH ORDER = -1
13223  051370  001403                          BEQ     .+10
13224  051372  004767  007244                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13225                                                                  ;HIGH ORDER IS WRONG
13226  051376  000152                                  152             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13227                                                                  ;BY  (013746  000172  000207)
13228
13229  051400  022701  177777                  CMP     #-1,%0!1        ;IS LOW ORDER = -1
13230  051404  001403                          BEQ     .+10
13231  051406                          1$:
13232  051406  004767  007230                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13233                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13234  051412  000153                                  153             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13235                                                                  ;BY  (013746  000172  000207)
13236
13237  051414  021527  000221                  CMP     (R5),#221
13238  051420  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13239  051422  005215                          INC     (R5)
13240
13241
```

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

051424  010767  126542        TST222: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
051430  012702  000002                MOV     #2,%2           ;LOAD MULTIPLICAND WITH 2
051434  070227  000002                MUL     #2,%2           ;MULTIPLY 2 * #2
051440  106737  042262                MFPS    @#PSWORD        ;SAVE PS
051444  122737  000000  042262        CMPB    #0,@#PSWORD     ;IS PS = 0
051452  001403                        BEQ     .+10
051454  004767  007162                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                        ;PS IS WRONG
051460  000154                        154             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

051462  022702  000000                CMP     #0,%2           ;IS HIGH ORDER = 0
051466  001403                        BEQ     .+10
051470  004767  007146                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                        ;HIGH ORDER IS WRONG
051474  000155                        155             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

051476  022703  000004                CMP     #4,%2!1         ;IS LOW ORDER = 4
051502  001403                        BEQ     .+10
051504                          1$:
051504  004767  007132                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
051510  000156                        156             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

051512  021527  000222                CMP     (R5),#222
051516  001372                        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
051520  005215                        INC     (R5)
```

```
13277                                   ;*****************************************************************************
13278                                   ;TEST:223       MUL     1000 * #200 = 1 0       PS = 1
13279                                   ;*****************************************************************************
13280
13281  051522  010767  126444           TST223: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13282  051526  010501                           MOV     R5,R1           ;SAVE R5
13283  051530  012704  001000                   MOV     #1000,%4        ;LOAD MULTIPLICAND WITH 1000
13284  051534  070427  000200                   MUL     #200,%4         ;MULTIPLY 1000 * #200
13285  051540  106737  042262                   MFPS    @#PSWORD        ;SAVE PS
13286  051544  122737  000001  042262           CMPB    #1,@#PSWORD     ;IS PS = 1
13287  051552  001403                           BEQ     .+10
13288  051554  004767  007062                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13289                                                                   ;PS IS WRONG
13290  051560  000157                           157             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13291                                                           ;BY  (013746  000172  000207)
13292
13293  051562  022704  000001                   CMP     #1,%4           ;IS HIGH ORDER = 1
13294  051566  001403                           BEQ     .+10
13295  051570  004767  007046                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13296                                                                   ;HIGH ORDER IS WRONG
13297  051574  000160                           160             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13298                                                           ;BY  (013746  000172  000207)
13299
13300  051576  022705  000000                   CMP     #0,%4!1         ;IS LOW ORDER = 0
13301  051602  001403                           BEQ     .+10
13302  051604                          1$:
13303  051604  004767  007032                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13304                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13305  051610  000161                           161             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13306                                                           ;BY  (013746  000172  000207)
13307
13308  051612  021127  000223           CMP     (R1),#223               ;CHECK THE TEST NUMBER
13309  051616  001372                   BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13310  051620  010105                   MOV     R1,R5           ;RESTORE R5
13311  051622  005215                   INC     (R5)
13312
13313
```

```
13314                                   ;*************************************************************************
13315                                   ;TEST:224       MUL     2 * #77777 = 0 177776       PS = 1
13316                                   ;*************************************************************************
13317
13318  051624  010767  126342         TST224: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13319  051630  012700  000002                 MOV     #2,%0           ;LOAD MULTIPLICAND WITH 2
13320  051634  070027  077777                 MUL     #77777,%0       ;MULTIPLY 2 * #77777
13321  051640  106737  042262                 MFPS    @#PSWORD        ;SAVE PS
13322  051644  122737  000001  042262         CMPB    #1,@#PSWORD     ;IS PS = 1
13323  051652  001403                         BEQ     .+10
13324  051654  004767  006762                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13325                                                                  ;PS IS WRONG
13326  051660  000162                         162             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13327                                                          ;BY  (013746  000172  000207)
13328
13329  051662  022700  000000                 CMP     #0,%0           ;IS HIGH ORDER = 0
13330  051666  001403                         BEQ     .+10
13331  051670  004767  006746                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13332                                                                  ;HIGH ORDER IS WRONG
13333  051674  000163                         163             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13334                                                          ;BY  (013746  000172  000207)
13335
13336  051676  022701  177776                 CMP     #177776,%0!1    ;IS LOW ORDER = 177776
13337  051702  001403                         BEQ     .+10
13338  051704                          1$:
13339  051704  004767  006732                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13340                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13341  051710  000164                         164             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13342                                                          ;BY  (013746  000172  000207)
13343
13344  051712  021527  000224                 CMP     (R5),#224
13345  051716  001372                         BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13346  051720  005215                         INC     (R5)
13347
13348
```

```
13349                                   ;*************************************************************************
13350                                   ;TEST:225       MUL     7777 * #10 = 0 77770        PS = 0
13351                                   ;*************************************************************************
13352
13353  051722  010767  126244          TST225: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13354  051726  012702  007777                  MOV     #7777,%2        ;LOAD MULTIPLICAND WITH 7777
13355  051732  070227  000010                  MUL     #10,%2          ;MULTIPLY 7777 * #10
13356  051736  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13357  051742  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
13358  051750  001403                          BEQ     .+10
13359  051752  004767  006664                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13360                                                                  ;PS IS WRONG
13361  051756  000165                          165             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13362                                                          ;BY  (013746  000172  000207)
13363
13364  051760  022702  000000                  CMP     #0,%2           ;IS HIGH ORDER = 0
13365  051764  001403                          BEQ     .+10
13366  051766  004767  006650                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13367                                                                  ;HIGH ORDER IS WRONG
13368  051772  000166                          166             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13369                                                          ;BY  (013746  000172  000207)
13370
13371  051774  022703  077770                  CMP     #77770,%2!1     ;IS LOW ORDER = 77770
13372  052000  001403                          BEQ     .+10
13373  052002                          1$:
13374  052002  004767  006634                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13375                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13376  052006  000167                          167             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13377                                                          ;BY  (013746  000172  000207)
13378
13379  052010  021527  000225                  CMP     (R5),#225
13380  052014  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13381  052016  005215                          INC     (R5)
13382
13383
```

```
13384                                   ;*************************************************************************
13385                                   ;TEST:226       MUL     77777 * #77777 = 37777 !        PS = 1
13386                                   ;*************************************************************************
13387
13388  052020  010767  126146          TST226: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13389  052024  010501                          MOV     R5,R1           ;SAVE R5
13390  052026  012704  077777                  MOV     #77777,%4       ;LOAD MULTIPLICAND WITH 77777
13391  052032  070427  077777                  MUL     #77777,%4       ;MULTIPLY 77777 * #77777
13392  052036  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13393  052042  122737  000001  042262          CMPB    #1,@#PSWORD     ;IS PS = 1
13394  052050  001403                          BEQ     .+10
13395  052052  004767  006564                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13396                                                                  ;PS IS WRONG
13397  052056  000170                          170             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13398                                                          ;BY  (013746  000172  000207)
13399
13400  052060  022704  037777                  CMP     #37777,%4       ;IS HIGH ORDER = 37777
13401  052064  001403                          BEQ     .+10
13402  052066  004767  006550                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13403                                                                  ;HIGH ORDER IS WRONG
13404  052072  000171                          171             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13405                                                          ;BY  (013746  000172  000207)
13406
13407  052074  022705  000001                  CMP     #1,%4!1         ;IS LOW ORDER = 1
13408  052100  001403                          BEQ     .+10
13409  052102                          1$:
13410  052102  004767  006534                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13411                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13412  052106  000172                          172             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13413                                                          ;BY  (013746  000172  000207)
13414
13415  052110  021127  000226                  CMP     (R1),#226               ;CHECK THE TEST NUMBER
13416  052114  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13417  052116  010105                          MOV     R1,R5           ;RESTORE R5
13418  052120  005215                          INC     (R5)
13419
13420
```

```
13421                                   ;*****************************************************************************
13422                                   ;TEST:227       MUL      -1 * #77777 = -1 100001        PS = 10
13423                                   ;*****************************************************************************
13424
13425  052122  010767  126044          TST227: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13426  052126  012702  177777                  MOV     #-1,%2          ;LOAD MULTIPLICAND WITH -1
13427  052132  070227  077777                  MUL     #77777,%2       ;MULTIPLY -1 * #77777
13428  052136  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13429  052142  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
13430  052150  001403                          BEQ     .+10
13431  052152  004767  006464                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13432                                                                  ;PS IS WRONG
13433  052156  000173                          173             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13434                                                          ;BY  (013746  000172  000207)
13435
13436  052160  022702  177777                  CMP     #-1,%2          ;IS HIGH ORDER = -1
13437  052164  001403                          BEQ     .+10
13438  052166  004767  006450                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13439                                                                  ;HIGH ORDER IS WRONG
13440  052172  000174                          174             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13441                                                          ;BY  (013746  000172  000207)
13442
13443  052174  022703  100001                  CMP     #100001,%2!1    ;IS LOW ORDER = 100001
13444  052200  001403                          BEQ     .+10
13445  052202                          1$:
13446  052202  004767  006434                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13447                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13448  052206  000175                          175             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13449                                                          ;BY  (013746  000172  000207)
13450
13451  052210  021527  000227                  CMP     (R5),#227
13452  052214  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13453  052216  005215                          INC     (R5)
13454
13455
```

```
13456                                   ;*****************************************************************************
13457                                   ;TEST:230       MUL     -2 * #77777 = -1 2        PS = 11
13458                                   ;*****************************************************************************
13459
13460  052220  010767  125746          TST230: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13461  052224  012700  177776                  MOV     #-2,%0          ;LOAD MULTIPLICAND WITH -2
13462  052230  070027  077777                  MUL     #77777,%0       ;MULTIPLY -2 * #77777
13463  052234  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13464  052240  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13465  052246  001403                          BEQ     .+10
13466  052250  004767  006366                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13467                                                                  ;PS IS WRONG
13468  052254  000176                          176             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13469                                                          ;BY  (013746  000172  000207)
13470
13471  052256  022700  177777                  CMP     #-1,%0          ;IS HIGH ORDER = -1
13472  052262  001403                          BEQ     .+10
13473  052264  004767  006352                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13474                                                                  ;HIGH ORDER IS WRONG
13475  052270  000177                          177             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13476                                                          ;BY  (013746  000172  000207)
13477
13478  052272  022701  000002                  CMP     #2,%0!1         ;IS LOW ORDER = 2
13479  052276  001403                          BEQ     .+10
13480  052300                          1$:
13481  052300  004767  006336                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13482                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13483  052304  000200                          200             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13484                                                          ;BY  (013746  000172  000207)
13485
13486  052306  021527  000230                  CMP     (R5),#230
13487  052312  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13488  052314  005215                          INC     (R5)
13489
13490
```

```
13491                                   ;*****************************************************************************
13492                                   ;TEST:231       MUL     125252 * #2 = -1 52524        PS = 11
13493                                   ;*****************************************************************************
13494
13495  052316  010767  125650          TST231: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13496  052322  012702  125252                  MOV     #125252,%2      ;LOAD MULTIPLICAND WITH 125252
13497  052326  070227  000002                  MUL     #2,%2           ;MULTIPLY 125252 * #2
13498  052332  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13499  052336  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13500  052344  001403                          BEQ     .+10
13501  052346  004767  006270                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13502                                                                  ;PS IS WRONG
13503  052352  000201                                  201             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13504                                                                  ;BY  (013746  000172  000207)
13505
13506  052354  022702  177777                  CMP     #-1,%2          ;IS HIGH ORDER = -1
13507  052360  001403                          BEQ     .+10
13508  052362  004767  006254                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13509                                                                  ;HIGH ORDER IS WRONG
13510  052366  000202                                  202             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13511                                                                  ;BY  (013746  000172  000207)
13512
13513  052370  022703  052524                  CMP     #52524,%2!1     ;IS LOW ORDER = 52524
13514  052374  001403                          BEQ     .+10
13515  052376                          1$:
13516  052376  004767  006240                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13517                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13518  052402  000203                                  203             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13519                                                                  ;BY  (013746  000172  000207)
13520
13521  052404  021527  000231                  CMP     (R5),#231
13522  052410  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13523  052412  005215                          INC     (R5)
13524
13525
```

```
13526                                   ;*************************************************************************
13527                                   ;TEST:232       MUL     125252 * #40000 = 165252 100000        PS = 11
13528                                   ;*************************************************************************
13529
13530  052414  010767  125552          TST232: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13531  052420  010501                          MOV     R5,R1           ;SAVE R5
13532  052422  012704  125252                  MOV     #125252,%4      ;LOAD MULTIPLICAND WITH 125252
13533  052426  070427  040000                  MUL     #40000,%4       ;MULTIPLY 125252 * #40000
13534  052432  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13535  052436  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13536  052444  001403                          BEQ     .+10
13537  052446  004767  006170                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13538                                                                  ;PS IS WRONG
13539  052452  000204                          204             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13540                                                          ;BY  (013746  000172  000207)
13541
13542  052454  022704  165252                  CMP     #165252,%4      ;IS HIGH ORDER = 165252
13543  052460  001403                          BEQ     .+10
13544  052462  004767  006154                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13545                                                                  ;HIGH ORDER IS WRONG
13546  052466  000205                          205             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13547                                                          ;BY  (013746  000172  000207)
13548
13549  052470  022705  100000                  CMP     #100000,%4!1    ;IS LOW ORDER = 100000
13550  052474  001403                          BEQ     .+10
13551  052476                          1$:
13552  052476  004767  006140                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13553                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13554  052502  000206                          206             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13555                                                          ;BY  (013746  000172  000207)
13556
13557  052504  021127  000232                  CMP     (R1),#232               ;CHECK THE TEST NUMBER
13558  052510  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13559  052512  010105                          MOV     R1,R5           ;RESTORE R5
13560  052514  005215                          INC     (R5)
13561
13562
```

```
13563                                   ;*************************************************************************
13564                                   ;TEST:233       MUL     107070 * #107070 = 31222 26100          PS = 1
13565                                   ;*************************************************************************
13566
13567  052516  010767  125450          TST233: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13568  052522  012700  107070                  MOV     #107070,%0      ;LOAD MULTIPLICAND WITH 107070
13569  052526  070027  107070                  MUL     #107070,%0      ;MULTIPLY 107070 * #107070
13570  052532  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13571  052536  122737  000001  042262          CMPB    #1,@#PSWORD     ;IS PS = 1
13572  052544  001403                          BEQ     .+10
13573  052546  004767  006070                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13574                                                                  ;PS IS WRONG
13575  052552  000207                                  207             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13576                                                                  ;BY  (013746  000172  000207)
13577
13578  052554  022700  031222                  CMP     #31222,%0       ;IS HIGH ORDER = 31222
13579  052560  001403                          BEQ     .+10
13580  052562  004767  006054                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13581                                                                  ;HIGH ORDER IS WRONG
13582  052566  000210                                  210             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13583                                                                  ;BY  (013746  000172  000207)
13584
13585  052570  022701  026100                  CMP     #26100,%0!1     ;IS LOW ORDER = 26100
13586  052574  001403                          BEQ     .+10
13587  052576                          1$:
13588  052576  004767  006040                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13589                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13590  052602  000211                                  211             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13591                                                                  ;BY  (013746  000172  000207)
13592
13593  052604  021527  000233                  CMP     (R5),#233
13594  052610  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13595  052612  005215                          INC     (R5)
13596
13597
```

```
13598                                   ;*****************************************************************************
13599                                   ;TEST:234       MUL     -1 * #1 = -1 -1         PS = 10
13600                                   ;*****************************************************************************
13601
13602  052614  010767  125352          TST234: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13603  052620  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
13604  052624  070127  000001                  MUL     #1,%1           ;MULTIPLY -1 * #1
13605  052630  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13606  052634  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
13607  052642  001403                          BEQ     .+10
13608  052644  004767  005772                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13609                                                                  ;PS IS WRONG
13610  052650  000212                          212             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13611                                                          ;BY  (013746  000172  000207)
13612
13613  052652  022701  177777                  CMP     #-1,%1          ;IS HIGH ORDER = -1
13614  052656  001403                          BEQ     .+10
13615  052660  004767  005756                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13616                                                                  ;HIGH ORDER IS WRONG
13617  052664  000213                          213             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13618                                                          ;BY  (013746  000172  000207)
13619
13620  052666  022701  177777                  CMP     #-1,%1!1        ;IS LOW ORDER = -1
13621  052672  001403                          BEQ     .+10
13622  052674                          1$:
13623  052674  004767  005742                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13624                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13625  052700  000214                          214             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13626                                                          ;BY  (013746  000172  000207)
13627
13628  052702  021527  000234                  CMP     (R5),#234
13629  052706  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13630  052710  005215                          INC     (R5)
13631
13632
```

```
13633                                   ;*************************************************************************
13634                                   ;TEST:235       MUL     -1 * #0 = 0 0         PS = 4
13635                                   ;*************************************************************************
13636
13637  052712  010767  125254          TST235: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13638  052716  012703  177777                  MOV     #-1,%3          ;LOAD MULTIPLICAND WITH -1
13639  052722  070327  000000                  MUL     #0,%3           ;MULTIPLY -1 * #0
13640  052726  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13641  052732  122737  000004  042262          CMPB    #4,@#PSWORD     ;IS PS = 4
13642  052740  001403                          BEQ     .+10
13643  052742  004767  005674                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13644                                                                  ;PS IS WRONG
13645  052746  000215                                  215             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13646                                                                  ;BY  (013746  000172  000207)
13647
13648  052750  022703  000000                  CMP     #0,%3           ;IS HIGH ORDER = 0
13649  052754  001403                          BEQ     .+10
13650  052756  004767  005660                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13651                                                                  ;HIGH ORDER IS WRONG
13652  052762  000216                                  216             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13653                                                                  ;BY  (013746  000172  000207)
13654
13655  052764  022703  000000                  CMP     #0,%3!1         ;IS LOW ORDER = 0
13656  052770  001403                          BEQ     .+10
13657  052772                          1$:
13658  052772  004767  005644                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13659                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13660  052776  000217                                  217             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13661                                                                  ;BY  (013746  000172  000207)
13662
13663  053000  021527  000235                  CMP     (R5),#235
13664  053004  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13665  053006  005215                          INC     (R5)
13666
13667
```

```
13668                                   ;*************************************************************************
13669                                   ;TEST:236       MUL     77777 * #100000 = 100000 100000        PS = 11
13670                                   ;*************************************************************************
13671
13672  053010  010767  125156          TST236: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13673  053014  010501                          MOV     R5,R1           ;SAVE R5
13674  053016  012705  077777                  MOV     #77777,%5       ;LOAD MULTIPLICAND WITH 77777
13675  053022  070527  100000                  MUL     #100000,%5      ;MULTIPLY 77777 * #100000
13676  053026  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13677  053032  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13678  053040  001403                          BEQ     .+10
13679  053042  004767  005574                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13680                                                                  ;PS IS WRONG
13681  053046  000220                          220             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13682                                                          ;BY  (013746  000172  000207)
13683
13684  053050  022705  100000                  CMP     #100000,%5      ;IS HIGH ORDER = 100000
13685  053054  001403                          BEQ     .+10
13686  053056  004767  005560                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13687                                                                  ;HIGH ORDER IS WRONG
13688  053062  000221                          221             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13689                                                          ;BY  (013746  000172  000207)
13690
13691  053064  022705  100000                  CMP     #100000,%5!1    ;IS LOW ORDER = 100000
13692  053070  001403                          BEQ     .+10
13693  053072                          1$:
13694  053072  004767  005544                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13695                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13696  053076  000222                          222             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13697                                                          ;BY  (013746  000172  000207)
13698
13699  053100  021127  000236                  CMP     (R1),#236               ;CHECK THE TEST NUMBER
13700  053104  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13701  053106  010105                          MOV     R1,R5           ;RESTORE R5
13702  053110  005215                          INC     (R5)
13703
13704
```

```
13705                                   ;*************************************************************************
13706                                   ;TEST:237       MUL     -1 * #77777 = 100001 100001        PS = 10
13707                                   ;*************************************************************************
13708
13709  053112  010767  125054          TST237: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13710  053116  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
13711  053122  070127  077777                  MUL     #77777,%1       ;MULTIPLY -1 * #77777
13712  053126  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13713  053132  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
13714  053140  001403                          BEQ     .+10
13715  053142  004767  005474                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13716                                                                  ;PS IS WRONG
13717  053146  000223                          223             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13718                                                          ;BY  (013746  000172  000207)
13719
13720  053150  022701  100001                  CMP     #100001,%1      ;IS HIGH ORDER = 100001
13721  053154  001403                          BEQ     .+10
13722  053156  004767  005460                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13723                                                                  ;HIGH ORDER IS WRONG
13724  053162  000224                          224             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13725                                                          ;BY  (013746  000172  000207)
13726
13727  053164  022701  100001                  CMP     #100001,%1!1    ;IS LOW ORDER = 100001
13728  053170  001403                          BEQ     .+10
13729  053172                          1$:
13730  053172  004767  005444                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13731                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13732  053176  000225                          225             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13733                                                          ;BY  (013746  000172  000207)
13734
13735  053200  021527  000237                  CMP     (R5),#237
13736  053204  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13737  053206  005215                          INC     (R5)
13738
13739
```

```
13740                                   ;*************************************************************************
13741                                   ;TEST:240       MUL     77777 * #77777 = 1 1        PS = 1
13742                                   ;*************************************************************************
13743
13744  053210  010767  124756          TST240: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13745  053214  012703  077777                  MOV     #77777,%3       ;LOAD MULTIPLICAND WITH 77777
13746  053220  070327  077777                  MUL     #77777,%3       ;MULTIPLY 77777 * #77777
13747  053224  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13748  053230  122737  000001  042262          CMPB    #1,@#PSWORD     ;IS PS = 1
13749  053236  001403                          BEQ     .+10
13750  053240  004767  005376                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13751                                                                  ;PS IS WRONG
13752  053244  000226                                  226             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13753                                                                  ;BY  (013746  000172  000207)
13754
13755  053246  022703  000001                  CMP     #1,%3           ;IS HIGH ORDER = 1
13756  053252  001403                          BEQ     .+10
13757  053254  004767  005362                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13758                                                                  ;HIGH ORDER IS WRONG
13759  053260  000227                                  227             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13760                                                                  ;BY  (013746  000172  000207)
13761
13762  053262  022703  000001                  CMP     #1,%3!1         ;IS LOW ORDER = 1
13763  053266  001403                          BEQ     .+10
13764  053270                          1$:
13765  053270  004767  005346                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13766                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13767  053274  000230                                  230             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13768                                                                  ;BY  (013746  000172  000207)
13769
13770  053276  021527  000240                  CMP     (R5),#240
13771  053302  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13772  053304  005215                          INC     (R5)
13773
13774
```

```
13775                                   ;*************************************************************************
13776                                   ;TEST:241       MUL     2 * #2 = 4 4        PS = 0
13777                                   ;*************************************************************************
13778
13779  053306  010767  124660          TST241: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13780  053312  010501                          MOV     R5,R1           ;SAVE R5
13781  053314  012705  000002                  MOV     #2,%5           ;LOAD MULTIPLICAND WITH 2
13782  053320  070527  000002                  MUL     #2,%5           ;MULTIPLY 2 * #2
13783  053324  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13784  053330  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
13785  053336  001403                          BEQ     .+10
13786  053340  004767  005276                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13787                                                                  ;PS IS WRONG
13788  053344  000231                          231             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13789                                                          ;BY  (013746  000172  000207)
13790
13791  053346  022705  000004                  CMP     #4,%5           ;IS HIGH ORDER = 4
13792  053352  001403                          BEQ     .+10
13793  053354  004767  005262                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13794                                                                  ;HIGH ORDER IS WRONG
13795  053360  000232                          232             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13796                                                          ;BY  (013746  000172  000207)
13797
13798  053362  022705  000004                  CMP     #4,%5!1         ;IS LOW ORDER = 4
13799  053366  001403                          BEQ     .+10
13800  053370                          1$:
13801  053370  004767  005246                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13802                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13803  053374  000233                          233             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13804                                                          ;BY  (013746  000172  000207)
13805
13806  053376  021127  000241                  CMP     (R1),#241               ;CHECK THE TEST NUMBER
13807  053402  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13808  053404  010105                          MOV     R1,R5           ;RESTORE R5
13809  053406  005215                          INC     (R5)
13810
13811
```

```
13812  053410  012702  040000              MOV     #40000,%2
13813  053414  012703  042314              MOV     #S5,%3
13814  053420  012704  042316              MOV     #S6,%4
13815
13816                                      ;*************************************************************************
13817                                      ;TEST:242       MUL      125252 * S5 = 165252 100000        PS = 11
13818                                      ;*************************************************************************
13819
13820  053424  010767  124542        TST242: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13821  053430  012700  125252                MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13822  053434  070067  166654                MUL     S5,%0           ;MULTIPLY 125252 * S5
13823  053440  106737  042262                MFPS    @#PSWORD        ;SAVE PS
13824  053444  122737  000011  042262        CMPB    #11,@#PSWORD    ;IS PS = 11
13825  053452  001403                        BEQ     .+10
13826  053454  004767  005162                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13827                                                                  ;PS IS WRONG
13828  053460  000234                                234             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13829                                                                  ;BY  (013746  000172  000207)
13830
13831  053462  022700  165252                CMP     #165252,%0      ;IS HIGH ORDER = 165252
13832  053466  001403                        BEQ     .+10
13833  053470  004767  005146                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13834                                                                  ;HIGH ORDER IS WRONG
13835  053474  000235                                235             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13836                                                                  ;BY  (013746  000172  000207)
13837
13838  053476  022701  100000                CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13839  053502  001403                        BEQ     .+10
13840  053504                          1$:
13841  053504  004767  005132                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13842                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13843  053510  000236                                236             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13844                                                                  ;BY  (013746  000172  000207)
13845
13846  053512  021527  000242                CMP     (R5),#242
13847  053516  001372                        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13848  053520  005215                        INC     (R5)
13849
13850
```

```
13851                                   ;*************************************************************************
13852                                   ;TEST:243       MUL     125252 * @S6 = 165252 100000        PS = 11
13853                                   ;*************************************************************************
13854
13855  053522  010767  124444          TST243: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13856  053526  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13857  053532  070077  166560                  MUL     @S6,%0          ;MULTIPLY 125252 * @S6
13858  053536  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13859  053542  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13860  053550  001403                          BEQ     .+10
13861  053552  004767  005064                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13862                                                                  ;PS IS WRONG
13863  053556  000237                                  237             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13864                                                                  ;BY  (013746  000172  000207)
13865
13866  053560  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13867  053564  001403                          BEQ     .+10
13868  053566  004767  005050                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13869                                                                  ;HIGH ORDER IS WRONG
13870  053572  000240                                  240             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13871                                                                  ;BY  (013746  000172  000207)
13872
13873  053574  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13874  053600  001403                          BEQ     .+10
13875  053602                          1$:
13876  053602  004767  005034                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13877                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13878  053606  000241                                  241             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13879                                                                  ;BY  (013746  000172  000207)
13880
13881  053610  021527  000243                  CMP     (R5),#243
13882  053614  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13883  053616  005215                          INC     (R5)
13884
13885
```

```
13886                                   ;*************************************************************************
13887                                   ;TEST:244       MUL     125252 * @#S5 = 165252 100000        PS = 11
13888                                   ;*************************************************************************
13889
13890  053620  010767  124346          TST244: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13891  053624  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13892  053630  070037  042314                  MUL     @#S5,%0         ;MULTIPLY 125252 * @#S5
13893  053634  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13894  053640  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13895  053646  001403                          BEQ     .+10
13896  053650  004767  004766                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13897                                                                  ;PS IS WRONG
13898  053654  000242                          242             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13899                                                          ;BY  (013746  000172  000207)
13900
13901  053656  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13902  053662  001403                          BEQ     .+10
13903  053664  004767  004752                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13904                                                                  ;HIGH ORDER IS WRONG
13905  053670  000243                          243             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13906                                                          ;BY  (013746  000172  000207)
13907
13908  053672  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13909  053676  001403                          BEQ     .+10
13910  053700                          1$:
13911  053700  004767  004736                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13912                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13913  053704  000244                          244             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13914                                                          ;BY  (013746  000172  000207)
13915
13916  053706  021527  000244                  CMP     (R5),#244
13917  053712  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13918  053714  005215                          INC     (R5)
13919
13920
```

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

053716  010767  124250                          TST245: MOV     PC,LPADR           ;STORE ERROR LOOP ADDRESS
053722  012700  125252                                  MOV     #125252,%0         ;LOAD MULTIPLICAND WITH 125252
053726  070002                                          MUL     %2,%0              ;MULTIPLY 125252 * %2
053730  106737  042262                                  MFPS    @#PSWORD           ;SAVE PS
053734  122737  000011  042262                          CMPB    #11,@#PSWORD       ;IS PS = 11
053742  001403                                          BEQ     .+10
053744  004767  004672                                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                                   ;PS IS WRONG
053750  000245                                          245             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                                        ;BY  (013746  000172  000207)

053752  022700  165252                                  CMP     #165252,%0         ;IS HIGH ORDER = 165252
053756  001403                                          BEQ     .+10
053760  004767  004656                                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                                   ;HIGH ORDER IS WRONG
053764  000246                                          246             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                                        ;BY  (013746  000172  000207)

053766  022701  100000                                  CMP     #100000,%0!1       ;IS LOW ORDER = 100000
053772  001403                                          BEQ     .+10
053774                                          1$:
053774  004767  004642                                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                                   ;LOW ORDER IS WRONG OR WRONG SEQUENCE
054000  000247                                          247             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                                        ;BY  (013746  000172  000207)

054002  021527  000245                                  CMP     (R5),#245
054006  001372                                          BNE     1$                 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
054010  005215                                          INC     (R5)
```

```
13956                                   ;*************************************************************************
13957                                   ;TEST:246       MUL     125252 * (3)+ = 165252 100000        PS = 11
13958                                   ;*************************************************************************
13959
13960  054012  010767  124154          TST246: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
13961  054016  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13962  054022  070023                          MUL     (3)+,%0         ;MULTIPLY 125252 * (3)+
13963  054024  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
13964  054030  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
13965  054036  001403                          BEQ     .+10
13966  054040  004767  004576                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13967                                                                  ;PS IS WRONG
13968  054044  000250                          250             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13969                                                          ;BY  (013746  000172  000207)
13970
13971  054046  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
13972  054052  001403                          BEQ     .+10
13973  054054  004767  004562                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13974                                                                  ;HIGH ORDER IS WRONG
13975  054060  000251                          251             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13976                                                          ;BY  (013746  000172  000207)
13977
13978  054062  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
13979  054066  001403                          BEQ     .+10
13980  054070                          1$:
13981  054070  004767  004546                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13982                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13983  054074  000252                          252             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
13984                                                          ;BY  (013746  000172  000207)
13985
13986  054076  021527  000246                  CMP     (R5),#246
13987  054102  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13988  054104  005215                          INC     (R5)
13989
13990
```

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

054106  010767  124060        TST247: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
054112  012700  125252                MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
054116  070043                        MUL     -(3),%0         ;MULTIPLY 125252 * -(3)
054120  106737  042262                MFPS    @#PSWORD        ;SAVE PS
054124  122737  000011  042262        CMPB    #11,@#PSWORD    ;IS PS = 11
054132  001403                        BEQ     .+10
054134  004767  004502                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;PS IS WRONG
054140  000253                        253             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054142  022700  165252                CMP     #165252,%0      ;IS HIGH ORDER = 165252
054146  001403                        BEQ     .+10
054150  004767  004466                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;HIGH ORDER IS WRONG
054154  000254                        254             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054156  022701  100000                CMP     #100000,%0+1    ;IS LOW ORDER = 100000
054162  001403                        BEQ     .+10
054164                          1$:
054164  004767  004452                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
054170  000255                        255             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054172  021527  000247                CMP     (R5),#247
054176  001372                        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
054200  005215                        INC     (R5)
```

```
14026                                   ;*****************************************************************************
14027                                   ;TEST:250       MUL     125252 * 2(4) = 165252 100000        PS = 11
14028                                   ;*****************************************************************************
14029
14030  054202  010767  123764          TST250: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14031  054206  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14032  054212  070064  000002                  MUL     2(4),%0         ;MULTIPLY 125252 * 2(4)
14033  054216  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14034  054222  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
14035  054230  001403                          BEQ     .+10
14036  054232  004767  004404                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14037                                                                  ;PS IS WRONG
14038  054236  000256                          256             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14039                                                          ;BY  (013746  000172  000207)
14040
14041  054240  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
14042  054244  001403                          BEQ     .+10
14043  054246  004767  004370                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14044                                                                  ;HIGH ORDER IS WRONG
14045  054252  000257                          257             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14046                                                          ;BY  (013746  000172  000207)
14047
14048  054254  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14049  054260  001403                          BEQ     .+10
14050  054262                          1$:
14051  054262  004767  004354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14052                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14053  054266  000260                          260             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14054                                                          ;BY  (013746  000172  000207)
14055
14056  054270  021527  000250                  CMP     (R5),#250
14057  054274  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14058  054276  005215                          INC     (R5)
14059
14060
```

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

054300  010767  123666          TST251: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
054304  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
054310  070074  000000                  MUL     @(4),%0         ;MULTIPLY 125252 * @(4)
054314  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
054320  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
054326  001403                          BEQ     .+10
054330  004767  004306                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;PS IS WRONG
054334  000261                          261             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054336  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
054342  001403                          BEQ     .+10
054344  004767  004272                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;HIGH ORDER IS WRONG
054350  000262                          262             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054352  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
054356  001403                          BEQ     .+10
054360                          1$:
054360  004767  004256                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
054364  000263                          263             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
                                                        ;BY  (013746  000172  000207)

054366  021527  000251                  CMP     (R5),#251
054372  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
054374  005215                          INC     (R5)
```

```
14096                                   ;*****************************************************************************
14097                                   ;TEST:252       MUL     125252 * @(4)+ = 165252 100000        PS = 11
14098                                   ;*****************************************************************************
14099
14100  054376  010767  123570          TST252: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14101  054402  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14102  054406  070034                          MUL     @(4)+,%0        ;MULTIPLY 125252 * @(4)+
14103  054410  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14104  054414  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
14105  054422  001403                          BEQ     .+10
14106  054424  004767  004212                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14107                                                                  ;PS IS WRONG
14108  054430  000264                          264             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14109                                                          ;BY  (013746  000172  000207)
14110
14111  054432  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
14112  054436  001403                          BEQ     .+10
14113  054440  004767  004176                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14114                                                                  ;HIGH ORDER IS WRONG
14115  054444  000265                          265             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14116                                                          ;BY  (013746  000172  000207)
14117
14118  054446  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14119  054452  001403                          BEQ     .+10
!4120  054454                           1$:
14121  054454  004767  004162                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14122                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14123  054460  000266                          266             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14124                                                          ;BY  (013746  000172  000207)
14125
14126  054462  021527  000252                  CMP     (R5),#252
14127  054466  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14128  054470  005215                          INC     (R5)
14129
14130
```

```
14131                                   ;*************************************************************************
14132                                   ;TEST:253       MUL     125252 * @-(4) = 165252 100000        PS = 11
14133                                   ;*************************************************************************
14134
14135  054472  010767  123474          TST253: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14136  054476  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14137  054502  070054                          MUL     @-(4),%0        ;MULTIPLY 125252 * @-(4)
14138  054504  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14139  054510  122737  000011  042262          CMPB    #11,@#PSWORD    ;IS PS = 11
14140  054516  001403                          BEQ     .+10
14141  054520  004767  004116                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14142                                                                  ;PS IS WRONG
14143  054524  000267                          267             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14144                                                          ;BY  (013746  000172  000207)
14145
14146  054526  022700  165252                  CMP     #165252,%0      ;IS HIGH ORDER = 165252
14147  054532  001403                          BEQ     .+10
14148  054534  004767  004102                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14149                                                                  ;HIGH ORDER IS WRONG
14150  054540  000270                          270             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14151                                                          ;BY  (013746  000172  000207)
14152
14153  054542  022701  100000                  CMP     #100000,%0!1    ;IS LOW ORDER = 100000
14154  054546  001403                          BEQ     .+10
14155  054550                          1$:
14156  054550  004767  004066                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14157                                                                  ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14158  054554  000271                          271             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14159                                                          ;BY  (013746  000172  000207)
14160
14161  054556  021527  000253                  CMP     (R5),#253
14162  054562  001372                          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14163  054564  005215                          INC     (R5)
14164
14165
```

```
14166                                   ;*********************************
14167                                   ;     DIV INSTRUCTION TESTS
14168                                   ;*********************************
14169
14170
14171
14172                           ;*************************************************************************
14173                           ;TEST:254       DIV     0 4 / #2 = 2        REM = 0       PS = 0
14174                           ;*************************************************************************
14175
14176  054566  010767  123400        1ST254: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14177  054572  012700  000000                MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14178  054576  012701  000004                MOV     #4,%0+1         ;LOAD LOW ORDER WITH 4
14179  054602  071027  000002                DIV     #2,%0           ;DIVIDE BY #2
14180  054606  106737  042262                MFPS    @#PSWORD        ;SAVE PS
14181
14182  054612  122737  000000  042262        CMPB    #0,@#PSWORD     ;IS PS = 0
14183  054620  001403                        BEQ     .+10
14184  054622  004767  004014                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14185                                                                 ;PS IS WRONG
14186  054626  000272                                272             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14187                                                                 ;BY  (013746  000172  000207)
14188
14189
14190  054630  022700  000002                CMP     #2,%0           ;IS QUOTIENT = 2
14191  054634  001403                        BEQ     .+10
14192  054636  004767  004000                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14193                                                                 ;QUOTIENT IS WRONG
14194  054642  000273                                273             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14195                                                                 ;BY  (013746  000172  000207)
14196
14197
14198  054644  022701  000000                CMP     #0,%0+1         ;IS REMAINDER = 0
14199  054650  001403                        BEQ     .+10
14200  054652  004767  003764                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14201                                                                 ;WRONG REMAINDER
14202  054656  000274                                274             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14203                                                                 ;BY  (013746  000172  000207)
14204
14205  054660  021527  000254                CMP     (R5),#254
14206  054664  001403                        BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14207  054666  004767  003750                JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14208                                                                 ;TEST IS IN WRONG SEQUENCE
14209  054672  000275                                275             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14210                                                                 ;BY  (013746  000172  000207)
14211
14212  054674  005215                        INC     (R5)
14213
```

```
14214                                   ;*****************************************************************************
14215                                   ;TEST:255      DIV     -1 -9. / #3 = -3       REM = 0      PS = 10
14216                                   ;*****************************************************************************
14217
14218  054676  010767  123270          TST255: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14219  054702  012702  177777                  MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
14220  054706  012703  177767                  MOV     #-9.,%2+1       ;LOAD LOW ORDER WITH -9.
14221  054712  071227  000003                  DIV     #3,%2           ;DIVIDE BY #3
14222  054716  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14223
14224  054722  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
14225  054730  001403                          BEQ     .+10
14226  054732  004767  003704                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14227                                                                  ;PS IS WRONG
14228  054736  000276                          276             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14229                                                          ;BY  (013746  000172  000207)
14230
14231
14232  054740  022702  177775                  CMP     #-3,%2          ;IS QUOTIENT = -3
14233  054744  001403                          BEQ     .+10
14234  054746  004767  003670                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14235                                                                  ;QUOTIENT IS WRONG
14236  054752  000277                          277             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14237                                                          ;BY  (013746  000172  000207)
14238
14239
14240  054754  022703  000000                  CMP     #0,%2+1         ;IS REMAINDER = 0
14241  054760  001403                          BEQ     .+10
14242  054762  004767  003654                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14243                                                                  ;WRONG REMAINDER
14244  054766  000300                          300             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14245                                                          ;BY  (013746  000172  000207)
14246
14247  054770  021527  000255                  CMP     (R5),#255
14248  054774  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14249  054776  004767  003640                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14250                                                                  ;TEST IS IN WRONG SEQUENCE
14251  055002  000301                          301             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14252                                                          ;BY  (013746  000172  000207)
14253
14254  055004  005215                          INC     (R5)
14255
```

```
14256                                   ;*************************************************************************
14257                                   ;TEST:256      DIV     0 9. / #2 = 4       REM = 1      PS = 0
14258                                   ;*************************************************************************
14259
14260  055006  010767  123160          TST256: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14261  055012  010501                          MOV     R5,R1           ;SAVE R5
14262  055014  012704  000000                  MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14263  055020  012705  000011                  MOV     #9.,%4+1        ;LOAD LOW ORDER WITH 9.
14264  055024  071427  000002                  DIV     #2,%4           ;DIVIDE BY #2
14265  055030  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14266
14267  055034  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14268  055042  001403                          BEQ     .+10
14269  055044  004767  003572                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14270                                                                  ;PS IS WRONG
14271  055050  000302                          302             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14272                                                          ;BY  (013746  000172  000207)
14273
14274
14275  055052  022704  000004                  CMP     #4,%4           ;IS QUOTIENT = 4
14276  055056  001403                          BEQ     .+10
14277  055060  004767  003556                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14278                                                                  ;QUOTIENT IS WRONG
14279  055064  000303                          303             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14280                                                          ;BY  (013746  000172  000207)
14281
14282
14283  055066  022705  000001                  CMP     #1,%4+1         ;IS REMAINDER = 1
14284  055072  001403                          BEQ     .+10
14285  055074  004767  003542                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14286                                                                  ;WRONG REMAINDER
14287  055100  000304                          304             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14288                                                          ;BY  (013746  000172  000207)
14289
14290  055102  010105                  MOV     R1,R5           ;RESTORE R5
14291  055104  021527  000256          CMP     (R5),#256
14292  055110  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14293  055112  004767  003524          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14294                                                          ;TEST IS IN WRONG SEQUENCE
14295  055116  000305                  305             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14296                                                  ;BY  (013746  000172  000207)
14297
14298  055120  005215                  INC     (R5)
14299
```

```
14300                                   ;*************************************************************************
14301                                   ;TEST:257      DIV     -1 -9. / #2 = -4       REM = -1      PS = 10
14302                                   ;*************************************************************************
14303
14304  055122  010767  123044          TST257: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14305  055126  012700  177777                  MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
14306  055132  012701  177767                  MOV     #-9.,%0+1       ;LOAD LOW ORDER WITH -9.
14307  055136  071027  000002                  DIV     #2,%0           ;DIVIDE BY #2
14308  055142  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14309
14310  055146  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
14311  055154  001403                          BEQ     .+10
14312  055156  004767  003460                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14313                                                                  ;PS IS WRONG
14314  055162  000306                          306             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14315                                                          ;BY  (013746  000172  000207)
14316
14317
14318  055164  022700  177774                  CMP     #-4,%0          ;IS QUOTIENT = -4
14319  055170  001403                          BEQ     .+10
14320  055172  004767  003444                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14321                                                                  ;QUOTIENT IS WRONG
14322  055176  000307                          307             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14323                                                          ;BY  (013746  000172  000207)
14324
14325
14326  055200  022701  177777                  CMP     #-1,%0+1        ;IS REMAINDER = -1
14327  055204  001403                          BEQ     .+10
14328  055206  004767  003430                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14329                                                                  ;WRONG REMAINDER
14330  055212  000310                          310             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14331                                                          ;BY  (013746  000172  000207)
14332
14333  055214  021527  000257                  CMP     (R5),#257
14334  055220  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14335  055222  004767  003414                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14336                                                                  ;TEST IS IN WRONG SEQUENCE
14337  055226  000311                          311             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14338                                                          ;BY  (013746  000172  000207)
14339
14340  055230  005215                          INC     (R5)
14341
```

```
14342                                   ;*************************************************************************
14343                                   ;TEST:260       DIV     0 2 / #-3 = 0       REM = 2       PS = 4
14344                                   ;*************************************************************************
14345
14346  055232  010767  122734          TST260: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14347  055236  012702  000000                  MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
14348  055242  012703  000002                  MOV     #2,%2+1         ;LOAD LOW ORDER WITH 2
14349  055246  071227  177775                  DIV     #-3,%2          ;DIVIDE BY #-3
14350  055252  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14351
14352  055256  122737  000004  042262          CMPB    #4,@#PSWORD     ;IS PS = 4
14353  055264  001403                          BEQ     .+10
14354  055266  004767  003350                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14355                                                                  ;PS IS WRONG
14356  055272  000312                          312             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14357                                                          ;BY  (013746  000172  000207)
14358
14359
14360  055274  022702  000000                  CMP     #0,%2           ;IS QUOTIENT = 0
14361  055300  001403                          BEQ     .+10
14362  055302  004767  003334                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14363                                                                  ;QUOTIENT IS WRONG
14364  055306  000313                          313             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14365                                                          ;BY  (013746  000172  000207)
14366
14367
14368  055310  022703  000002                  CMP     #2,%2+1         ;IS REMAINDER = 2
14369  055314  001403                          BEQ     .+10
14370  055316  004767  003320                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14371                                                                  ;WRONG REMAINDER
14372  055322  000314                          314             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14373                                                          ;BY  (013746  000172  000207)
14374
14375  055324  021527  000260                  CMP     (R5),#260
14376  055330  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14377  055332  004767  003304                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14378                                                                  ;TEST IS IN WRONG SEQUENCE
14379  055336  000315                          315             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14380                                                          ;BY  (013746  000172  000207)
14381
14382  055340  005215                          INC     (R5)
14383
```

```
14384                                   ;*************************************************************************
14385                                   ;TEST:261      DIV     -1 -2 / #3 = 0       REM = -2      PS = 4
14386                                   ;*************************************************************************
14387
14388  055342  010767  122624          TST261: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14389  055346  010501                          MOV     R5,R1           ;SAVE R5
14390  055350  012704  177777                  MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
14391  055354  012705  177776                  MOV     #-2,%4+1        ;LOAD LOW ORDER WITH -2
14392  055360  071427  000003                  DIV     #3,%4           ;DIVIDE BY #3
14393  055364  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14394
14395  055370  122737  000004  042262          CMPB    #4,@#PSWORD     ;IS PS = 4
14396  055376  001403                          BEQ     .+10
14397  055400  004767  003236                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14398                                                                  ;PS IS WRONG
14399  055404  000316                          316             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14400                                                          ;BY  (013746  000172  000207)
14401
14402
14403  055406  022704  000000                  CMP     #0,%4           ;IS QUOTIENT = 0
14404  055412  001403                          BEQ     .+10
14405  055414  004767  003222                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14406                                                                  ;QUOTIENT IS WRONG
14407  055420  000317                          317             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14408                                                          ;BY  (013746  000172  000207)
14409
14410
14411  055422  022705  177776                  CMP     #-2,%4+1        ;IS REMAINDER = -2
14412  055426  001403                          BEQ     .+10
14413  055430  004767  003206                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14414                                                                  ;WRONG REMAINDER
14415  055434  000320                          320             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14416                                                          ;BY  (013746  000172  000207)
14417
14418  055436  010105                  MOV     R1,R5           ;RESTORE R5
14419  055440  021527  000261          CMP     (R5),#261
14420  055444  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14421  055446  004767  003170          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14422                                                          ;TEST IS IN WRONG SEQUENCE
14423  055452  000321                  321             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14424                                                  ;BY  (013746  000172  000207)
14425
14426  055454  005215                  INC     (R5)
14427
```

```
                                                H 8
14428                                   ;*************************************************************************
14429                                   ;TEST:262      DIV     -1 -1 / #1 = -1       REM = 0      PS = 10
14430                                   ;*************************************************************************
14431
14432  055456  010767  122510          TST262: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14433  055462  012700  177777                  MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
14434  055466  012701  177777                  MOV     #-1,%0+1        ;LOAD LOW ORDER WITH -1
14435  055472  071027  000001                  DIV     #1,%0           ;DIVIDE BY #1
14436  055476  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14437
14438  055502  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
14439  055510  001403                          BEQ     .+10
14440  055512  004767  003124                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14441                                                                  ;PS IS WRONG
14442  055516  000322                          322             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14443                                                          ;BY  (013746  000172  000207)
14444
14445
14446  055520  022700  177777                  CMP     #-1,%0          ;IS QUOTIENT = -1
14447  055524  001403                          BEQ     .+10
14448  055526  004767  003110                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14449                                                                  ;QUOTIENT IS WRONG
14450  055532  000323                          323             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14451                                                          ;BY  (013746  000172  000207)
14452
14453
14454  055534  022701  000000                  CMP     #0,%0+1         ;IS REMAINDER = 0
14455  055540  001403                          BEQ     .+10
14456  055542  004767  003074                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14457                                                                  ;WRONG REMAINDER
14458  055546  000324                          324             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14459                                                          ;BY  (013746  000172  000207)
14460
14461  055550  021527  000262                  CMP     (R5),#262
14462  055554  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14463  055556  004767  003060                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14464                                                                  ;TEST IS IN WRONG SEQUENCE
14465  055562  000325                          325             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14466                                                          ;BY  (013746  000172  000207)
14467
14468  055564  005215                          INC     (R5)
14469
```

```
14470                                   ;*****************************************************************************
14471                                   ;TEST:263      DIV     0 0 / #1 = 0       REM = 0      PS = 4
14472                                   ;*****************************************************************************
14473
14474  055566  010767  122400           TST263: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14475  055572  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14476  055576  012701  000000                   MOV     #0,%0+1         ;LOAD LOW ORDER WITH 0
14477  055602  071027  000001                   DIV     #1,%0           ;DIVIDE BY #1
14478  055606  106737  042262                   MFPS    @#PSWORD        ;SAVE PS
14479
14480  055612  122737  000004  042262           CMPB    #4,@#PSWORD     ;IS PS = 4
14481  055620  001403                           BEQ     .+10
14482  055622  004767  003014                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14483                                                                   ;PS IS WRONG
14484  055626  000326                           326             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14485                                                           ;BY  (013746  000172  000207)
14486
14487
14488  055630  022700  000000                   CMP     #0,%0           ;IS QUOTIENT = 0
14489  055634  001403                           BEQ     .+10
14490  055636  004767  003000                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14491                                                                   ;QUOTIENT IS WRONG
14492  055642  000327                           327             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14493                                                           ;BY  (013746  000172  000207)
14494
14495
14496  055644  022701  000000                   CMP     #0,%0+1         ;IS REMAINDER = 0
14497  055650  001403                           BEQ     .+10
14498  055652  004767  002764                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14499                                                                   ;WRONG REMAINDER
14500  055656  000330                           330             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14501                                                           ;BY  (013746  000172  000207)
14502
14503  055660  021527  000263                   CMP     (R5),#263
14504  055664  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14505  055666  004767  002750                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14506                                                                   ;TEST IS IN WRONG SEQUENCE
14507  055672  000331                           331             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14508                                                           ;BY  (013746  000172  000207)
14509
14510  055674  005215                           INC     (R5)
14511
```

```
14512                                   ;*************************************************************************
14513                                   ;TEST:264       DIV     -1 125252 / #2 = 152525       REM = 0      PS = 10
14514                                   ;*************************************************************************
14515
14516  055676  010767  122270          TST264: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14517  055702  012702  177777                  MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
14518  055706  012703  125252                  MOV     #125252,%2+1    ;LOAD LOW ORDER WITH 125252
14519  055712  071227  000002                  DIV     #2,%2           ;DIVIDE BY #2
14520  055716  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14521
14522  055722  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
14523  055730  001403                          BEQ     .+10
14524  055732  004767  002704                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14525                                                                  ;PS IS WRONG
14526  055736  000332                          332             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14527                                                          ;BY  (013746  000172  000207)
14528
14529
14530  055740  022702  152525                  CMP     #152525,%2      ;IS QUOTIENT = 152525
14531  055744  001403                          BEQ     .+10
14532  055746  004767  002670                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14533                                                                  ;QUOTIENT IS WRONG
14534  055752  000333                          333             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14535                                                          ;BY  (013746  000172  000207)
14536
14537
14538  055754  022703  000000                  CMP     #0,%2+1         ;IS REMAINDER = 0
14539  055760  001403                          BEQ     .+10
14540  055762  004767  002654                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14541                                                                  ;WRONG REMAINDER
14542  055766  000334                          334             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14543                                                          ;BY  (013746  000172  000207)
14544
14545  055770  021527  000264                  CMP     (R5),#264
14546  055774  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14547  055776  004767  002640                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14548                                                                  ;TEST IS IN WRONG SEQUENCE
14549  056002  000335                          335             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14550                                                          ;BY  (013746  000172  000207)
14551
14552  056004  005215                          INC     (R5)
14553
```

```
14554                                   ;*************************************************************************
14555                                   ;TEST:265       DIV     -1 -1 / #-1 = 1       REM = 0       PS = 0
14556                                   ;*************************************************************************
14557
14558  056006  010767  122160          TST265: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14559  056012  010501                          MOV     R5,R1           ;SAVE R5
14560  056014  012704  177777                  MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
14561  056020  012705  177777                  MOV     #-1,%4+1        ;LOAD LOW ORDER WITH -1
14562  056024  071427  177777                  DIV     #-1,%4          ;DIVIDE BY #-1
14563  056030  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14564
14565  056034  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14566  056042  001403                          BEQ     .+10
14567  056044  004767  002572                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14568                                                                  ;PS IS WRONG
14569  056050  000336                          336             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14570                                                          ;BY  (013746  000172  000207)
14571
14572
14573  056052  022704  000001                  CMP     #1,%4           ;IS QUOTIENT = 1
14574  056056  001403                          BEQ     .+10
14575  056060  004767  002556                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14576                                                                  ;QUOTIENT IS WRONG
14577  056064  000337                          337             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14578                                                          ;BY  (013746  000172  000207)
14579
14580
14581  056066  022705  000000                  CMP     #0,%4+1         ;IS REMAINDER = 0
14582  056072  001403                          BEQ     .+10
14583  056074  004767  002542                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14584                                                                  ;WRONG REMAINDER
14585  056100  000340                          340             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14586                                                          ;BY  (013746  000172  000207)
14587
14588  056102  010105                  MOV     R1,R5           ;RESTORE R5
14589  056104  021527  000265          CMP     (R5),#265
14590  056110  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14591  056112  004767  002524          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14592                                                          ;TEST IS IN WRONG SEQUENCE
14593  056116  000341                  341             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14594                                                  ;BY  (013746  000172  000207)
14595
14596  056120  005215                  INC     (R5)
14597
```

```
14598                                   ;*************************************************************************
14599                                   ;TEST:266      DIV     25253 1 / #125252 = 100000      REM = 1      PS = 10
14600                                   ;*************************************************************************
14601
14602  056122  010767  122044          TST266: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14603  056126  012700  025253                  MOV     #25253,%0       ;LOAD HIGH ORDER WITH 25253
14604  056132  012701  000001                  MOV     #1,%0+1         ;LOAD LOW ORDER WITH 1
14605  056136  071027  125252                  DIV     #125252,%0      ;DIVIDE BY #125252
14606  056142  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14607
14608  056146  122737  000010  042262          CMPB    #10,@#PSWORD    ;IS PS = 10
14609  056154  001403                          BEQ     .+10
14610  056156  004767  002460                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14611                                                                  ;PS IS WRONG
14612  056162  000342                          342             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14613                                                          ;BY  (013746  000172  000207)
14614
14615
14616  056164  022700  100000                  CMP     #100000,%0      ;IS QUOTIENT = 100000
14617  056170  001403                          BEQ     .+10
14618  056172  004767  002444                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14619                                                                  ;QUOTIENT IS WRONG
14620  056176  000343                          343             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14621                                                          ;BY  (013746  000172  000207)
14622
14623
14624  056200  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14625  056204  001403                          BEQ     .+10
14626  056206  004767  002430                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14627                                                                  ;WRONG REMAINDER
14628  056212  000344                          344             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14629                                                          ;BY  (013746  000172  000207)
14630
14631  056214  021527  000266                  CMP     (R5),#266
14632  056220  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14633  056222  004767  002414                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14634                                                                  ;TEST IS IN WRONG SEQUENCE
14635  056226  000345                          345             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14636                                                          ;BY  (013746  000172  000207)
14637
14638  056230  005215                          INC     (R5)
14639
```

```
14640                                   ;*************************************************************************
14641                                   ;TEST:267       DIV     37777 77777 / #77777 = 77777       REM = 77776      PS = 0
14642                                   ;*************************************************************************
14643
14644  056232  010767  121734          TST267: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14645  056236  012702  037777                  MOV     #37777,%2       ;LOAD HIGH ORDER WITH 37777
14646  056242  012703  077777                  MOV     #77777,%2+1     ;LOAD LOW ORDER WITH 77777
14647  056246  071227  077777                  DIV     #77777,%2       ;DIVIDE BY #77777
14648  056252  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14649
14650  056256  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14651  056264  001403                          BEQ     .+10
14652  056266  004767  002350                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14653                                                                  ;PS IS WRONG
14654  056272  000346                          346             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14655                                                           ;BY  (013746  000172  000207)
14656
14657
14658  056274  022702  077777                  CMP     #77777,%2       ;IS QUOTIENT = 77777
14659  056300  001403                          BEQ     .+10
14660  056302  004767  002334                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14661                                                                  ;QUOTIENT IS WRONG
14662  056306  000347                          347             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14663                                                           ;BY  (013746  000172  000207)
14664
14665
14666  056310  022703  077776                  CMP     #77776,%2+1     ;IS REMAINDER = 77776
14667  056314  001403                          BEQ     .+10
14668  056316  004767  002320                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14669                                                                  ;WRONG REMAINDER
14670  056322  000350                          350             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14671                                                           ;BY  (013746  000172  000207)
14672
14673  056324  021527  000267                  CMP     (R5),#267
14674  056330  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14675  056332  004767  002304                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14676                                                                  ;TEST IS IN WRONG SEQUENCE
14677  056336  000351                          351             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14678                                                           ;BY  (013746  000172  000207)
14679
14680  056340  005215                          INC     (R5)
14681
```

```
14682                                   ;*************************************************************************
14683                                   ;TEST:270       DIV     0 100000 / #2 = 40000       REM = 0      PS = 0
14684                                   ;*************************************************************************
14685
14686  056342  010767  121624         TST270: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14687  056346  010501                         MOV     R5,R1           ;SAVE R5
14688  056350  012704  000000                 MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14689  056354  012705  100000                 MOV     #100000,%4+1    ;LOAD LOW ORDER WITH 100000
14690  056360  071427  000002                 DIV     #2,%4           ;DIVIDE BY #2
14691  056364  106737  042262                 MFPS    @#PSWORD        ;SAVE PS
14692
14693  056370  122737  000000  042262         CMPB    #0,@#PSWORD     ;IS PS = 0
14694  056376  001403                         BEQ     .+10
14695  056400  004767  002236                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14696                                                                 ;PS IS WRONG
14697  056404  000352                         352             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14698                                                         ;BY  (013746  000172  000207)
14699
14700
14701  056406  022704  040000                 CMP     #40000,%4       ;IS QUOTIENT = 40000
14702  056412  001403                         BEQ     .+10
14703  056414  004767  002222                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14704                                                                 ;QUOTIENT IS WRONG
14705  056420  000353                         353             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14706                                                         ;BY  (013746  000172  000207)
14707
14708
14709  056422  022705  000000                 CMP     #0,%4+1         ;IS REMAINDER = 0
14710  056426  001403                         BEQ     .+10
14711  056430  004767  002206                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14712                                                                 ;WRONG REMAINDER
14713  056434  000354                         354             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14714                                                         ;BY  (013746  000172  000207)
14715
14716  056436  010105                         MOV     R1,R5           ;RESTORE R5
14717  056440  021527  000270                 CMP     (R5),#270
14718  056444  001403                         BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14719  056446  004767  002170                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14720                                                                 ;TEST IS IN WRONG SEQUENCE
14721  056452  000355                         355             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14722                                                         ;BY  (013746  000172  000207)
14723
14724  056454  005215                         INC     (R5)
14725
```

```
14726                                   ;*************************************************************************
14727                                   ;TEST:271      DIV     177777 77777 / #177776 = 40000       REM = 177777      PS = 0
14728                                   ;*************************************************************************
14729
14730  056456  010767  121510          TST271: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14731  056462  012700  177777                  MOV     #177777,%0      ;LOAD HIGH ORDER WITH 177777
14732  056466  012701  077777                  MOV     #77777,%0+1     ;LOAD LOW ORDER WITH 77777
14733  056472  071027  177776                  DIV     #177776,%0      ;DIVIDE BY #177776
14734  056476  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14735
14736  056502  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14737  056510  001403                          BEQ     .+10
14738  056512  004767  002124                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14739                                                                  ;PS IS WRONG
14740  056516  000356                          356             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14741                                                          ;BY  (013746  000172  000207)
14742
14743
14744  056520  022700  040000                  CMP     #40000,%0       ;IS QUOTIENT = 40000
14745  056524  001403                          BEQ     .+10
14746  056526  004767  002110                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14747                                                                  ;QUOTIENT IS WRONG
14748  056532  000357                          357             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14749                                                          ;BY  (013746  000172  000207)
14750
14751
14752  056534  022701  177777                  CMP     #177777,%0+1    ;IS REMAINDER = 177777
14753  056540  001403                          BEQ     .+10
14754  056542  004767  002074                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14755                                                                  ;WRONG REMAINDER
14756  056546  000360                          360             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14757                                                          ;BY  (013746  000172  000207)
14758
14759  056550  021527  000271                  CMP     (R5),#271
14760  056554  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14761  056556  004767  002060                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14762                                                                  ;TEST IS IN WRONG SEQUENCE
14763  056562  000361                          361             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14764                                                          ;BY  (013746  000172  000207)
14765
14766  056564  005215                          INC     (R5)
14767
```

```
14768                                   ;*****************************************************************************
14769                                   ;TEST:272       DIV     0 52525 / #52525 = 1       REM = 0      PS = 0
14770                                   ;*****************************************************************************
14771
14772  056566  010767  121400           TST272: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14773  056572  012702  000000                   MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
14774  056576  012703  052525                   MOV     #52525,%2+1     ;LOAD LOW ORDER WITH 52525
14775  056602  071227  052525                   DIV     #52525,%2       ;DIVIDE BY #52525
14776  056606  106737  042262                   MFPS    @#PSWORD        ;SAVE PS
14777
14778  056612  122737  000000  042262           CMPB    #0,@#PSWORD     ;IS PS = 0
14779  056620  001403                           BEQ     .+10
14780  056622  004767  002014                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14781                                                                   ;PS IS WRONG
14782  056626  000362                           362             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14783                                                           ;BY  (013746  000172  000207)
14784
14785
14786  056630  022702  000001                   CMP     #1,%2           ;IS QUOTIENT = 1
14787  056634  001403                           BEQ     .+10
14788  056636  004767  002000                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14789                                                                   ;QUOTIENT IS WRONG
14790  056642  000363                           363             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14791                                                           ;BY  (013746  000172  000207)
14792
14793
14794  056644  022703  000000                   CMP     #0,%2+1         ;IS REMAINDER = 0
14795  056650  001403                           BEQ     .+10
14796  056652  004767  001764                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14797                                                                   ;WRONG REMAINDER
14798  056656  000364                           364             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14799                                                           ;BY  (013746  000172  000207)
14800
14801  056660  021527  000272                   CMP     (R5),#272
14802  056664  001403                           BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14803  056666  004767  001750                   JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14804                                                                   ;TEST IS IN WRONG SEQUENCE
14805  056672  000365                           365             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14806                                                           ;BY  (013746  000172  000207)
14807
14808  056674  005215                           INC     (R5)
14809
```

```
14810                                   ;*****************************************************************************
14811                                   ;TEST:273      DIV     0 77777 / #0 = DUMMY       REM = DUMMY      PS = 3
14812                                   ;*****************************************************************************
14813
14814  056676  010767  121270          TST273: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14815  056702  010501                          MOV     R5,R1           ;SAVE R5
14816  056704  012704  000000                  MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
14817  056710  012705  077777                  MOV     #77777,%4+1     ;LOAD LOW ORDER WITH 77777
14818  056714  071427  000000                  DIV     #0,%4           ;DIVIDE BY #0
14819  056720  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14820  056724  042737  000014  042262          BIC     #14,@#PSWORD
14821
14822  056732  122737  000003  042262          CMPB    #3,@#PSWORD     ;IS PS = 3
14823  056740  001403                          BEQ     .+10
14824  056742  004767  001674                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14825                                                                  ;PS IS WRONG
14826  056746  000366                          366             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14827                                                          ;BY  (013746  000172  000207)
14828
14829
14830  056750  010105                          MOV     R1,R5           ;RESTORE R5
14831  056752  021527  000273                  CMP     (R5),#273
14832  056756  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14833  056760  004767  001656                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14834                                                                  ;TEST IS IN WRONG SEQUENCE
14835  056764  000367                          367             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14836                                                          ;BY  (013746  000172  000207)
14837
14838  056766  005215                          INC     (R5)
14839
```

```
14840                                   ;*************************************************************************
14841                                   ;TEST:274      DIV     77777 177777 / #2 = DUMMY       REM = DUMMY     PS = 2
14842                                   ;*************************************************************************
14843
14844  056770  010767  121176          TST274: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14845  056774  012700  077777                  MOV     #77777,%0       ;LOAD HIGH ORDER WITH 77777
14846  057000  012701  177777                  MOV     #177777,%0+1    ;LOAD LOW ORDER WITH 177777
14847  057004  071027  000002                  DIV     #2,%0           ;DIVIDE BY #2
14848  057010  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14849  057014  042737  000014  042262          BIC     #14,@#PSWORD
14850
14851  057022  122737  000002  042262          CMPB    #2,@#PSWORD     ;IS PS = 2
14852  057030  001403                          BEQ     .+10
14853  057032  004767  001604                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14854                                                                  ;PS IS WRONG
14855  057036  000370                          370             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14856                                                          ;BY  (013746  000172  000207)
14857
14858
14859  057040  021527  000274                  CMP     (R5),#274
14860  057044  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14861  057046  004767  001570                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14862                                                                  ;TEST IS IN WRONG SEQUENCE
14863  057052  000371                          371             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14864                                                          ;BY  (013746  000172  000207)
14865
14866  057054  005215                          INC     (R5)
14867
```

```
14868  057056  012702  000002                  MOV     #2,%2
14869  057062  012703  042324                  MOV     #S9,%3
14870  057066  012704  042326                  MOV     #S10,%4
14871
14872                                  ;*************************************************************************
14873                                  ;TEST:275      DIV     0 52525 / S9 = 25252       REM = 1      PS = 0
14874                                  ;*************************************************************************
14875
14876  057072  010767  121074          TST275: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14877  057076  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14878  057102  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14879  057106  071067  163212                  DIV     S9,%0           ;DIVIDE BY S9
14880  057112  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14881
14882  057116  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14883  057124  001403                          BEQ     .+10
14884  057126  004767  001510                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14885                                                                  ;PS IS WRONG
14886  057132  000372                                  372             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14887                                                                  ;BY  (013746  000172  000207)
14888
14889
14890  057134  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14891  057140  001403                          BEQ     .+10
14892  057142  004767  001474                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14893                                                                  ;QUOTIENT IS WRONG
14894  057146  000373                                  373             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14895                                                                  ;BY  (013746  000172  000207)
14896
14897
14898  057150  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14899  057154  001403                          BEQ     .+10
14900  057156  004767  001460                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14901                                                                  ;WRONG REMAINDER
14902  057162  000374                                  374             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14903                                                                  ;BY  (013746  000172  000207)
14904
14905  057164  021527  000275                  CMP     (R5),#275
14906  057170  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14907  057172  004767  001444                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14908                                                                  ;TEST IS IN WRONG SEQUENCE
14909  057176  000375                                  375             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14910                                                                  ;BY  (013746  000172  000207)
14911
14912  057200  005215                          INC     (R5)
14913
```

```
14914                                   ;*************************************************************************
14915                                   ;TEST:276       DIV     0 52525 / @S10 = 25252        REM = 1       PS = 0
14916                                   ;*************************************************************************
14917
14918  057202  010767  120764          TST276: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14919  057206  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14920  057212  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14921  057216  071077  163104                  DIV     @S10,%0         ;DIVIDE BY @S10
14922  057222  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14923
14924  057226  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14925  057234  001403                          BEQ     .+10
14926  057236  004767  001400                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14927                                                                  ;PS IS WRONG
14928  057242  000376                          376             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14929                                                          ;BY  (013746  000172  000207)
14930
14931
14932  057244  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14933  057250  001403                          BEQ     .+10
14934  057252  004767  001364                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14935                                                                  ;QUOTIENT IS WRONG
14936  057256  000377                          377             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14937                                                          ;BY  (013746  000172  000207)
14938
14939
14940  057260  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14941  057264  001403                          BEQ     .+10
14942  057266  004767  001350                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14943                                                                  ;WRONG REMAINDER
14944  057272  000400                          400             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14945                                                          ;BY  (013746  000172  000207)
14946
14947  057274  021527  000276                  CMP     (R5),#276
14948  057300  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14949  057302  004767  001334                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14950                                                                  ;TEST IS IN WRONG SEQUENCE
14951  057306  000401                          401             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14952                                                          ;BY  (013746  000172  000207)
14953
14954  057310  005215                          INC     (R5)
14955
```

```
14956                                   ;*************************************************************************
14957                                   ;TEST:277      DIV     0 52525 / @#S9 = 25252       REM = 1      PS = 0
14958                                   ;*************************************************************************
14959
14960  057312  010767  120654          TST277: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
14961  057316  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
14962  057322  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
14963  057326  071037  042324                  DIV     @#S9,%0         ;DIVIDE BY @#S9
14964  057332  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
14965
14966  057336  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
14967  057344  001403                          BEQ     .+10
14968  057346  004767  001270                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14969                                                                  ;PS IS WRONG
14970  057352  000402                          402             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14971                                                          ;BY  (013746  000172  000207)
14972
14973
14974  057354  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
14975  057360  001403                          BEQ     .+10
14976  057362  004767  001254                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14977                                                                  ;QUOTIENT IS WRONG
14978  057366  000403                          403             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14979                                                          ;BY  (013746  000172  000207)
14980
14981
14982  057370  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
14983  057374  001403                          BEQ     .+10
14984  057376  004767  001240                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14985                                                                  ;WRONG REMAINDER
14986  057402  000404                          404             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14987                                                          ;BY  (013746  000172  000207)
14988
14989  057404  021527  000277                  CMP     (R5),#277
14990  057410  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
14991  057412  004767  001224                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14992                                                                  ;TEST IS IN WRONG SEQUENCE
14993  057416  000405                          405             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
14994                                                          ;BY  (013746  000172  000207)
14995
14996  057420  005215                          INC     (R5)
14997
```

```
14998                                   ;*****************************************************************************
14999                                   ;TEST:300      DIV     0 52525 / %2 = 25252       REM = 1      PS = 0
15000                                   ;*****************************************************************************
15001
15002  057422  010767  120544          TST300: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15003  057426  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15004  057432  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15005  057436  071002                          DIV     %2,%0           ;DIVIDE BY %2
15006  057440  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15007
15008  057444  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15009  057452  001403                          BEQ     .+10
15010  057454  004767  001162                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15011                                                                   ;PS IS WRONG
15012  057460  000406                          406             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15013                                                           ;BY  (013746  000172  000207)
15014
15015
15016  057462  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15017  057466  001403                          BEQ     .+10
15018  057470  004767  001146                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15019                                                                   ;QUOTIENT IS WRONG
15020  057474  000407                          407             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15021                                                           ;BY  (013746  000172  000207)
15022
15023
15024  057476  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15025  057502  001403                          BEQ     .+10
15026  057504  004767  001132                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15027                                                                   ;WRONG REMAINDER
15028  057510  000410                          410             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15029                                                           ;BY  (013746  000172  000207)
15030
15031  057512  021527  000300                  CMP     (R5),#300
15032  057516  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15033  057520  004767  001116                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15034                                                                   ;TEST IS IN WRONG SEQUENCE
15035  057524  000411                          411             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15036                                                           ;BY  (013746  000172  000207)
15037
15038  057526  005215                          INC     (R5)
15039
```

```
15040                                   ;*************************************************************************
15041                                   ;TEST:301      DIV     0 52525 / (3)+ = 25252       REM = 1      PS = 0
15042                                   ;*************************************************************************
15043
15044  057530  010767  120436          TST301: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15045  057534  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15046  057540  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15047  057544  071023                          DIV     (3)+,%0         ;DIVIDE BY (3)+
15048  057546  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15049
15050  057552  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15051  057560  001403                          BEQ     .+10
15052  057562  004767  001054                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15053                                                                  ;PS IS WRONG
15054  057566  000412                                  412             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15055                                                                  ;BY  (013746  000172  000207)
15056
15057
15058  057570  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15059  057574  001403                          BEQ     .+10
15060  057576  004767  001040                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15061                                                                  ;QUOTIENT IS WRONG
15062  057602  000413                                  413             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15063                                                                  ;BY  (013746  000172  000207)
15064
15065
15066  057604  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15067  057610  001403                          BEQ     .+10
15068  057612  004767  001024                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15069                                                                  ;WRONG REMAINDER
15070  057616  000414                                  414             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15071                                                                  ;BY  (013746  000172  000207)
15072
15073  057620  021527  000301                  CMP     (R5),#301
15074  057624  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15075  057626  004767  001010                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15076                                                                  ;TEST IS IN WRONG SEQUENCE
15077  057632  000415                                  415             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15078                                                                  ;BY  (013746  000172  000207)
15079
15080  057634  005215                          INC     (R5)
15081
```

```
15082                                   ;*************************************************************************
15083                                   ;TEST:302      DIV     0 52525 / -(3) = 25252       REM = 1      PS = 0
15084                                   ;*************************************************************************
15085
15086  057636  010767  120330          TST302: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15087  057642  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15088  057646  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15089  057652  071043                          DIV     -(3),%0         ;DIVIDE BY -(3)
15090  057654  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15091
15092  057660  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15093  057666  001403                          BEQ     .+10
15094  057670  004767  000746                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15095                                                                  ;PS IS WRONG
15096  057674  000416                          416             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15097                                                           ;BY  (013746  000172  000207)
15098
15099
15100  057676  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15101  057702  001403                          BEQ     .+10
15102  057704  004767  000732                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15103                                                                  ;QUOTIENT IS WRONG
15104  057710  000417                          417             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15105                                                           ;BY  (013746  000172  000207)
15106
15107
15108  057712  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15109  057716  001403                          BEQ     .+10
15110  057720  004767  000716                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15111                                                                  ;WRONG REMAINDER
15112  057724  000420                          420             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15113                                                           ;BY  (013746  000172  000207)
15114
15115  057726  021527  000302                  CMP     (R5),#302
15116  057732  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15117  057734  004767  000702                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15118                                                                  ;TEST IS IN WRONG SEQUENCE
15119  057740  000421                          421             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15120                                                           ;BY  (013746  000172  000207)
15121
15122  057742  005215                          INC     (R5)
15123
```

```
15124                                   ;*************************************************************************
15125                                   ;TEST:303      DIV     0 52525 / 2(4) = 25252       REM = 1      PS = 0
15126                                   ;*************************************************************************
15127
15128  057744  010767  120222          TST303: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15129  057750  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15130  057754  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15131  057760  071064  000002                  DIV     2(4),%0         ;DIVIDE BY 2(4)
15132  057764  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15133
15134  057770  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15135  057776  001403                          BEQ     .+10
15136  060000  004767  000636                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15137                                                                  ;PS IS WRONG
15138  060004  000422                          422             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15139                                                          ;BY  (013746  000172  000207)
15140
15141
15142  060006  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15143  060012  001403                          BEQ     .+10
15144  060014  004767  000622                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15145                                                                  ;QUOTIENT IS WRONG
15146  060020  000423                          423             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15147                                                          ;BY  (013746  000172  000207)
15148
15149
15150  060022  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15151  060026  001403                          BEQ     .+10
15152  060030  004767  000606                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15153                                                                  ;WRONG REMAINDER
15154  060034  000424                          424             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15155                                                          ;BY  (013746  000172  000207)
15156
15157  060036  021527  000303                  CMP     (R5),#303
15158  060042  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15159  060044  004767  000572                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15160                                                                  ;TEST IS IN WRONG SEQUENCE
15161  060050  000425                          425             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15162                                                          ;BY  (013746  000172  000207)
15163
15164  060052  005215                          INC     (R5)
15165
```

```
15166                                   ;*************************************************************************
15167                                   ;TEST:304      DIV     0 52525 / @(4) = 25252       REM = 1      PS = 0
15168                                   ;*************************************************************************
15169
15170  060054  010767  120112          TST304: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15171  060060  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15172  060064  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15173  060070  071074  000000                  DIV     @(4),%0         ;DIVIDE BY @(4)
15174  060074  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15175
15176  060100  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15177  060106  001403                          BEQ     .+10
15178  060110  004767  000526                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15179                                                                  ;PS IS WRONG
15180  060114  000426                          426             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15181                                                           ;BY  (013746  000172  000207)
15182
15183
15184  060116  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15185  060122  001403                          BEQ     .+10
15186  060124  004767  000512                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15187                                                                  ;QUOTIENT IS WRONG
15188  060130  000427                          427             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15189                                                           ;BY  (013746  000172  000207)
15190
15191
15192  060132  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15193  060136  001403                          BEQ     .+10
15194  060140  004767  000476                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15195                                                                  ;WRONG REMAINDER
15196  060144  000430                          430             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15197                                                           ;BY  (013746  000172  000207)
15198
15199  060146  021527  000304                  CMP     (R5),#304
15200  060152  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15201  060154  004767  000462                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15202                                                                  ;TEST IS IN WRONG SEQUENCE
15203  060160  000431                          431             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15204                                                           ;BY  (013746  000172  000207)
15205
15206  060162  005215                          INC     (R5)
15207
```

```
15208                                   ;*************************************************************************
15209                                   ;TEST:305      DIV     0 52525 / @(4)+ = 25252       REM = 1      PS = 0
15210                                   ;*************************************************************************
15211
15212  060164  010767  120002          TST305: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15213  060170  012700  000000                  MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15214  060174  012701  052525                  MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15215  060200  071034                          DIV     @(4)+,%0        ;DIVIDE BY @(4)+
15216  060202  106737  042262                  MFPS    @#PSWORD        ;SAVE PS
15217
15218  060206  122737  000000  042262          CMPB    #0,@#PSWORD     ;IS PS = 0
15219  060214  001403                          BEQ     .+10
15220  060216  004767  000420                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15221                                                                   ;PS IS WRONG
15222  060222  000432                          432             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15223                                                          ;BY  (013746  000172  000207)
15224
15225
15226  060224  022700  025252                  CMP     #25252,%0       ;IS QUOTIENT = 25252
15227  060230  001403                          BEQ     .+10
15228  060232  004767  000404                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15229                                                                   ;QUOTIENT IS WRONG
15230  060236  000433                          433             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15231                                                          ;BY  (013746  000172  000207)
15232
15233
15234  060240  022701  000001                  CMP     #1,%0+1         ;IS REMAINDER = 1
15235  060244  001403                          BEQ     .+10
15236  060246  004767  000370                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15237                                                                   ;WRONG REMAINDER
15238  060252  000434                          434             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15239                                                          ;BY  (013746  000172  000207)
15240
15241  060254  021527  000305                  CMP     (R5),#305
15242  060260  001403                          BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15243  060262  004767  000354                  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15244                                                                   ;TEST IS IN WRONG SEQUENCE
15245  060266  000435                          435             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15246                                                          ;BY  (013746  000172  000207)
15247
15248  060270  005215                          INC     (R5)
15249
```

```
                                                B 10
15250                           ;*****************************************************************
15251                           ;TEST:306      DIV     0 52525 / @-(4) = 25252       REM = 1      PS = 0
15252                           ;*****************************************************************
15253
15254  060272  010767  117674  TST306: MOV     PC,LPADR        ;STORE ERROR LOOP ADDRESS
15255  060276  012700  000000          MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
15256  060302  012701  052525          MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
15257  060306  071054                  DIV     @-(4),%0        ;DIVIDE BY @-(4)
15258  060310  106737  042262?         MFPS    @#PSWORD        ;SAVE PS
15259
15260  060314  122737  000000  042262  CMPB    #0,@#PSWORD     ;IS PS = 0
15261  060322  001403                  BEQ     .+10
15262  060324  004767  000312          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15263                                                          ;PS IS WRONG
15264  060330  000436                  436             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15265                                                  ;BY  (013746  000172  000207)
15266
15267
15268  060332  022700  025252          CMP     #25252,%0       ;IS QUOTIENT = 25252
15269  060336  001403                  BEQ     .+10
15270  060340  004767  000276          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15271                                                          ;QUOTIENT IS WRONG
15272  060344  000437                  437             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15273                                                  ;BY  (013746  000172  000207)
15274
15275
15276  060346  022701  000001          CMP     #1,%0+1         ;IS REMAINDER = 1
15277  060352  001403                  BEQ     .+10
15278  060354  004767  000262          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15279                                                          ;WRONG REMAINDER
15280  060360  000440                  440             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15281                                                  ;BY  (013746  000172  000207)
15282
15283  060362  021527  000306          CMP     (R5),#306
15284  060366  001403                  BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT
15285  060370  004767  000246          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15286                                                          ;TEST IS IN WRONG SEQUENCE
15287  060374  000441                  441             ;TO SCOPE, REPLACE LAST:  BEQ  .+10 (001403)
15288                                                  ;BY  (013746  000172  000207)
15289
15290  060376  005215                  INC     (R5)
15291
```

```
15292  060400  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOCATION 4 IN STACK
15293  060404  012737  060420  000004          MOV     #1$,@#4         ;SET UP TIME OUT TRAP VECTOR
15294  060412  012737  000001  164000          MOV     #1,@#164000     ;TURN OFF MULTI-TESTER
15295  060420  012637  000004          1$:     MOV     (SP)+,@#4       ;RESTORE LOCATION 4
15296  060424  005227  177777                  INC     #-1             ;TYPE ID. THE FIRST PASS
15297  060430  001002                          BNE     SKPMSG          ;BRANCH AROUND AFTER THE FIRST PASS
15298  060432  104401  060456                  TYPE    ,MSG1
15299  060436  005267  001756          SKPMSG: INC     PASSPT          ;SHOULD PRINT THIS PASS?
15300  060442  022767  000017  001750          CMP     #17,PASSPT      ;DID 17-OCTAL PASSES YET?
15301  060450  001424                          BEQ     BREOP           ;IF YES, BRANCH TO EOP
15302  060452  000137  001024                  JMP     @#RESTRT        ;JUMP TO RESTART
15303  060456  005015  042115  030455  MSG1:   .ASCIZ  <15><12>*MD-11 CJKDBA-A  F-11 CPU DIAG.*<15><12>
15304  060464  020061  045103  042113
15305  060472  040502  040455  020040
15306  060500  026506  030461  041440
15307  060506  052520  042040  040511
15308  060514  027107  005015     000
15309          060522                  .EVEN
15310                                  .SBTTL ** ROUTINES LIST **
15311
15312  060522  012767  177777  001670  BREOP:  MOV     #-1,PASSPT      ;RESET PASSPT
15313
15314                                  .SBTTL  END OF PASS ROUTINE
15315
15316                                  ;*****************************************************************************************
15317                                  ;*INCREMENT THE PASS NUMBER ($PASS)
15318                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
15319                                  ;*IF THERES A MONITOR GO TO IT
15320                                  ;*IF THERE ISN'T JUMP TO RESTRT
15321
15322  060530                          $EOP:
15323  060530  000240                          SCOPE
15324  060532  005267  117550                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
15325  060536  042767  100000  117542          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
15326  060544  005327                          DEC     (PC)+           ;;LOOP?
15327  060546  000001          $EOPCT: .WORD   1
15328  060550  003022                          BGT     $DOAGN          ;;YES
15329  060552  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
15330  060554  000001          $ENDCT: .WORD   1
15331  060556  060546                          $EOPCT
15332  060560  104401  060625                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
15333  060564  016746  117516                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
15334  060570  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
15335  060572  104401  060622                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
15336  060576  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
15337  060602  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
15338  060604  000005                          RESET                   ;;CLEAR THE WORLD
15339  060606  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
15340  060610  000240                          NOP                     ;;SAVE ROOM
15341  060612  000240                          NOP                     ;;FOR
15342  060614  000240                          NOP                     ;;ACT11
15343  060616                  $DOAGN:
15344  060616  000137                          JMP     @(PC)+          ;;RETURN
15345  060620  001024          $RTNAD: .WORD   RESTRT
15346  060622     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
15347  060625     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
```

```
 15348  060632  050040  051501  020123
 15349  060640  000043
 15350
 15351
 15352                          .SBTTL  HALT ROUTINE
```

```
15353
15354                                   ;*      HALT ROUTINE
15355                                   ;*      ------------
15356                                   ;*
15357                                   ;*
15358                                   ;*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
15359                                   ;*
15360
15361  060642  017637  000000  000302  $HLT:   MOV     @(SP),@#$FATAL  ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
15362  060650  011637  061030          MOV     (SP),@#CONTIN   ;SAVE ERROR NUMBER ADDRESS
15363  060654  032777  020000  161450  BIT     #20000,@SWR     ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE OUTS
15364  060662  001021                  BNE     6$
15365  060664  104401  042344          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15366  060670  104401  060770          TYPE    ,MSGERR         ;INFORM ERROR
15367  060674  104401  042344          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15368  060700  013746  061030          MOV     @#CONTIN,-(SP)  ;RETREIVE ERROR NUMBER ADDR
15369  060704  162716  000004          SUB     #4,(SP)         ;CALCULATE ERROR PC
15370  060710  104402                  TYPOC                   ;TYPE PC
15371  060712  104401  042344          TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15372  060716  013746  000302          MOV     @#$FATAL,-(SP)  ;RETREIVE ERROR NUMBER
15373  060722  104403                  TYPOS                   ;TYPE ERROR NUMBER
15374  060724     003                  .BYTE   3
15375  060725     000                  .BYTE   0
15376  060726  105767  117366  6$:     TSTB    $ENV            ;IF WE ARE NOT UNDER APT. THEN GO TO
15377  060732  001403                  BEQ     8$              ;8$
15378  060734  005237  000300          INC     @#$MSGTY        ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
15379  060740  000777                  BR      .               ;AND LOOP
15380  060742  104401  042344  8$:     TYPE    ,$CRLF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15381  060746  005777  161360          TST     @SWR    ;IS IT REQUIRED TO HALT ON ERROR ?
15382  060752  100001                  BPL     10$             ;IF NOT THEN GO TO 10$
15383  060754  000000                  HALT
15384  060756  013746  061030  10$:    MOV     @#CONTIN,-(SP)  ;
15385  060762  062716  000002          ADD     #2,(SP)         ;CALCULATE RETURN ADDRESS
15386  060766  000207                  RTS     PC              ;RETURN
15387  060770  042440  051122  051117  MSGERR: .ASCIZ / ERROR!   PC, AND ERROR # ARE: /
15388  060776  020041  020040  041520
15389  061004  020054  047101  020104
15390  061012  051105  047522  020122
15391  061020  020043  051101  035105
15392  061026  000040
15393                                   .EVEN
15394  061030  000000                  CONTIN: .WORD 0
15395
15396                                   .SBTTL  POWER FAIL ROUTINE
15397  061032  012767  061042  116764  PWRDN:  MOV     #PWRUP,24
15398  061040  000000                  HALT
15399
15400  061042  012767  061032  116754  PWRUP:  MOV     #PWRDN,24
15401  061050  012706  001000          MOV     #BUFF,SP
15402  061054  132767  000040  117237  BITB    #40,$ENVM       ;WILL APT ALLOW PRINTING?
15403  061062  001013                  BNE     PFRES           ;NO
15404  061064  012700  061116          MOV     #MSGPWF,R0      ;GET MSG ADDR.
15405  061070  105737  177564  PWAIT:  TSTB    @#TPS           ;TTY READY
15406  061074  100375                  BPL     PWAIT           ;NO WAIT
15407  061076  112037  177566          MOVB    (R0)+,@#TPB     ;PRINT CHARACTER
15408  061102  001372                  BNE     PWAIT           ;NEXT IF NOT DONE.
```

```
15409  061104  105737  177564          PWAIT1: TSTB    @#TPS
15410  061110  100375                          BPL     PWAIT1
15411  061112  000167  117706          PFRES:  JMP     RESTRT
15412  061116  005015  047520  042527  MSGPWF: .ASCIZ  <15><12>.POWER FAILED!.
15413  061124  020122  040506  046111
15414  061132  042105  000041
15415                                  .EVEN
15416                                  .SBTTL  TYPE ROUTINE
15417
15418                                  ;*****************************************************************************
15419                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
15420                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
15421                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
15422                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
15423                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
15424                                  ;*
15425                                  ;*CALL:
15426                                  ;*1) USING A TRAP INSTRUCTION
15427                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
15428                                  ;*OR
15429                                  ;*      TYPE
15430                                  ;*      MESADR
15431                                  ;*
15432
15433  061136  105767  000261          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
15434  061142  100002                          BPL     1$              ;;BR IF YES
15435  061144  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
15436  061146  000430                          BR      3$              ;;LEAVE
15437  061150  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
15438  061152  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
15439  061156  122767  000001  117134          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
15440  061164  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
15441  061166  132767  000100  117125          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
15442  061174  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
15443  061176  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
15444  061202  004767  000230                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
15445  061206  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
15446  061210  132767  000040  117103  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
15447  061216  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
15448  061220  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
15449  061222  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
15450  061224  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
15451  061226  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
15452  061230  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
15453  061234  000002                          RTI                     ;;RETURN
15454  061236  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
15455  061242  001430                          BEQ     8$
15456  061244  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
15457  061250  001006                          BNE     5$
15458  061252  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
15459  061254  104401                          TYPE                    ;;TYPE A CR AND LF
15460  061256  042344                          $CRLF
15461  061260  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
15462  061264  000755                          BR      2$              ;;GET NEXT CHARACTER
15463  061266  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
15464  061272  126726  000124          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
```

```
15465  061276  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
15466  061300  016746  000114                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
15467                                                                 ;;AND THE NULL CHAR.
15468  061304  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
15469  061310  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
15470  061312  004767  000032                JSR     PC,$TYPEC       ;;GO TYPE A NULL
15471  061316  105367  000072                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
15472  061322  000770                        BR      7$              ;;LOOP
15473
15474                                        ;HORIZONTAL TAB PROCESSOR
15475
15476  061324  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
15477  061330  004767  000014        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
15478  061334  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
15479  061342  001372                        BNE     9$              ;;TAB STOP
15480  061344  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
15481  061346  000724                        BR      2$              ;;GET NEXT CHARACTER
15482  061350  105777  160766        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
15483  061354  100375                        BPL     $TYPEC
15484  061356  116677  000002  160754        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
15485  061364  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
15486  061372  001003                        BNE     1$              ;;BRANCH IF NO
15487  061374  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
15488  061400  000406                        BR      $TYPEX          ;;EXIT
15489  061402  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
15490  061410  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
15491  061412  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
15492  061414  000000        $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
15493  061416  000207        $TYPEX: RTS     PC
15494
15495  061420      000       $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
15496  061421      002       $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
15497  061422      012       $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
15498  061423      000       $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
15499  061424      077       $QUES:  .ASCII  "?"             ;;QUESTION MARK
15500  061425      012   000 $LF:    .ASCIZ  <12>            ;;LINEFEED
15501          061430                .EVEN
15502                                .SBTTL  APT COMMUNICATIONS ROUTINE
15503
15504                                ;*****************************************************************************************
15505  061430  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
15506  061436  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
15507  061444  000403                        BR      $ATYC
15508  061446  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
15509  061454                        $ATYC:
15510  061454  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
15511  061456  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
15512  061460  105767  000206                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
15513  061464  001450                        BEQ     5$              ;;IF NOT:  BR
15514  061466  122767  000001  116624        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
15515  061474  001031                        BNE     3$              ;;IF NOT:  BR
15516  061476  132767  000100  116615        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
15517  061504  001425                        BEQ     3$              ;;IF NOT:  BR
15518  061506  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
15519  061512  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
15520  061520  005767  116554          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
```

```
                                                H 10
15521  061524  001375                          BNE     1$              ;;IF NOT:  WAIT
15522  061526  010067  116562                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
15523  061532  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
15524  061534  001376                          BNE     2$
15525  061536  166700  116552                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
15526  061542  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
15527  061544  010067  116546                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
15528  061550  012767  000004  116522          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
15529  061556  000413                          BR      5$
15530  061560  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
15531  061566  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
15532  061574  016746  116176                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
15533  061600  004767  177332                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
15534  061604  000000                  4$:     .WORD   0
15535  061606                          5$:
15536  061606  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
15537  061612  001416                          BEQ     12$             ;;IF NOT:  BR
15538  061614  005767  116500                  TST     $ENV            ;;RUNNING UNDER APT?
15539  061620  001413                          BEQ     12$             ;;IF NOT:  BR
15540  061622  005767  116452          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
15541  061626  001375                          BNE     11$             ;;IF NOT:  WAIT
15542  061630  017667  000004  116444          MOV     @4(SP),$FATAL   ;;GET ERROR #
15543  061636  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
15544  061644  005267  116430                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
15545  061650  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
15546  061654  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
15547  061660  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
15548  061664  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
15549  061666  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
15550  061670  000207                          RTS     PC              ;;RETURN
15551  061672     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
15552  061673     000          $LFLG:  .BYTE   0               ;;LOG FLAG
15553  061674     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
15554          061676                  .EVEN
15555          000200          APTSIZE=200
15556          000001          APTENV=001
15557          000100          APTSPOOL=100
15558          000040          APTCSUP=040
15559                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
15560
15561                          ;*************************************************************************************
15562                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15563                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15564                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15565                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15566                          ;*REPLACED WITH SPACES.
15567                          ;*CALL:
15568                          ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
15569                          ;*      TYPDS                   ;;GO TO THE ROUTINE
15570
15571  061676                  $TYPDS:
15572  061676  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
15573  061700  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
15574  061702  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
15575  061704  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
15576  061706  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
15577  061710  012746  020200          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
15578  061714  016605  000020          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
15579  061720  100004                  BPL     1$              ;;BR IF INPUT IS POS.
15580  061722  005405                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
15581  061724  112766  000055  000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
15582  061732  005000          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
15583  061734  012703  062112          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
15584  061740  112723  000040          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
15585  061744  005002          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
15586  061746  016001  062102          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
15587  061752  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
15588  061754  002402                  BLT     4$              ;;BR IF DONE
15589  061756  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
15590  061760  000774                  BR      3$
15591  061762  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
15592  061764  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
15593  061766  001002                  BNE     5$              ;;FALL THROUGH IF 0
15594  061770  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
15595  061772  100407                  BMI     7$              ;;BR IF YES
15596  061774  106316          5$:     ASLB    (SP)            ;;MSD?
15597  061776  103003                  BCC     6$              ;;BR IF NO
15598  062000  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
15599  062006  052702  000060  6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
15600  062012  052702  000040  7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
15601  062016  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
15602  062020  005720                  TST     (R0)+           ;;JUST INCREMENTING
15603  062022  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
15604  062026  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
15605  062030  003002                  BGT     8$              ;;GO TO EXIT
15606  062032  010502                  MOV     R5,R2           ;;GET THE LSD
15607  062034  000764                  BR      6$              ;;GO CHANGE TO ASCII
15608  062036  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
15609  062040  100003                  BPL     9$              ;;BR IF NO
15610  062042  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
15611  062050  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
15612  062052  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
15613  062054  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
15614  062056  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
15615  062060  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
15616  062062  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
15617  062064  104401  062112          TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
15618  062070  016666  000002  000004  MOV     2(SP),4(SP)     ;;ADJUST THE STACK
15619  062076  012616                  MOV     (SP)+,(SP)
15620  062100  000002                  RTI                     ;;RETURN TO USER
15621  062102  023420          $DTBL:  10000.
15622  062104  001750                  1000.
15623  062106  000144                  100.
15624  062110  000012                  10.
15625  062112  000004          $DBLK:  .BLKW   4
15626                           .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
15627
15628                           ;*************************************************************************************
15629                           ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
15630                           ;*OCTAL (ASCII) NUMBER AND TYPE IT.
15631                           ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
15632                           ;*CALL:
```

```
15633                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15634                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
15635                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
15636                                   ;*      .BYTE   M               ;;M=1 OR 0
15637                                   ;*                                      ;;1=TYPE LEADING ZEROS
15638                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
15639                                   ;*
15640                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
15641                                   ;*$TYPOS OR $TYPOC
15642                                   ;*CALL:
15643                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15644                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
15645                                   ;*
15646                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
15647                                   ;*CALL:
15648                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
15649                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
15650
15651  062122  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
15652  062126  116667  000001  000211           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
15653  062134  112667  000207                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
15654  062140  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
15655  062144  000406                           BR      $TYPON
15656  062146  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
15657  062154  112767  000006  000165           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
15658  062162  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
15659  062170  010346                           MOV     R3,-(SP)        ;;SAVE R3
15660  062172  010446                           MOV     R4,-(SP)        ;;SAVE R4
15661  062174  010546                           MOV     R5,-(SP)        ;;SAVE R5
15662  062176  116704  000145                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
15663  062202  005404                           NEG     R4
15664  062204  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
15665  062210  110467  000132                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
15666  062214  116704  000125                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
15667  062220  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
15668  062224  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
15669  062226  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
15670  062230  000404                           BR      3$              ;;GO DO MSB
15671  062232  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
15672  062234  006105                           ROL     R5
15673  062236  006105                           ROL     R5
15674  062240  010503                           MOV     R5,R3
15675  062242  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
15676  062244  105367  000076                   DECB    $OMODE          ;;TYPE THIS DIGIT?
15677  062250  100016                           BPL     7$              ;;BR IF NO
15678  062252  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
15679  062256  001002                           BNE     4$              ;;TEST FOR 0
15680  062260  005704                           TST     R4              ;;SUPPRESS THIS 0?
15681  062262  001403                           BEQ     5$              ;;BR IF YES
15682  062264  005204                   4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
15683  062266  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
15684  062272  052703  000040           5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
15685  062276  110367  000040                   MOVB    R3,8$           ;;SAVE FOR TYPING
15686  062302  104401  062342                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
15687  062306  105367  000032           7$:     DECB    $OCNT           ;;COUNT BY 1
15688  062312  003347                           BGT     2$              ;;BR IF MORE TO DO
```

```
15689  062314  002402                          BLT     6$              ;;BR IF DONE
15690  062316  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
15691  062320  000744                          BR      2$              ;;GO DO THE LAST DIGIT
15692  062322  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
15693  062324  012604                          MOV     (SP)+,R4        ;;RESTORE R4
15694  062326  012603                          MOV     (SP)+,R3        ;;RESTORE R3
15695  062330  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
15696  062336  012616                          MOV     (SP)+,(SP)
15697  062340  000002                          RTI                     ;;RETURN
15698  062342     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
15699  062343     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
15700  062344     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
15701  062345     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
15702  062346  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
15703                                  .SBTTL  TRAP DECODER
15704
15705                                  ;*******************************************************************************
15706                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
15707                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
15708                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
15709                                  ;*GO TO THAT ROUTINE.
15710
15711  062350  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
15712  062352  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
15713  062356  005740                          TST     -(R0)           ;;BACKUP BY 2
15714  062360  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
15715  062362  006300                          ASL     R0              ;;POSITION FOR INDEXING
15716  062364  016000  062404                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
15717  062370  000200                          RTS     R0              ;;GO TO ROUTINE
15718
15719
15720                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
15721
15722  062372  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
15723  062374  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
15724  062402  000002                          RTI                     ;;RESTORE THE PSW
15725
15726                                  .SBTTL  TRAP TABLE
15727
15728                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
15729                                  ;*BY THE "TRAP" INSTRUCTION.
15730
15731                                  ;       ROUTINE
15732                                  ;       -------
15733  062404  062372                  $TRPAD: .WORD   $TRAP2
15734  062406  061136                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
15735  062410  062146                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
15736  062412  062122                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
15737  062414  062162                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
15738  062416  061676                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
15739
15740
15741  062420  177777                  PASSPT: -1
15742          000001                  .END
```

```
A       026126
ABASE = 000000
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
ADC1    020430
ADC2    020440
ADC3    020460
ADC4    020470
ADC5    020506
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    020244
ADD2    020254
ADD3    020270
ADD4    020300
ADD5    020316
ADD6    020326
ADD7    020340
ADD8    020350
ADD9    020370
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   025324
AROUND  042016
ASL1    021744
ASL2    021754
ASL3    021772
ASL4    022002
ASL5    022016
ASL6    022026
ASL7    022052
ASR1    022114
ASR2    022124
ASR3    022146
ASR4    022156
ASR5    022172
ASR6    022202
ASR7    022232
ASTART  042522
ASWREG= 000000
ATESTN= 000000
ATRAP   037554
AUNIT = 000000
AUSWR = 000000
AUTO1   037662
AVECT1= 000000
AVECT2= 000000
BELL  = 000240
BIC1    017374
BIC2    017404
BIC3    017422
BIS1    017464
BIS2    017474
BIS3    017514
BITCHK  025514
BITCLR  025442
BITCON  025576
BITSET  025460
BIT1    017304
BIT2    017314
BIT3    017332
BRA1    001130
BRA2    001140
BRA3    001152
BRA4    001160
BRA5    001170
BRCT    025362
BRC1    003376
BRC2    003406
BRC3    003416
BREOP   060522
BRH     025334
BRMFPD  027146
BRN1    003256
BRN2    003266
BRN3    003276
BRTAB   027242
BRV1    003326
BRV2    003336
BRV3    003346
BRZ1    003206
BRZ2    003216
BRZ3    003226
BR1     027536
BR10    027752
BR11    030056
BR12    030124
BR13    030172
BR14    030240
BR15    030336
BR16    030356
BR17    030376
BR2     027564
BR20    030416
BR21    030436
BR22    030456
BR23    030476
BR3     027606
BR33    030546
BR34    030560
BR35    030572
BR36    030604
BR37    030626
BR4     027632
BR40    030640
BR41    030652
BR45    032200
BR46    032746
BR46A   032756
BR47    033536
BR5     027656
BR51    034304
BR51A   034314
BR6     027702
BR7     027726
BR70    037272
BR71    040222
BTCON   025652
BTERR   025636
BTRAP   037630
BUFF  = 001000
CC    = 177776
CCERR   026072
CC1     025330
CC2     025774
CC3     025760
CLRCD   025756
CLR1    020032
CMP1    020674
CMP2    020704
CMP3    020726
CMP4    020736
CMP5    020762
CMP6    020772
CMP7    021012
COM1    021052
CONT    025364
CONTIN  061030
CON1    026010
CON2    026102
CORH    037614
COUNT   042260
CPUTYP  041644
CR    = 000015
CRLF  = 000200
CTRAP   037572
DAERR   025566
DBE     041460
DBE1    041502
DBE2    041472
DBE3    041522
DBE4    041542
DBE5    041560
DEC1    017672
DEC2    017702
DEC3    017716
DEC4    017726
DEC5    017742
DEC6    017752
DEC7    017774
DISPLA  042334
DISPRE  000174
DL11W   037732
DL11W1  037736
DNMB0A  011072
DNMB0B  011102
DNMB2A  011330
DNMB2B  011340
DNMB2C  011354
DNMB2D  011370
DNMB2E  011400
DNMB2F  011416
DNMB3A  011500
DNMB3B  011510
DNMB3C  011526
DNMB3D  011544
DNMB3E  011554
DNMB4A  011744
DNMB4B  011754
DNMB4C  011772
DNMB4D  012002
DNMB4E  012012
DNMB4F  012026
DNM03A  010164
DNM03B  010174
DNM03C  010204
DNM1    010036
DNM1A   011150
DNM1B   011160
DNM2    010052
DNM2A   011226
DNM2B   011236
DNM2C   011244
DNM2D   011254
DNM3    010070
DNM4    010112
DNM4A   011636
DNM4B   011646
DNM4C   011662
DNM5A   012106
DNM5B   012116
DNM5C   012134
DNM6A   012214
DNM6B   012224
DNM6C   012242
DNM7A   012324
DNM7B   012334
DNM7C   012352
DOPB2A  010576
DOPB2B  010654
DOP0A   007600
DOP0B   007624
DOP0C   007644
DOP0D   007674
DOP03A  007752
DOP03B  007762
DOP1    010410
DOP2    010522
DOP4    014026
DOP5    014112
DTRAP1  037612
DUMMY = 000000
END1    027232
ENT176  046734
ENT51   044274
ER      025346
```

```
ERRNM = 000442
F     = 000063
FINISH  042244
FIRST =%000005
FOVER   036520
FPP     042234
GIN1    042036
GIN2    042054
GIN3    042102
HERE  = 000000
HICORE  037464
HLT   = 000000
HT    = 000011
IJMP    016410
IJMP4   016162
IJMP5   016352
ILLA  = 004700
ILLB  = 000100
INC1    017554
INC2    017564
INC3    017606
INC4    017616
INC5    017632
INST    042246
INSTC   031004
INSTK   035162
ITRAP5= 000004
JMPCK   016412
JMPERR  025706
JMPSEQ  016432
JMPT    025676
JMP2    016164
JMP2A   016202
JMP3    016112
JMP3A   016130
JMP3B   016150
JMP4    016214
JMP4A   016232
JMP4B   016252
JMP5    016320
JMP5A   016340
JMP6    016264
JMP6A   016304
JMP7    016354
JMP7A   016374
JSRCK   017064
JSRCKA  017060
JSRCK1  017102
JSRSEQ  017062
JSR0    016446
JSR1    016452
JSR1A   016474
JSR2    016564
JSR2A   016614
JSR2B   016624
JSR3    016504
JSR3A   016544
JSR3B   016554
JSR4    016646
JSR4A   016664
JSR4B   016674
JSR5    016752
JSR5A   016776
JSR5B   017006
JSR6    016706
JSR6A   016732
JSR6AD  017056
JSR6B   016742
JSR7    017020
JSR7A   017036
JSR7B   017046
K1      027456
K10     027474
K11     027476
K12     027500
K2      027460
K3      027462
K4      027464
K5      027466
K6      027470
K7      027472
LAST  =%000001
LF    = 000012
LPADR   000172
MBDM2A  012570
MBDM2B  012600
MBDM2C  012616
MBDM2D  012630
MBDM2E  012640
MBDM2F  012656
MBDM4A  013212
MBDM4B  013230
MBDM4C  013242
MBDM4D  013252
MBDM4E  013266
MDM1A   012422
MDM1B   012432
MDM2A   012476
MDM2B   012506
MDM2C   012514
MDM2D   012524
MDM3A   012732
MDM3B   012742
MDM3C   012760
MDM3D   013000
MDM3E   013026
MDM4A   013106
MDM4B   013116
MDM4C   013132
MDM5A   013336
MDM5B   013346
MDM5C   013364
MDM5D   013402
MDM5E   013430
MDM6A   013500
MDM6B   013510
MDM6C   013526
MDM6D   013546
MDM6E   013576
MDM7A   013652
MDM7B   013662
MDM7C   013700
MDM7D   013720
MDM7E   013744
MFP10   025036
MFP10A  025064
MFPS1   023560
MFPS2A  023650
MFPS2B  023660
MFPS2C  023700
MFPS3A  023756
MFPS3B  023766
MFPS3C  024006
MFPS4A  024064
MFPS4B  024074
MFPS4C  024114
MFPS5A  024172
MFPS5B  024202
MFPS5C  024222
MFPS6A  024302
MFPS6B  024312
MFPS6C  024332
MFPS7A  024412
MFPS7B  024422
MFPS7C  024442
MFPT  = 000007
MOR0    026134
MOR1    026174
MOR2    026254
MOR3    026346
MOR4    026366
MOR5    026406
MOR6    026476
MOR7    026516
MOR8    026536
MOV1    017214
MOV2    017224
MOV3    017242
MRK1    022620
MRK2    022642
MRK3    022652
MRK4    022674
MRK5    022706
MRK6    022722
MSGERR  060770
MSGPWF  061116
MSG1    060456
MTP10   025150
MTPS1   022772
MTPS1A  023012
MTPS2   023066
MTPS3   023156
MTPS4   023244
MTPS5   023324
MTPS6   023414
MTPS7   023504
N     = 000307
NBR     025340
NEGAT   043674
NEG00   004346
NEG01   004356
NEG02   004372
NEG03   004406
NEG04   004416
NEG1    020550
NEG10   004462
NEG11   004472
NEG12   004510
NEG13   004524
NEG14   004534
NEG2    020560
NEG20   004602
NEG21   004612
NEG22   004640
NEG3    020602
NEG30   005162
NEG31   005172
NEG32   005206
NEG33   005232
NEG34   005246
NEG4    020612
NEG40   005640
NEG41   005650
NEG42   005664
NEG5    020632
NEG50   005740
NEG51   005750
NEG52   005764
NEG60   006042
NEG61   006052
NEG70   006122
NEG71   006132
NODL    040244
NODL1   040430
NODL2   040542
NODL3   041030
NOP   = 000240
NOR     037460
NOSUB   037576
PASSPT  062420
PCN01   026570
PCN1    041236
PCN2    026624
PCN3    026672
PCN4    026734
PCN5    026774
PFRES   061112
POWER   042352
PROFTE  037730
PS    = 177776
PSWORD  042262
PUSRM = 030000
PWAIT   061070
PWAIT1  061104
PWRDN   061032
PWRUP   061042
RA      033546
RA1     032210
RB      033534
RB1     032176
RC      033530
RC1     032172
REGR1   042670
REGR2   043036
REGR23  045546
REGR3   043204
REGR4   043350
REGR5   043520
REG01   045346
REG1    002066
REG1A   002132
REG1E   002100
REG2    002202
REG2A   002230
REG2B   002256
REG2C   002310
REG3    002336
REG3A   002402
REG3E   002350
REG4    002452
REG4A   002516
REG4E   002464
REG45   045746
REG5    002566
REG5A   002632
```

```
REG5E   002600
REG6    002702
REG6A   002746
REG6E   002714
RESET2  040630
RESET3  040620
REST    024536
RESTRT  001024
RET     042126
RETA    030710
RETAH   030720
RETAT   036610
RETA1   031464
RETA2   032272
RETA3   033024
RETA4   033630
RETA5   034362
RETB    030744
RETBT   036656
RETB1   031510
RETB2   032316
RETB3   033050
RETB4   033654
RETB5   034406
RETC    031006
RETCT   036730
RETC1   031552
RETC2   032360
RETC3   033112
RETC4   033716
RETC5   034450
RETD    031060
RETD1   031624
RETD2   032432
RETD3   033164
RETD4   033770
RETD5   034522
RETE    031124
RETE1   031666
RETE2   032476
RETE3   033230
RETE4   034034
RETE5   034566
RETF    031174
RETF1   031736
RETF2   032546
RETF3   033300
RETF4   034104
RETF5   034636
RETG    031306
RETG1   032050
RETG2   032660
RETG3   033412
RETG4   034216
RETG5   034750
RETH5   035076
RETJ    035122
RETK    035164
RETL    035236
RETM    035302
RETN    035352
RETO    035462
RETR1   041252
RETR2   041330
RETR3   041376
RET1    042144
RET2    042164
RET3    042202
RET4    042172
RITSH   046156
ROL1    021434
ROL2    021444
ROL3    021462
ROL4    021472
ROL5    021506
ROL6    021516
ROL7    021540
ROR1    021602
ROR2    021612
ROR3    021630
ROR4    021640
ROR5    021656
ROR6    021666
ROR7    021702
ROTX    015220
ROTXAD  015344
ROT0A   014336
ROT0B   014346
ROT0C   014370
ROT1A   014436
ROT1B   014446
ROT1C   014472
ROT1D   014502
ROT1E   014532
ROT2A   014604
ROT2B   014614
ROT2C   014644
ROT2D   014654
ROT2E   014710
ROT3A   014756
ROT3B   014766
ROT3C   015014
ROT3D   015024
ROT3E   015052
ROT4    015126
ROT5    015210
ROT6    015260
ROT7    015334
RTI1    037160
RTI2    037172
RTRAP  = 000010
RTRAP1= 000034
RTRAP2= 000020
RTRAP3= 000030
RTRAP4= 000014
RTRAP5= 000004
RTS1    017150
RTT1    037006
RTT2    037022
RTT3    037064
RTT4    037102
RTT5    037036
RTT6    037122
ROTRAP  037512
R1ERR   002150
R2ERR   002300
R3ERR   002420
R4ERR   002534
R5ERR   002650
R6     =%000006
R6ERR   002764
R7     =%000007
R7TRX   040100
SBC1    021266
SBC2    021276
SBC3    021314
SBC4    021324
SBC5    021342
SBC6    021352
SBC7    021372
SB0     015376
SB2     015520
SB4     015644
SB5     015732
SB5A    015724
SB5X    015742
SB5XAD  015744
SB6     016004
SB6X    016014
SB7     016054
SB7X    016064
SB7XAD  016066
SCOPE  = 000240
SC3     026052
SC4     026066
SECPRT  027452
SETBR   025220
SETCC   025240
SETCD   026050
SETUP   025202
SET2BR  025310
SHL     001536
SHLE    001552
SHR     001652
SHRE    001666
SKPMSG  060436
SKP104  037744
SKTST2  040646
SNMB0A  006326
SNMB1A  006432
SNMB1B  006442
SNMB1C  006464
SNMB2A  006606
SNMB2B  006616
SNMB2C  006632
SNMB2D  006652
SNMB2E  006662
SNMB3A  007024
SNMB3B  007034
SNMB3C  007052
SNMB3D  007062
SNM0A   006266
SNM1A   006370
SNM2A   006526
SNM2B   006536
SNM3A   006736
SNM3B   006746
SNM4A   007132
SNM4B   007142
SNM5A   007214
SNM5B   007224
SNM6A   007300
SNM6B   007310
SNM7A   007362
SNM7B   007372
SOB1    022502
SOB2    022510
SOB3    022520
SOB4    022540
SOPA    006204
SOPB    006224
SOPB0A  003614
SOPB0B  003624
SOPB1A  003736
SOPB1B  003754
SOPB1C  004020
SOPB1D  004040
SOPB2A  004174
SOPB2B  004214
SOPB2C  004264
SOPB2D  004310
SOPB3A  004776
SOPB3B  005022
SOPB3C  005070
SOPB3D  005112
SOPX    006170
SOPXAD  006234
SOPZA   004106
SOP0A   003452
SOP0B   003472
SOP0C   003534
SOP0D   003560
SOP1A   003662
SOP1B   003674
SOP2B   004126
SOP3A   004712
SOP3B   004726
SOP4A   005324
SOP4B   005344
SOP5A   005422
SOP5B   005436
SOP6A   005502
SOP6B   005516
SOP7A   005564
SOP7B   005600
START   001002
STATUS= 177776
STBOT   001000
STP     036530
STPP    031374
STPPA   031404
STP3    037366
STP3D   037376
STP4    041204
STP4E   041214
SUB0    007526
SUB0A   007536
SUB1    021114
SUB2    021124
SUB3    021146
SUB4    021156
SUB5    021174
SUB6    021204
SUB7    021224
SWB1    020156
SWB2    020166
SWB3    020204
SWR     042332
SWREG   000176
SW09   = 001000
SW10   = 002000
SW11   = 004000
SW12   = 010000
SXT0    022300
SXT1    022310
```

```
SXT2    022340
S0      042302
S1      042304
S10     042326
S11     042330
S2      042306
S3      042310
S4      042312
S5      042314
S6      042316
S7      042320
S8      042322
S9      042324
TAB    =%000003
TABLE   042214
TBL1    014046
TBL2    014132
TDEC1   035610
TDEC2   035634
TDEC3   035702
TDEC4   035750
TDEC6   035764
TDEC7   035774
TDEC77  040062
TDEC8   040052
TEMP1   042264
TEMP2   042266
TEMP3   042270
TEMP4   042272
TEMP5   042274
TEMP6   042276
TENSAV  042014
TESTN1  027502
TEST1   020070
TEST2   020100
TEST3   020116
THRPRT  042362
TONT1   037302
TO10    027362
TO114   027422
TO14    027372
TO244   027432
TO250   027442
TO30    027402
TO34    027412
TO4     027352
TPB    = 177566
TPS    = 177564
TRACE   037262
TRAPA  = 000077
TRAPB   037672
TRAP10  042012
TRAP24  042002
TRCSR  = 177560
TRC1    037356
TRPADR  037502
TRT    = 000003
TR0     040212
TR2     040232
TR3     040360
TR4     040372
TR5     040362
TST160  046212
TST161  046240
TST162  046256
TST163  046274
TST164  046336
TST165  046374
TST166  046420
TST167  046450
TST170  046506
TST171  046532
TST172  046560
TST173  046602
TST174  046640
TST175  046674
TST176  046752
TST177  047050
TST200  047146
TST201  047250
TST202  047346
TST203  047444
TST204  047546
TST205  047644
TST206  047772
TST207  050076
TST210  050202
TST211  050306
TST212  050410
TST213  050512
TST214  050614
TST215  050720
TST216  051024
TST217  051126
TST220  051230
TST221  051326
TST222  051424
TST223  051522
TST224  051624
TST225  051722
TST226  052020
TST227  052122
TST230  052220
TST231  052316
TST232  052414
TST233  052516
TST234  052614
TST235  052712
TST236  053010
TST237  053112
TST240  053210
TST241  053306
TST242  053424
TST243  053522
TST244  053620
TST245  053716
TST246  054012
TST247  054106
TST250  054202
TST251  054300
TST252  054376
TST253  054472
TST254  054566
TST255  054676
TST256  055006
TST257  055122
TST260  055232
TST261  055342
TST262  055456
TST263  055566
TST264  055676
TST265  056006
TST266  056122
TST267  056232
TST270  056342
TST271  056456
TST272  056566
TST273  056676
TST274  056770
TST275  057072
TST276  057202
TST277  057312
TST300  057422
TST301  057530
TST302  057636
TST303  057744
TST304  060054
TST305  060164
TST306  060272
TST37   043722
TST40   043756
TST41   043772
TST42   044010
TST43   044044
TST44   044076
TST45   044130
TST46   044162
TST47   044220
TST50   044250
TST51   044326
TST52   044410
TST53   044472
TST54   044554
TST55   044634
TST56   044714
TST57   044774
TST60   045056
TST61   045140
TST62   045220
TS1     001112
TS10    001512
TS100   006474
TS101   006554
TS102   006700
TS103   006764
TS104   007100
TS105   007156
TS106   007242
TS107   007324
TS11    001562
TS110   007406
TS111   007442
TS112   007476
TS113   007552
TS114   007714
TS115   010004
TS116   010130
TS117   010214
TS12    001626
TS120   010252
TS121   010310
TS122   010346
TS123   010426
TS124   010466
TS125   010540
TS126   010614
TS127   010672
TS13    001676
TS130   010734
TS131   010776
TS132   011040
TS133   011116
TS134   011174
TS135   011272
TS136   011434
TS137   011574
TS14    001726
TS140   011702
TS141   012044
TS142   012154
TS143   012262
TS144   012372
TS145   012446
TS146   012542
TS147   012676
TS15    001760
TS150   013054
TS151   013150
TS152   013304
TS153   013446
TS154   013616
TS155   013764
TS156   014050
TS157   014134
TS16    002012
TS160   014220
TS161   014304
TS162   014400
TS163   014542
TS164   014720
TS165   015062
TS166   015136
TS167   015222
TS17    002044
TS170   015270
TS171   015346
TS172   015414
TS173   015456
TS174   015536
TS175   015600
TS176   015660
TS177   015746
TS2     001202
TS20    002110
TS200   016016
TS201   016070
TS202   016434
TS203   017112
TS204   017166
TS205   017252
TS206   017342
TS207   017432
TS21    002160
TS210   017524
TS211   017642
TS212   020004
TS213   020042
TS214   020126
TS215   020214
TS216   020400
TS217   020516
TS22    002234
TS220   020642
TS221   021022
TS222   021062
```

```
TS223   021234      TS301   030000      TS360   036234      TS66    005526      YNTAB   027300
TS224   021402      TS302   030306      TS361   036304      TS67    005610      $APTHD  000330
TS225   021550      TS303   030516      TS362   036372      TS7     001434      $ATYC   061454
TS226   021712      TS304   030664      TS363   036442      TS70    005700      $ATY1   061430
TS227   022062      TS305   030720      TS364   036542      TS71    006002      $ATY3   061436
TS23    002314      TS306   030762      TS365   036610      TS72    006066      $ATY4   061446
TS230   022242      TS307   031026      TS366   036674      TS73    006156      $CHARC  061414
TS231   022350      TS31    002660      TS367   036750      TS74    006236      $CPUOP  000326
TS232   022462      TS310   031144      TS37    003166      TS75    006276      $CRLF   042344
TS233   022550      TS311   031416      TS370   037022      TS76    006336      $DBLK   062112
TS234   022732      TS312   031464      TS371   037122      TS77    006400      $DEVCT  000310
TS235   023022      TS313   031526      TS372   037206      TTCSR = 177564      $DOAGN  060616
TS236   023104      TS314   031572      TS373   037316      TTT37   041024      $DTBL   062102
TS237   023174      TS315   031706      TS374   037404      TTYOUT  042336      $ENDAD  060606
TS24    002360      TS316   032146      TS375   037744      TTY11   041014      $ENDCT  060554
TS240   023260      TS317   032236      TS376   040100      TTY3    040734      $ENDMG  060625
TS241   023342      TS32    002724      TS377   040244      TTY4    041006      $ENULL  060622
TS242   023432      TS320   032272      TS4     001274      TYPCNT  042300      $ENV    000320
TS243   023522      TS321   032334      TS40    003236      TYPDS = 104405      $ENVM   000321
TS244   023614      TS322   032400      TS400   040430      TYPE  = 104401      $EOP    060530
TS245   023716      TS323   032516      TS401   040542      TYPOC = 104402      $EOPCT  060546
TS246   024024      TS324   032770      TS402   040646      TYPON = 104404      $ERN  = 001162
TS247   024132      TS325   033024      TS403   041030      TYPOS = 104403      $ERROR= 000302
TS25    002430      TS326   033066      TS404   041214      USP1    024562      $ETABL  000320
TS250   024240      TS327   033132      TS405   041272      USP1A   024606      $ETEND  000330
TS251   024350      TS33    002774      TS406   041340      USP2    024662      $FATAL  000302
TS252   024460      TS330   033250      TS407   041414      USP3    024706      $FFLG   061674
TS253   024536      TS331   033504      TS41    003306      USP4    024742      $FILLC  061422
TS254   024614      TS332   033574      TS410   041564      USRM  = 140000      $FILLS  061421
TS255   024760      TS333   033630      TS411   041652      USTBOT  027352      $GET42  060576
TS256   025064      TS334   033672      TS412   041736      VDEC1   036036      $HIBTS  000330
TS257   025172      TS335   033736      TS42    003356      VDEC10  036266      $HLT    060642
TS26    002474      TS336   034054      TS43    003426      VDEC11  036336      $LF     061425
TS260   025432      TS337   034326      TS44    003506      VDEC12  036346      $LFLG   061673
TS261   025602      TS34    003034      TS45    003570      VDEC13  036424      $MAIL   000300
TS262   025656      TS340   034362      TS46    003634      VDEC14  036434      $MBADR  000332
TS263   025716      TS341   034424      TS47    003704      VDEC2   036026      $MFLG   061672
TS264   026134      TS342   034470      TS5     001332      VDEC3   036106      $MSGAD  000314
TS265   026214      TS343   034606      TS50    003764      VDEC4   036076      $MSGLG  000316
TS266   026274      TS344   035042      TS51    004050      VDEC5   036156      $MSGTY  000300
TS267   026424      TS345   035076      TS52    004136      VDEC6   036146      $NULL   061420
TS27    002544      TS346   035140      TS53    004224      VDEC7   036226      $OCNT   062344
TS270   026554      TS347   035204      TS54    004320      VDEC8   036216      $OMODE  062346
TS271   026610      TS35    003072      TS55    004432      VDEC9   036276      $PASS   000306
TS272   026650      TS350   035322      TS56    004552      WATE    041156      $PASTM  000336
TS273   026716      TS351   035554      TS57    004656      WATE1   041102      $QUES   061424
TS274   026760      TS352   035610      TS6     001370      WATE2   041116      $RTNAD  060620
TS275   027020      TS353   035652      TS60    004736      WATE3   041144      $SETUP= 000020
TS276   027066      TS354   035774      TS61    005032      XOR1    022414      $STUP = 177777
TS277   027146      TS355   036044      TS62    005122      XOR2    022424      $SVPC = 000400
TS3     001236      TS356   036114      TS63    005272      XOR3    022452      $SWR  = 000000
TS30    002610      TS357   036164      TS64    005354      XXT     041646      $SWREG  000322
TS300   027502      TS36    003130      TS65    005446      YBR     025344      $TESTN  000304
```

```
$TN    = 000413          $TRAP2  062372          $TYPE   061136          $UNIT   000312          $$GET4= 000000
$TPB     042340          $TRP    = 000006        $TYPEC  061350          $UNITM  000340          $OFILL  062345
$TPCNT   042301          $TRPAD  062404          $TYPEX  061416          $USWR   000324          .       = 062422
$TPFLG   061423          $TSTM   000334          $TYPOC  062146          $X      = 041746          .$X     = 000330
$TPS     042342          $TSTNM= 000304          $TYPON  062162          $XX     = 177666
$TRAP    062350          $TYPDS  061676          $TYPOS  062122          $XXX    = 000665

. ABS.  062422     000


 ERRORS DETECTED:  0

 CJKDBA,CJKDBA/SOL/NL:TOC=CJKDBA.P11
 RUN-TIME: 49 62 1 SECONDS
 RUN-TIME RATIO: 260/113=2.2
 CORE USED:  19K  (37 PAGES)
```