IDENTIFICATION

PRODUCT CODE: AC-8041C-MC

PRODUCT NAME: CFKAAC0 11/34 BSC INST TST

PRODUCT DATE: 30-OCT-78

MAINTAINER: DIAGNOSTIC ENGINEERING

```
**************************************
*                                    *
* SUMMARY OF OPERATING INSTRUCTIONS  *
*                                    *
**************************************
```

THE FOLLOWING PROCEDURE CAN BE USED TO RUN THIS DIAGNOSTIC IN A STANDARD CONFIGURATION WITH AT LEAST 4K OF MEMORY AND A TELETYPE. IF THE PROGRAM DOES NOT RUN SUCCESSFULLY CONSULT THE FOLLOWING DOCUMENT FOR ASSISTANCE.

OPERATING PROCEDURES:

1. LOAD THE PROGRAM USING NORMAL PROCEDURES
2. START THE PROGRAM AT LOCATION 200
3. PROGRAM SHOULD PRINT THE TITLE WITHIN THE 1ST SECOND AND END PASS REPEATABLY THEREAFTER AT APPROX. 10 SEC. INTERVALS UNTIL EXTERNALLY HALTED.
4. IF THE PROGRAM DOES NOT RUN AS DESCRIBED ABOVE, CONSULT THE FULL OPERATING INTRUCTIONS WHICH FOLLOW.

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP-11/34 BASIC INSTRUCTION SET. THE PROGRAM EXERCISES ALL OF THE PROCESSOR LOGIC AND MICROCODE FOR ALL INSTRUCTIONS EXCEPT THE TRAP AND MEMORY MANAGEMENT INSTRUCTIONS. THE PROGRAM DOES NOT TEST INSTRUCTIONS OR HARDWARE RELATED TO THE TRAP OR INTERRUPT MECHANISMS OF THE 11/34 (E.G. RTT, RT1, WAIT, RESET, TRAP, EMT).

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE

PDP-11/34 PROCESSOR
8K MEMORY -- THE PROGRAM USES LOCATIONS 0 - 26520

1.2.2 SOFTWARE

THIS PROGRAM IS WRITTEN TO BE RUN AS A STAND-ALONE PROGRAM. HOWEVER, THE PROGRAM IS DESIGNED TO RUN UNDER AUTOMATED PRODUCT TEST SYSTEM (APT) IN ALL THREE MODES.

THE PROGRAM CAN ALSO BE RUN UNDER THE ACT 11 MONITOR

1.3 RELATED DOCUMENTS AND STANDARDS

PDP-11/34 MICROCODE LISTING

PDP-11/34 ELECTRICAL SCHEMATICS

DIAGNOSTIC ENGINEERING PROJECT PLANFOR 11/34

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES
DOCUMENT NO. 175-003-009-00

APT INTERFACE SPECIFICATION, REVISION 9.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

1.5 FAILURE ASSUMPTIONS

NONE

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING

USE NORMAL PROCEDURES FOR LOADING ABSOLUTE BINARY TAPES.

2.1.2 NORMAL START

THIS IS THE PROCEDURE FOR NORMAL PROGRAM RUNNING (I.E., STARTING WITH TEST 1 AND EXECUTING ENTIRE DIAGNOSTIC).

LOAD ADDRESS = 200
START

2.1.3 SUBTEST START

THIS IS THE PROCEDURE FOR STARTING AT A SUBTEST OTHER THAN 1.

1. LOAD $TESTN (IN MAILBOX SECTION) WITH THE NUMBER OF SUBTEST MINUS ONE (IN OCTAL) FOR EXAMPLE, TO START AT SUBTEST 100, $TESTN=77.
2. LOAD STARTING ADDRESS OF SUBTEST IN LOC. 216
3. LOAD ADDRESS = 204
4. START

2.2 SPECIAL ENVIRONMENTS

THIS PROGRAM IS WRITTEN TO COMPLY WITH ALL THE REQUIREMENTS OF THE APT INTERFACE SPECIFICATION. IT WILL RUN UNDER APT IN EITHER QUICK VERIFY, PROGRAM OR RUN-TIME MODES.

THIS PROGRAM IS WRITTEN TO COMPLY WITH ALL OF THE REQUIREMENTS OF PROGRAMS TO RUN UNDER THE ACT11 MONITOR.

2.3 PROGRAM OPTIONS

THIS PROGRAM IS INTENDED TO BE A BASIC PROCESSOR TEST.
IT IS INTENDED TO BE THE LOWEST LEVEL DIAGNOSTIC RUN.
IT PROVIDES FOR NO SELECTABLE OPTIONS.

IN ORDER THAT THE TEST BE RUNNABLE ON A PROCESSOR WITHOUT A TELETYPE, IT IS POSSIBLE TO SUPPRESS THE END OF PASS MESSAGE. IF NO TELETYPE IS AVAILABLE, ALTER THE BYTE, $ENVM, WHICH IS LOCATED IN THE APT MAILBOX. SETTING $ENVM TO 40(8) WILL SUPPRESS ALL CONSOLE OUTPUT.
THE EXACT LOCATION OF THIS BYTE CAN BE FOUND IN THE SYMBOL TABLE AT THE END OF THE LISTING.

2.4 EXECUTION TIMES

THE DIAGNOSTIC COMPLETES THE FIRST PASS IN LESS THAN 1 SEC.
SUBSEQUENT PASSES REQUIRE APPROXIMATELY 10 SECS. EACH.
THE PROGRAM WILL RUN CONTINUOUSLY UNTIL EXTERNALLY HALTED.

3.0 ERROR INFORMATION

3.1 ERROR TYPES

THERE ARE TWO BASIC TYPES OF ERRORS IN THE DIAGNOSTIC.

3.1.1 FUNCTIONAL ERRORS

THESE ARE ERRORS WHICH REPRESENT A MALFUNCTION OF AN INSTRUCTION OR SEQUENCE OF INSTRUCTION. (E.G., THE PROPER CONDITION CODE NOT SET OR IMPROPER RESULT OF AN ARITHMETIC OR LOGICAL OPERATION).

3.1.2 SEQUENCE ERRORS

THE RESULT OF A TESTS BEING EXECUTED OUT OF SEQUENCE. (E.G. WILD MACHINE OR IMPROPER BRANCH OR JUMP).

3.2 ERROR REPORTING PROCEDURES

THE DIAGNOSTIC RESPONDS TO THE DETECTION OF ALL ERRORS BY STORING CERTAIN INFORMATION IN MEMORY AND HALTING THE PROCESSOR. THE INFORMATION STORED IN MEMORY CAN BE USED BY THE OPERATOR TO IDENTIFY THE ERROR DETECTED.

CERTAIN FAILURES WILL CAUSE THE PROESSOR TO HANG.
THIS TYPE OF FAILURE IS INDICATED IF THE PROGRAM
DOES NOT PRINT ITS END OF PASS INDICATION WITHIN A REASONABLE
AMOUNT OF TIME. (FIRST MESSAGE SHOULD APPEAR WITHIN 1 SEC.)

3.3 ERROR DESCRIPTOR INFORMATION

THE DIAGNOSTIC MAILBOX HOLDS THE ERROR INFORMATION NECESSARY TO IDENTIFY THE DETECTED ERROR. THIS INFORMATION HAS BEEN DESIGNED FOR COMPLIANCE WITH THE APT TO DIAGNOSTIC INTERFACE SPECIFICATION. IT IS THE PRIMARY MEDIUM FOR IDENTIFYING ERRORS.

3.2.1 $MSGTYP

THIS LOCATION IS INCREMENTED FROM ZERO TO ONE BEFORE THE PROGRAM COMES TO A PROGRAMMED HALT. IF THIS LOCATION IS NOT ONE, THEN THE DIAGNOSTIC HAS COME TO AN UNPROGRAMMED HALT. CHECK THE STACK AND PC FOR A CLUE TO THE CAUSE. SUSPECT A TRAP.

3.2.2 $FATAL

THIS LOCATION IS LOADED WITH A NUMBER BEFORE A HALT IS EXECUTED. EACH PROGRAMMED HALT HAS A UNIQUE NUMBER ASSOCIATED WITH IT WHICH CAN BE USED TO IDENTIFY THE ERROR WHICH HAS BEEN DETECTED.

3.2.3 $PASS

THIS LOCATION IS INCREMENTED FOR EVERY COMPLETE PASS OF THE DIAGNOSTIC. MONITORING THIS LOCATION WILL INDICATE WHETHER OR NOT THE PROGRAM IS HUNG. IT WILL ALSO INDICATE THE NUMBER OF SUCCESSFUL PASSES COMPLETED BEFORE THE ERROR HALT. A HIGH PASS COUNT MIGHT INDICATE THAT THE ERROR HALT IS ASSOCIATED WITH AN INTERMITTANT FAULT.

3.2.4 $TESTN

THIS LOCATION IS INCREMENTED IN EACH NEW SUBTEST. THIS SHOULD INDICATE THE TEST BEING EXECUTED WHEN THE ERROR WAS DETECTED. THIS LOCATION IS ALSO USED TO DETECT A SEQUENCE ERROR.

3.4 ERROR IDENTIFICATION

BECAUSE OF THE OVERHEAD ASSOCIATED WITH EACH HALT IN AN APT COMPATIBLE PROGRAM THE SEQUENCE CHECK CODE WILL SHARE THE ERROR HALT OF FUNCTIONAL ERROR WITHIN EACH SUBTEST. TO DETERMINE WHICH ERROR IS BEING REPORTED, LOCATIONS $FATAL AND $TESTN ARE USED TOGETHER. WHEN AN ERROR HALT OCCURS, CHECK, $FATAL TO DETERMINE THE NUMBER OF THE ERROR DETECTED. NOW, CHECK THAT THE TEST NUMBER WHERE THIS ERROR IS DETECTED CORRESPONDS TO THE VALUE IN $TESTN. IF THESE AGREE THE ERROR WAS A FUNCTIONAL ERROR AS DESCRIBED IN THE LISTINGS. IF THESE NUMBERS DO NOT AGREE, THEN A SEQUENCE ERROR WAS DETECTED. IN THIS CASE $TESTN WILL CONTAIN ONE MORE THAN THE NUMBER OF THE LAST TEST SUCCESSFULLY COMPLETED. SEQUENCE ERRORS WHICH SHARE THE ERROR HALTS OF FUNCTIONAL ERRORS WILL ALWAYS BE REPORTED BY THE LAST HALT IN THE SUBTEST IN WHICH THEY WERE DISCOVERED.

4.0 PROGRESS REPORT

AT THE END OF EACH SUCCESSFUL PASS (THE EQUIVALENT OF 400 (8) PROGRAM PASSES, EXCEPT THE FIRST PASS WHICH IS ONLY ONE PROGRAM PASS) THE PROGRAM INCREMENTS THE LOCATION $PASS WHICH IS IN THE APT MAILBOX. THIS LOCATION WILL ALWAYS CONTAIN THE NUMBER OF SUCCESSFUL PASSES COMPLETED. $PASS IS RESET WITH EVERY RETART FROM LOC. 200.

ADDITIONALLY, THE TITLE AND END PASS MESSAGE IS PRINTED ON THE CONSOLE TELETYPE AFTER THE FIRST PASS. THE END PASS MESSAGE IS REPEATED EVERY SUBSEQUENT PASS (400 PROGRAM LOOPS) THEREAFTER.

IF NO TELETYPE IS AVAILABLE, THE CONSOLE OUTPUT MUST BE SUPPRESSED. (SEE SECTION 2.3).

5.0 TROUBLE SHOOTING

WHEN THE PROGRAM DISCOVERS A FAULT IT WILL HALT. TO DETERMINE THE CAUSE OF THE HALT, THE DIAGNOSTIC PROVIDES ERROR INFORMATION. THIS INFORMATION IS STORED IN THE APT MAILBOX AND IS THE PRIMARY SOURCE OF ERROR IDENTIFICATION.

UPON FINDING AN ERROR, THE FOLLOWING PROCEDURE SHOULD AID IN ISOLATING THE FAULT.

5.1 CHECK THE MAILBOX

1. $MSGTY — THIS LOCATION SHOULD CONTAIN A 1. IF THE PROCESSOR HALTS AND THIS LOCATION IS ZERO, THEN THE PROCESSOR HAS COME TO AN UNEXPECTED HALT. FIRST SUSPECT A TRAP. CHECK THE PC AND IF A TRAP CHECK R6 AND THE STACK FOR THE LOCATION OF THE FAILING INSTRUCTION.

2. $FATAL — THIS LOCATION IS USED TO HOLD THE NUMBER OF THE ERROR WHICH HAS BEEN DETECTED. EACH ERROR BEING CHECKED BY THE DIAGNOSTIC IS ASSIGNED A UNIQUE NUMBER WHICH IS STORED IN $FATAL WHEN THAT ERROR IS DETECTED.

   WHEN AN ERROR IS DETECTED, CHECK THE LISTING TO SEE THAT THE ERROR NUMBER STORED IN $FATAL IS ONE WHICH IS DETECTED IN THE TEST WHOSE NUMBER IS IN $TESTN. IF THERE IS A DISAGREEMENT THEN THE ERROR BEING REPORTED IS A SEQUENCE ERROR. $TESTN CONTAINS ONE MORE THAN THE LAST TEST WHICH WAS SUCCESSFULLY COMPLETED.

3. $TESTN — THIS LOCATION IS USED TO INDICATE THE NUMBER OF THE TEST WHICH WAS BEING EXECUTED WHEN THE FUALT WAS DETECTED. $TESTN IS USED IN CONJUCTION WITH $FATAL TO DISTINGUISH BETWEEN SEQUENCE AND FUNCTONAL ERRORS. (SEE 2. THIS SECTION)

4. $PASS — THIS LOCATION IS USED TO INDICATE THE NUMBER OF SUCCESSFUL PASSES WHICH THE DIAGNOSTIC HAS COMPLETED. THIS WILL GIVE AN INDICATION THAT THE DIAGNOSTIC HAS NOT JUST BEEN HUNG IN A LOOP IF NOT TELETYPE IS AVAILABLE TO REPORT THE PRINTED PROGRESS REPORTS.

   IF AN ERROR HAS BEEN DETECTED $PASS WILL SHOW WHETHER IT WAS A HARD ERROR DISCOVERED DURING THE FIRST TRY OR WHETHER IT WAS INTERMITTANT OR DEVERLOPED DURING THE RUNNING OF THE DIAGNOSTIC.

5.2 SCOPING

WHILE THIS DIAGNOSTIC IS PRIMARILY INTENDED TO BE A FAULT DETECTION PROGRAM, PROVISIONS ARE MADE TO ASSIST A TECHNICIAN WHO MIGHT WANT TO USE THE PROGRAM AS A TROUBLE SHOOTING TEST.

THE PROCEDURE FOR SCOPING A SUBTEST INVOLVES MODIFYING SEVERAL MEMORY LOCATIONS IN THE TEST ITSELF. THE PHILOSOPHY IS TO PROVIDE A SCOPING LOOP WHICH WILL INCLUDE THE CODE WHERE THE ERROR WAS DETECTED. THE LOOP IS SET UP SO THAT THE LOOP WILL NOT BE TERMINATED SHOULD THE ERROR INTERMITTANTLY DISAPPEAR.

THE PROCEDURE IS AS FOLLOWS:

1. DETERMINE WHICH ERROR IS TO BE SCOPED. USE $FATAL AND $TESTN FOR THIS (SEE ABOVE)
2. LOCATE THE ERROR ROUTINE IN THE LISTING.
3. CLEAR THE RIGHT BYTE OF THE CONDITIONAL BRANCH INSTRUCTION ASSOCIATED WITH THE ERROR. (THIS IS MARKED WITH <===='S IN THE LISTING.)
4. REPLACE THE INSTRUCTION FOLLOWING <MOV #XXX,-(R2)) WITH THE SCOPING BRANCH PROVIDED IN THE LISTING COMMENTS.
5. RESTART THE PROGRAM. THE PROGRAM MAY BE RESTARTED FROM THE BEGINNING OR FROM THE SUBTEST (SEE 2.0).

6.0 LISTING

```
                                        .TITLE  CFKAAC0 11/34 BSC INST TST
                                        .ENABLE ABS
            000500                      STBOT=500
                                        .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   026034                     $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
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```



```
            000330                      .$X=.   ;;SAVE CURRENT LOCATION
            000024                      .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
   000024   000200                     200     ;;FOR APT START UP
            000044                      .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
   000044   000330                     $APTHDR ;;POINT TO APT HEADER BLOCK
            000330                      .=.$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   000010              $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
   000340   000000              $UNITM: .WORD   0       ;;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   026424                     TO4
   000006   000000                     0
   000010   026434                     TO10
   000012   000000                     0
   000014   026444                     TO14
            000030                      .=30
   000030   026454                     TO30
   000032   000000                     0
   000034   026464                     TO34
   000036   000000                     0
            000114                      .=114
   000114   026474                     TO114
   000116   000000                     0
            000244                      .=244
   000244   026504                     TO244
   000246   000000                     0
   000250   026514                     TO250
   000252   000000                     0

                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;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
            000500                      .=500
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;SET UP STARTING ADDRESS
            000500                      .$X=.
            000200                      .=200
   000200   000167   000274              JMP     START

   000204   012706   000500              MOV     #STBOT,R6               ;SET STACK POINTER
   000210   012702   000304              MOV     #$TESTN,R2              ;SET MAILBOX POINTER
```

```
                                        JMP     @(PC)+                  ;JUMP TO SUBTEST
000214  000137                          0                               ;ADDR. OF SUBTEST GOES HERE

                000500                  .=.$X
                000302                  $ERROR=$FATAL
                000304                  $TSTNM=$TESTN
000500  012737  026310  000024  START:  MOV     #PWRDN,@#24             ;SET UP FOR POWER FAIL
000506  012737  000000  000306          MOV     #0,@#$PASS      ;CLEAR PASS COUNT
000514  012737  177777  026060          MOV     #-1,@#PASSPT    ;SET PRINT COUNTER
000522  012706  000500          RESTRT: MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
000526  012702  000304                  MOV     #TESTN,R2       ;SET UP POINTER TO MESSAGE TYPE
000532  012737  000000  000304          MOV     #0,@#$TSTNM     ;CLEAR TEST NUMBER
000540  012737  000000  000302          MOV     #0,@#$ERROR     ;CLEAR ERROR NUMBER
000546  012737  000000  000300          MOV     #0,@#$MSGTY     ;CLEAR MESSAGE TYPE(FOR APT)
```



```
                        ;*****************************************************************************
                        ;TEST 1 CHECK BRANCHES ON Z BIT
                        ;*****************************************************************************
000554  005212  TST1:   INC     (R2)            ;UPDATE TEST NUMBER
000556  022712  000001  CMP     #1,(R2)         ;SEQUENCE ERROR?
000562  001024          BNE     TST2-10         ;BR TO ERROR HALT ON SEQ ERROR
000564  000257          CCC                     ;CLEAR ALL CONDITION CODES
000566  001401          BEQ     BR1             ;SHOULD BRANCH
000570  000404          BR      BR2             ;BAD BRANCH OF Z-BIT
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           BRANCH INSTRUCTION AND        <====
                                                ;           REPLACE THE MOVE INSTRUCTION  <====
                                                ;           FOLLOWING W/ 774              <====
000572          BR1:
000572  012742  000001          MOV     #1,-(R2)        ;MOVE TO MAILBOX #  *******  1  *******
000576  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000600  000000                  HALT                    ;SHOULD HAVE BRANCHED: Z=0
000602          BR2:
000602  001004          BNE     BR3

                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND  <====
                                                ;           REPLACE THE MOVE INSTRUCTION  <====
                                                ;           WHICH FOLLOWS W/ 770          <====
000604  012742  000002          MOV     #2,-(R2)        ;MOVE TO MAILBOX #  *******  2  *******
000610  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000612  000000                  HALT
000614  000264          BR3:    SEZ                     ;
000616  001001          BNE     BR4
000620  000404          BR      BR5

                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           BRANCH INSTRUCTION AND        <====
                                                ;           REPLACE THE MOVE INSTRUCTION  <====
                                                ;           FOLLOWING W/ 760              <====
000622          BR4:
000622  012742  000003          MOV     #3,-(R2)        ;MOVE TO MAILBOX #  *******  3  *******
000626  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000630  000000                  HALT                    ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
000632          BR5:
000632  001404          BEQ     TST2

                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND  <====
                                                ;           REPLACE THE MOVE INSTRUCTION  <====
                                                ;           WHICH FOLLOWS W/ 754          <====
000634  012742  000004          MOV     #4,-(R2)        ;MOVE TO MAILBOX #  *******  4  *******
000640  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000642  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, THE UNIBUS
                                ;DATA TRANSCIEVERS, AND AMUX CONTROL FOR ALU AND UBUS INPUTS.
                                ;       IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)
                                ;TO SEE WHICH BITS OF THE DATA PATH ARE FAILING.  IF THIS PROVIDES
                                ;INCONCLUSIVE DATA, TRY TO CHECK MODE 3 IR DECODE BY RUNNING
                                ;JUST THE MICROCODE AND IR DECODE TESTS FOR THE MOVE AND COMPARE
                                ;INSTRUCTIONS.
                                ;*****************************************************************************
                                ;TEST 2 TEST OF ZEROES IN THE DATA PATH
                                ;*****************************************************************************
000644  005212                  TST2:   INC     (R2)            ;UPDATE TEST NUMBER
000646  022712  000002                  CMP     #2,(R2)         ;SEQUENCE ERROR?
000652  001006                          BNE     TST3-10         ;BR TO ERROR HALT ON SEQ ERROR
000654  012737  000000  000000          MOV     #0,@#0          ;MOVE ZEROES THRU ADDRESS LINES, DATA
                                                                ;LINES AND INTERNAL PATHS
000662  005737  000000                  TST     @#0             ;SUCCESSFUL?
000666  001404                          BEQ     TST3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 772           <====
000670  012742  000005                  MOV     #5,-(R2)        ;MOVE TO MAILBOX #  *******  5  *******
000674  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000676  000000                          HALT                    ;DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 3 TEST OF PATTERN 125252 IN DATA PATH
                                ;*****************************************************************************
000700  005212                  TST3:   INC     (R2)            ;UPDATE TEST NUMBER
000702  022712  000003                  CMP     #3,(R2)         ;SEQUENCE ERROR?
000706  001007                          BNE     TST4-10         ;BR TO ERROR HALT ON SEQ ERROR
000710  012737  125252  000000          MOV     #125252,@#0     ;MOVE ALTERNATING ONES AND ZEROES
                                                                ;THRU DATA PATHS
000716  022737  125252  000000          CMP     #125252,@#0     ;SUCCESSFUL
000724  001404                          BEQ     TST4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
000726  012742  000006                  MOV     #6,-(R2)        ;MOVE TO MAILBOX #  *******  6  *******
000732  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000734  000000                          HALT                    ;DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;TEST 4 TEST OF PATTERN 052525 IN DATA PATH
                                ;*****************************************************************************
000736  005212                  TST4:   INC     (R2)            ;UPDATE TEST NUMBER
000740  022712  000004                  CMP     #4,(R2)         ;SEQUENCE ERROR?
000744  001007                          BNE     TST5-10         ;BR TO ERROR HALT ON SEQ ERROR
000746  012737  052525  000000          MOV     #052525,@#0     ;MOVE ALTERNATING ZEROES AND ONES
                                                                ;THRU DATA PATH
000754  022737  052525  000000          CMP     #052525,@#0     ;SUCCESSFUL?
000762  001404                          BEQ     TST5
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
000764  012742  000007                  MOV     #7,-(R2)        ;MOVE TO MAILBOX #  *******  7  *******
000770  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
000772  000000                          HALT                    ;DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 5 TEST OF ALL ONES IN DATA PATH
                                ;*****************************************************************************
000774  005212                  TST5:   INC     (R2)            ;UPDATE TEST NUMBER
000776  022712  000005                  CMP     #5,(R2)         ;SEQUENCE ERROR?
001002  001007                          BNE     TST6-10         ;BR TO ERROR HALT ON SEQ ERROR
001004  012737  177777  000000          MOV     #177777,@#0     ;MOVE ONES THRU DATA PATH
001012  022737  177777  000000          CMP     #177777,@#0     ;SUCCESSFUL
001020  001404                          BEQ     TST6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
001022  012742  000010                  MOV     #10,-(R2)       ;MOVE TO MAILBOX #  *******  10  *******
001026  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001030  000000                          HALT                    ;DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                .SBTTL  B-REGISTER TEST
                                ;
                                ;       THE B-REGISTER 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.  IF THIS PROVIDES
                                ;INCONCLUSIVE DATA TRY TO CHECK THE MODE 3 IR DECODE BY RUNNING JUST
                                ;THE MICROCODE AND IR DECODE TESTS FOR THE PARTICULAR INSTRUCTIONS.
                                ;
                                ;*****************************************************************************
                                ;TEST 6 SHIFT BIT 0 TO BIT 1
                                ;*****************************************************************************
001032  005212                  ;ST6:   INC     (R2)            ;UPDATE TEST NUMBER
001034  022712  000006          ;       CMP     #6,(R2)         ;SEQUENCE ERROR?
001040  001012                  ;       BNE     ;ST7-10         ;BR TO ERROR HALT ON SEQ ERROR
001042  000241                  ;       CLC                     ;CLEAR CARRY BIT
001044  012737  000001  000000  ;       MOV     #1,@#0          ;LOAD A 1
001052  006137  000000          ;       ROL     @#0             ;SHIFT LEFT
001056  022737  000002  000000  ;       CMP     #2,@#0          ;SUCCESSFUL
001064  001404                  ;       BEQ     ;ST7
                                ;                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;                               ;           CONDITIONAL BRANCH INST. AND    <====
                                ;                               ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;                               ;           WHICH FOLLOWS W/ 766            <====
001066  012742  000011          ;       MOV     #11,-(R2)       ;MOVE TO MAILBOX #  *******  11  *******
001072  005242                  ;       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001074  000000                  ;       HALT                    ;BIT 1 NOT SET
                                ;                               ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 7 SHIFT CARRY INTO BIT 0
                                ;*****************************************************************************
001076  005212                  ;ST7:   INC     (R2)            ;UPDATE TEST NUMBER
001100  022712  000007          ;       CMP     #7,(R2)         ;SEQUENCE ERROR?
001104  001017                  ;       BNE     ;ST10-10        ;BR TO ERROR HALT ON SEQ ERROR
001106  012737  000000  000000  ;       MOV     #0,@#0          ;CLEAR LOCATION
001114  000261                  ;       SEC                     ;SET CARRY
001116  006137  000000          ;       ROL     @#0             ;ROTATE CARRY BIT TO BIT 0
001122  103014                  ;       BCC     ;ST10
                                ;                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;                               ;           CONDITIONAL BRANCH INST. AND    <====
                                ;                               ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;                               ;           WHICH FOLLOWS W/ 771            <====
001124  012742  000012          ;       MOV     #12,-(R2)       ;MOVE TO MAILBOX #  *******  12  *******
001130  005242                  ;       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001132  000000                  ;       HALT                    ;CARRY CLEAR
                                ;                               ;  OR SEQUENCE ERROR
001134  022737  000001  000000  ;       CMP     #1,@#0          ;BIT 0 SET
001142  001404                  ;       BEQ     ;ST10
```



```
                                ;                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;                               ;           CONDITIONAL BRANCH INST. AND    <====
                                ;                               ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;                               ;           WHICH FOLLOWS W/ 761            <====
001144  012742  000013          ;       MOV     #13,-(R2)       ;MOVE TO MAILBOX #  *******  13  *******
001150  005242                  ;       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001152  000000                  ;       HALT                    ;BIT 0 NOT SET
                                ;                               ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 10        LEFT SHIFT FROM BIT 0 TO C-BIT
                                ;*****************************************************************************
001154  005212                  ;ST10:  INC     (R2)            ;UPDATE TEST NUMBER
001156  022712  000010          ;       CMP     #10,(R2)        ;SEQUENCE ERROR?
001162  001014                  ;       BNE     ;ST11-10        ;BR TO ERROR HALT ON SEQ ERROR
001164  012737  000001  000000  ;       MOV     #1,@#0          ;SET BIT 0
001172  012700  177757          ;       MOV     #-21,R0         ;SET BIT COUNTER
001176  000241                  ;       CLC                     ;CLEAR C-BIT
001200  005200                  ;SHL:   INC     R0              ;INCREMENT BIT COUNTER
001202  001404                  ;       BEQ     ;SHLE           ;BR TO ERROR HALT IF BIT IS LOST
001204  006137  000000          ;       ROL     @#0             ;SHIFT LEFT ONE POSITION
001210  103373                  ;       BCC     ;SHL            ;BRANCH IF C-BIT NOT SET
001212  001404                  ;       BEQ     ;ST11
                                ;                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;                               ;           CONDITIONAL BRANCH INST. AND    <====
                                ;                               ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;                               ;           WHICH FOLLOWS W/ 764            <====
001214                          ;SHLE:
001214  012742  000014          ;       MOV     #14,-(R2)       ;MOVE TO MAILBOX #  *******  14  *******
001220  005242                  ;       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001222  000000                  ;       HALT                    ;LEFT SHIFTING LOGIC FAILED
                                ;                               ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 11        SHIFT BIT 15 TO BIT 14
                                ;*****************************************************************************
001224  005212                  ;ST11:  INC     (R2)            ;UPDATE TEST NUMBER
001226  022712  000011          ;       CMP     #11,(R2)        ;SEQUENCE ERROR?
001232  001012                  ;       BNE     ;ST12-10        ;BR TO ERROR HALT ON SEQ ERROR
001234  012737  100000  000000  ;       MOV     #100000,@#0     ;SET BIT 15
001242  000241                  ;       CLC                     ;CLEAR CARRY
001244  006037  000000          ;       ROR     @#0             ;SHIFT BIT 15 TO BIT 14
001250  022737  040000  000000  ;       CMP     #40000,@#0      ;SUCCESSFUL
001256  001404                  ;       BEQ     ;ST12
                                ;                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;                               ;           CONDITIONAL BRANCH INST. AND    <====
                                ;                               ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;                               ;           WHICH FOLLOWS W/ 766            <====
001260  012742  000015          ;       MOV     #15,-(R2)       ;MOVE TO MAILBOX #  *******  15  *******
001264  005242                  ;       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001266  000000                  ;       HALT                    ;BIT 14 NOT SET
                                ;                               ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 12        RIGHT SHIFT FROM BIT 15 TO C-BIT
                                ;*****************************************************************************
```

```
001270  005212                  TST12:  INC     (R2)            ;UPDATE TEST NUMBER
001272  022712  000012                  CMP     #12,(R2)        ;SEQUENCE ERROR?
001276  001014                          BNE     TST13-10        ;BR TO ERROR HALT ON SEQ ERROR
001300  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
001306  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
001312  000241                          CLC                     ;CLEAR C-BIT
001314  005200                  SHR:    INC     R0              ;INCREMENT BIT COUNTER
001316  001404                          BEQ     SHRE            ;BR TO ERROR HALT IF BIT IS LOST
001320  006037  000000                  ROR     @#0             ;ROTATE RIGHT ONE POSITION
001324  103373                          BCC     SHR             ;BRANCH IF C-BIT CLEAR
001326  001404                          BEQ     TST13
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
001330                          SHRE:
001330  012742  000016                  MOV     #16,-(R2)       ;MOVE TO MAILBOX #  *******  16  *******
001334  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001336  000000                          HALT                    ;RIGHT SHIFT LOGIC FAILED
                                                                ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        .SBTTL  SCRATCH PAD TESTS
                        ;
                        ;       THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
                        ;DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
                        ;CIRCUITRY.  MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
                        ;R0 CAN HOLD VARIOUS DATA PATTERNS.  EACH DATA PATTERN IS
                        ;MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING.  THE
                        ;SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
                        ;TO THE SCRATCH PAD ITSELF.
                        ;       THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
                        ;A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
                        ;BIT AT A TIME INTO THE CARRY BIT.  THE RESULT IS THEN CHECKED TO INSURE THAT
                        ;NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
                        ;CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
                        ;ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
                        ;THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.
                        ;       AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
                        ;AS WELL AS REGISTER 11.  REGISTERS 10 AND 12 HAVE BEEN ACCESSED BY
                        ;THE INSTRUCTIONS.  REGISTERS 13,14,AND 17 WILL BE TESTED LATER IN THE
                        ;MICROCODE TESTS.
                        ;       IF THE PATTERN TESTS WITH REGISTER 0 FAIL CHECK THE RESULTANT
                        ;DATA FOR A CLUE TO A FAULT IN THE EXTERNAL CIRCUITRY.  IF THE
                        ;PATTERN TESTS WITH R0 ARE SUCCESSFUL BUT THE TESTS WITH THE OTHER
                        ;REGISTERS FAIL, SUSPECT THE REGISTER SELECT LINES AND THEN THE SCRATCH
                        ;PAD ITSELF.
                        ;
                        ;*****************************************************************************
                        ;TEST 13        TEST IF R0 CAN HOLD ALL ZEROES
                        ;*****************************************************************************
001340  005212                  TST13:  INC     (R2)            ;UPDATE TEST NUMBER
001342  022712  000013                  CMP     #13,(R2)        ;SEQUENCE ERROR?
001346  001004                          BNE     TST14-10        ;BR TO ERROR HALT ON SEQ ERROR

001350  012700  000000                  MOV     #0,R0           ;MOVE ZEROES TO R0
001354  005700                          TST     R0              ;SUCCESSFUL?
001356  001404                          BEQ     TST14
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 774           <====
001360  012742  000017                  MOV     #17,-(R2)       ;MOVE TO MAILBOX #  *******  17  *******
001364  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001366  000000                          HALT                    ;R0 NOT 0
                                                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 14        TEST IF R0 CAN HOLD ONES AND ZEROES
                        ;*****************************************************************************
001370  005212                  TST14:  INC     (R2)            ;UPDATE TEST NUMBER
001372  022712  000014                  CMP     #14,(R2)        ;SEQUENCE ERROR?
001376  001005                          BNE     TST15-10        ;BR TO ERROR HALT ON SEQ ERROR
001400  012700  125252                  MOV     #125252,R0      ;MOVE ALTERNATING ONES AND ZEROES TO R0
001404  020027  125252                  CMP     R0,#125252      ;SUCCESSFUL?
001410  001404                          BEQ     TST15
```

```
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 773           <====
001412  012742  000020          MOV     #20,-(R2)       ;MOVE TO MAILBOX #  *******  20  *******
001416  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001420  000000                  HALT                    ;R0 NOT 125252
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 15        TEST IF R0 CAN HOLD ZEROES AND ONES
                        ;*****************************************************************************
001422  005212          TST15:  INC     (R2)            ;UPDATE TEST NUMBER
001424  022712  000015          CMP     #15,(R2)        ;SEQUENCE ERROR?
001430  001005                  BNE     TST16-10        ;BR TO ERROR HALT ON SEQ ERROR
001432  012700  052525          MOV     #052525,R0      ;MOVE ALTERNATING ZEROES AND ONES TO R0
001436  020027  052525          CMP     R0,#052525      ;SUCCESSFUL?
001442  001404                  BEQ     TST16
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 773           <====
001444  012742  000021          MOV     #21,-(R2)       ;MOVE TO MAILBOX #  *******  21  *******
001450  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001452  000000                  HALT                    ;R0 NOT 52525
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 16        TEST IF R0 CAN HOLD ALL ONES
                        ;*****************************************************************************
001454  005212          TST16:  INC     (R2)            ;UPDATE TEST NUMBER
001456  022712  000016          CMP     #16,(R2)        ;SEQUENCE ERROR?
001462  001005                  BNE     TST17-10        ;BR TO ERROR HALT ON SEQ ERROR
001464  012700  177777          MOV     #177777,R0      ;MOVE ALL ONES TO R0
001470  020027  177777          CMP     R0,#177777      ;SUCCESSFUL?
001474  001404                  BEQ     TST17
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 773           <====
001476  012742  000022          MOV     #22,-(R2)       ;MOVE TO MAILBOX #  *******  22  *******
001502  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001504  000000                  HALT                    ;R0 NOT 177777
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 17        TEST IF R1 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
001506  005212          TST17:  INC     (R2)            ;UPDATE TEST NUMBER
001510  022712  000017          CMP     #17,(R2)        ;SEQUENCE ERROR?
001514  001012                  BNE     TST20-10        ;BR TO ERROR HALT ON SEQ ERROR
001516  012701  000001          MOV     #1,R1           ;SET BIT 0
001522  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
001526  000241                  CLC                     ;CLEAR C-BIT
001530  005200          REG1:   INC     R0              ;INCREMENT BIT COUNTER
001532  001403                  BEQ     REG1E           ;BR TO ERROR HALT IF BIT IS LOST
```



```
001534  006101                  ROL     R1              ;ROTATE 1 POSITION
001536  103374                  BCC     REG1            ;ALL DONE
001540  001404                  BEQ     TST20
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 766           <====
001542          REG1E:
001542  012742  000023          MOV     #23,-(R2)       ;MOVE TO MAILBOX #  *******  23  *******
001546  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001550  000000                  HALT                    ;FAILURE WITH R1
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 20        TEST IF R1 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
001552  005212          TST20:  INC     (R2)            ;UPDATE TEST NUMBER
001554  022712  000020          CMP     #20,(R2)        ;SEQUENCE ERROR?
001560  001014                  BNE     TST21-10        ;BR TO ERROR HALT ON SEQ ERROR
001562  012701  177776          MOV     #-2,R1          ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
001566  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
001572  000261                  SEC                     ;SET C-BIT
001574  005200          REG1A:  INC     R0              ;INCREMENT COUNTER
001576  001405                  BEQ     R1ERR           ;BR TO ERROR HALT IF COUNTER=0
001600  006101                  ROL     R1              ;ROTATE 1 POSITION
001602  103774                  BCS     REG1A           ;CONTINUE UNTIL C-BIT IS CLEAR
001604  022701  177777          CMP     #-1,R1          ;CHECK DATA IN R1
001610  001404                  BEQ     TST21
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           WHICH FOLLOWS W/ 764           <====
001612          R1ERR:
001612  012742  000024          MOV     #24,-(R2)       ;MOVE TO MAILBOX #  *******  24  *******
001616  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001620  000000                  HALT                    ;FAILURE WITH R1
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 21        TEST IF R2 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
001622  005212          TST21:  INC     (R2)            ;UPDATE TEST NUMBER
001624  022712  000021          CMP     #21,(R2)        ;SEQUENCE ERROR?
001630  001012                  BNE     REG2A-14
                                                        ;BR TO ERROR HALT ON SEQ ERROR
001632  012702  000001          MOV     #1,R2           ;SET BIT 0
001636  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
001642  000241                  CLC                     ;CLEAR C-BIT
001644  005200          REG2:   INC     R0              ;INCREMENT BIT COUNTER
001646  001403                  BEQ     REG2A-14        ;BR TO ERROR HALT IF BIT IS LOST
001650  006102                  ROL     R2              ;ROTATE 1 POSITION
001652  103374                  BCC     REG2            ;ALL DONE
001654  001406                  BEQ     REG2A
                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                ;           BRANCH INSTRUCTION AND         <====
                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                ;           FOLLOWING W/ 771               <====
001656  012702  000304          MOV     #$TESTN,R2      ;RESTORE POINTER
```

```
001662  012742  000025          MOV     #25,-(R2)       ;MOVE TO MAILBOX #  *******  25  *******
001666  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001670  000000                  HALT                    ;FAILURE WITH R2
001672  012702  000304  REG2A:   MOV     #$TESTN,R2              ;RESTORE POINTER

                        ;*****************************************************************************
                        ;TEST 22        TEST IF R2 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
001676  005212          TST22:  INC     (R2)            ;UPDATE TEST NUMBER
001700  022712  000022          CMP     #22,(R2)        ;SEQUENCE ERROR?
001704  001020                  BNE     TST23-10        ;BR TO ERROR HALT ON SEQ ERROR
001706  012702  177776          MOV     #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
001712  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
001716  000261                  SEC                     ;SET C-BIT
001720  005200  REG2B:   INC     R0              ;INCREMENT BIT COUNTER
001722  001407                  BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
001724  006102                  ROL     R2              ;ROTATE 1 POSITION
001726  103774                  BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
001730  022702  177777          CMP     #-1,R2          ;CHECK DATA IN R2
001734  001406                  BEQ     REG2C
001736  012702  000304          MOV     #$TESTN,R2      ;RESTORE POINTER
001742          R2ERR:
001742  012742  000026          MOV     #26,-(R2)       ;MOVE TO MAILBOX #  *******  26  *******
001746  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
001750  000000                  HALT                    ;FAILURE WITH R2
001752  012702  000304  REG2C:   MOV     #$TESTN,R2      ;RESTORE POINTER

                        ;*****************************************************************************
                        ;TEST 23        TEST IF R3 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
001756  005212          TST23:  INC     (R2)            ;UPDATE TEST NUMBER
001760  022712  000023          CMP     #23,(R2)        ;SEQUENCE ERROR?
001764  001012                  BNE     TST24-10        ;BR TO ERROR HALT ON SEQ ERROR
001766  012703  000001          MOV     #1,R3           ;SET BIT 0
001772  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
001776  000241                  CLC                     ;CLEAR C-BIT
002000  005200  REG3:    INC     R0              ;INCREMENT BIT COUNTER
002002  001403                  BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
002004  006103                  ROL     R3              ;ROTATE 1 POSITION
002006  103374                  BCC     REG3            ;ALL DONE
002010  001404                  BEQ     TST24

                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;           CONDITIONAL BRANCH INST. AND    <====
                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;           WHICH FOLLOWS W/ 766            <====

002012          REG3E:
002012  012742  000027          MOV     #27,-(R2)       ;MOVE TO MAILBOX #  *******  27  *******
002016  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002020  000000                  HALT                    ;FAILURE WITH R3
                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 24        TEST IF R3 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
002022  005212          TST24:  INC     (R2)            ;UPDATE TEST NUMBER
002024  022712  000024          CMP     #24,(R2)        ;SEQUENCE ERROR?
```



```
002030  001014                  BNE     TST25-10        ;BR TO ERROR HALT ON SEQ ERROR
002032  012703  177776          MOV     #-2,R3          ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
002036  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002042  000261                  SEC                     ;SET C-BIT
002044  005200  REG3A:   INC     R0              ;INCREMENT BIT COUNTER
002046  001405                  BEQ     R3ERR           ;BR TO ERROR HALT IF COUNTER=0
002050  006103                  ROL     R3              ;ROTATE 1 POSITION
002052  103774                  BCS     REG3A           ;CONTINUE UNTIL C-BIT IS CLEAR
002054  022703  177777          CMP     #-1,R3          ;CHECK DATA
002060  001404                  BEQ     TST25

                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;           CONDITIONAL BRANCH INST. AND    <====
                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;           WHICH FOLLOWS W/ 764            <====

002062          R3ERR:
002062  012742  000030          MOV     #30,-(R2)       ;MOVE TO MAILBOX #  *******  30  *******
002066  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002070  000000                  HALT                    ;FAILURE WITH R3
                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 25        TEST IF R4 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
002072  005212          TST25:  INC     (R2)            ;UPDATE TEST NUMBER
002074  022712  000025          CMP     #25,(R2)        ;SEQUENCE ERROR?
002100  001012                  BNE     TST26-10        ;BR TO ERROR HALT ON SEQ ERROR
002102  012704  000001          MOV     #1,R4           ;SET BIT 0
002106  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002112  000241                  CLC                     ;CLEAR C-BIT
002114  005200  REG4:    INC     R0              ;INCREMENT BIT COUNTER
002116  001403                  BEQ     REG4E           ;BR TO ERROR HALT IF BIT IS LOST
002120  006104                  ROL     R4              ;ROTATE 1 POSITION
002122  103374                  BCC     REG4            ;ALL DONE
002124  001404                  BEQ     TST26

                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;           CONDITIONAL BRANCH INST. AND    <====
                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;           WHICH FOLLOWS W/ 766            <====

002126          REG4E:
002126  012742  000031          MOV     #31,-(R2)       ;MOVE TO MAILBOX #  *******  31  *******
002132  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002134  000000                  HALT                    ;FAILURE WITH R4
                                ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 26        TEST IF R4 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
002136  005212          TST26:  INC     (R2)            ;UPDATE TEST NUMBER
002140  022712  000026          CMP     #26,(R2)        ;SEQUENCE ERROR?
002144  001014                  BNE     TST27-10        ;BR TO ERROR HALT ON SEQ ERROR
002146  012704  177776          MOV     #-2,R4          ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
002152  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002156  000261                  SEC                     ;SET C-BIT
002160  005200  REG4A:   INC     R0              ;INCREMENT BIT COUNTER
002162  001405                  BEQ     R4ERR           ;BR TO ERROR HALT IF COUNTER=0
002164  006104                  ROL     R4              ;ROTATE 1 POSITION
```

```
002166  103774                  BCS     REG4A           ;CONTINUE UNTIL C-BIT IS CLEAR
002170  022704  177777          CMP     #-1,R4          ;CHECK DATA
002174  001404                  BEQ     TST27
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 764            <====
002176                  R4ERR:
002176  012742  000032          MOV     #32,-(R2)       ;MOVE TO MAILBOX #  *******  32  *******
002202  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002204  000000                  HALT                    ;FAILURE WITH R4
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 27         TEST IF R5 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
002206  005212          TST27:  INC     (R2)            ;UPDATE TEST NUMBER
002210  022712  000027          CMP     #27,(R2)        ;SEQUENCE ERROR?
002214  001012                  BNE     TST30-10        ;BR TO ERROR HALT ON SEQ ERROR
002216  012705  000001          MOV     #1,R5           ;SET BIT 0
002222  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002226  000241                  CLC                     ;CLEAR C-BIT
002230  005200          REG5:   INC     R0              ;INCREMENT BIT COUNTER
002232  001403                  BEQ     REG5E           ;BR TO ERROR HALT IF BIT IS LOST
002234  006105                  ROL     R5              ;ROTATE 1 POSITION
002236  103774                  BCC     REG5            ;ALL DONE
002240  001404                  BEQ     TST30
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 766            <====
002242                  REG5E:
002242  012742  000033          MOV     #33,-(R2)       ;MOVE TO MAILBOX #  *******  33  *******
002246  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002250  000000                  HALT                    ;FAILURE WITH R5
                                                        ;  OR SEQUENCE ERROR


                        ;*****************************************************************************
                        ;TEST 30         TEST IF R5 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
002252  005212          TST30:  INC     (R2)            ;UPDATE TEST NUMBER
002254  022712  000030          CMP     #30,(R2)        ;SEQUENCE ERROR?
002260  001014                  BNE     TST31-10        ;BR TO ERROR HALT ON SEQ ERROR
002262  012705  177776          MOV     #-2,R5          ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
002266  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002272  000261                  SEC                     ;SET C-BIT
002274  005200          REG5A:  INC     R0              ;INCREMENT BIT COUNTER
002276  001405                  BEQ     R5ERR           ;BR TO ERROR HALT IF COUNTER=0
002300  006105                  ROL     R5              ;ROTATE 1 POSITION
002302  103774                  BCS     REG5A           ;CONTINUE UNTIL C-BIT IS C;EAR
002304  022705  177777          CMP     #-1,R5          ;CHECK DATA
002310  001404                  BEQ     TST31
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
```



```
                                                        ;           WHICH FOLLOWS W/ 764            <====
002312                  R5ERR:
002312  012742  000034          MOV     #34,-(R2)       ;MOVE TO MAILBOX #  *******  34  *******
002316  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002320  000000                  HALT                    ;FAILURE WITH R5
                                                        ;  OR SEQUENCE ERROR


                        ;*****************************************************************************
                        ;TEST 31         TEST IF R6 CAN HOLD A ONE IN ALL BITS
                        ;*****************************************************************************
002322  005212          TST31:  INC     (R2)            ;UPDATE TEST NUMBER
002324  022712  000031          CMP     #31,(R2)        ;SEQUENCE ERROR?
002330  001012                  BNE     TST32-10        ;BR TO ERROR HALT ON SEQ ERROR
002332  012706  000001          MOV     #1,R6           ;SET BIT 0
002336  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002342  000241                  CLC                     ;CLEAR C-BIT
002344  005200          REG6:   INC     R0              ;INCREMENT BIT COUNTER
002346  001403                  BEQ     REG6E           ;BR TO ERROR HALT IF BIT IS LOST
002350  006106                  ROL     R6              ;ROTATE 1 POSITION
002352  103774                  BCC     REG6            ;ALL DONE
002354  001404                  BEQ     TST32
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 766            <====
002356                  REG6E:
002356  012742  000035          MOV     #35,-(R2)       ;MOVE TO MAILBOX #  *******  35  *******
002362  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002364  000000                  HALT                    ;FAILURE WITH R6
                                                        ;  OR SEQUENCE ERROR


                        ;*****************************************************************************
                        ;TEST 32         TEST IF R6 CAN HOLD A ZERO IN ALL BITS
                        ;*****************************************************************************
002366  005212          TST32:  INC     (R2)            ;UPDATE TEST NUMBER
002370  022712  000032          CMP     #32,(R2)        ;SEQUENCE ERROR?
002374  001014                  BNE     TST33-10        ;BR TO ERROR HALT ON SEQ ERROR
002376  012706  177776          MOV     #-2,R6          ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
002402  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
002406  000261                  SEC                     ;SET C-BIT
002410  005200          REG6A:  INC     R0              ;INCREMENT BIT COUNT
002412  001405                  BEQ     R6ERR           ;BR TO ERROR HALT IF COUNTER=0
002414  006106                  ROL     R6              ;ROTATE 1 POSITION
002416  103774                  BCS     REG6A           ;CONTINUE UNTIL C-BIT IS CLEAR
002420  022706  177777          CMP     #-1,R6          ;CHECK DATA
002424  001404                  BEQ     TST33
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 764            <====
002426                  R6ERR:
002426  012742  000036          MOV     #36,-(R2)       ;MOVE TO MAILBOX #  *******  36  *******
002432  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002434  000000                  HALT                    ;FAILURE WITH R6
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                .SBTTL  PSW TESTS
                                ;
                                ;       THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
                                ;PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE
                                ;PSW ADDRESSING LOGIC IS FUNCTIONING.  MOVE AND COMPARE INSTRUCTIONS
                                ;ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.
                                ;EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR
                                ;SCOPING.
                                ;       THE PSW REGISTER ITSELF IS TESTED AS WELL AS THE ADDRESS
                                ;SELECT CIRCUITRY.  THE AMUX INPUTS TO THE PSW MUX ARE TESTED.  THE
                                ;CC INPUTS ARE TESTED LATER IN THE MICROCODE TESTS.  SETTING OF
                                ;THE T-BIT BY THE TEST PATTERNS IS PURPOSELY AVOIDED; TESTING OF THE
                                ;T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.
                                ;
                                ;*****************************************************************************
                                ;TEST 33         TEST IF PSW WILL HOLD ZEROES
                                ;*****************************************************************************
002436  005212                  TST33:  INC     (R2)            ;UPDATE TEST NUMBER
002440  022712  000033                  CMP     #33,(R2)        ;SEQUENCE ERROR?
002444  001010                          BNE     TST34-10        ;BR TO ERROR HALT ON SEQ ERROR
002446  012706  000500                  MOV     #STBOT,R6
002452  012737  000000  177776          MOV     #0,@#PS         ;SET PSW TO ZERO
002460  005737  177776                  TST     @#PS            ;SUCCESSFUL
002464  001404                          BEQ     TST34
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
002466  012742  000037                  MOV     #37,-(R2)       ;MOVE TO MAILBOX #  *******  37  *******
002472  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002474  000000                          HALT                    ;PSW NOT 0
                                                                ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 34         TEST IF PSW WILL HOLD ONES AND ZEROES
                                ;*****************************************************************************
002476  005212                  TST34:  INC     (R2)            ;UPDATE TEST NUMBER
002500  022712  000034                  CMP     #34,(R2)        ;SEQUENCE ERROR?
002504  001007                          BNE     TST35-10        ;BR TO ERROR HALT ON SEQ ERROR
002506  012737  000252  177776          MOV     #252,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
002514  023727  177776  000252          CMP     @#PS,#252       ;SUCCESSFUL?
002522  001404                          BEQ     TST35
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
002524  012742  000040                  MOV     #40,-(R2)       ;MOVE TO MAILBOX #  *******  40  *******
002530  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002532  000000                          HALT                    ;PSW NOT 252
                                                                ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;TEST 35         TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
                                ;*****************************************************************************
002534  005212                  TST35:  INC     (R2)            ;UPDATE TEST NUMBER
002536  022712  000035                  CMP     #35,(R2)        ;SEQUENCE ERROR?
002542  001007                          BNE     TST36-10        ;BR TO ERROR HALT ON SEQ ERROR
002544  012737  000105  177776          MOV     #105,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
002552  023727  177776  000105          CMP     @#PS,#105       ;SUCCESSFUL?
002560  001404                          BEQ     TST36
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
002562  012742  000041                  MOV     #41,-(R2)       ;MOVE TO MAILBOX #  *******  41  *******
002566  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002570  000000                          HALT                    ;PSW NOT 105
                                                                ;  OR SEQUENCE ERROR
                                ;
                                ;*****************************************************************************
                                ;TEST 36         TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
                                ;*****************************************************************************
002572  005212                  TST36:  INC     (R2)            ;UPDATE TEST NUMBER
002574  022712  000036                  CMP     #36,(R2)        ;SEQUENCE ERROR?
002600  001007                          BNE     TST37-10        ;BR TO ERROR HALT ON SEQ ERROR
002602  012737  000357  177776          MOV     #357,@#PS       ;MOVE ONES TO PSW
002610  023727  177776  000357          CMP     @#PS,#357       ;SUCCESSFUL
002616  001404                          BEQ     TST37
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
002620  012742  000042                  MOV     #42,-(R2)       ;MOVE TO MAILBOX #  *******  42  *******
002624  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002626  000000                          HALT                    ;PSW NOT 357
                                                                ;  OR SEQUENCE ERROR
```

```
                                .SBTTL  CONDITION CODE TEST

                                ;*****************************************************************************************
                                ;
                                ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.
                                ;THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
                                ;BEQ AND BNE ARE TESTED FOR PROPER EXECUTION.  THEN THE Z-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 37         TEST BRANCHES AROUND Z-BIT
                                ;*****************************************************************************************
002630  005212                  TST37:  INC     (R2)            ;UPDATE TEST NUMBER
002632  022712  000037                  CMP     #37,(R2)        ;SEQUENCE ERROR?
002636  001014                          BNE     .+TST40-.-10    ;BR TO ERROR HALT ON SEQ ERROR
                                        ;FIRST WITH Z-BIT ON
002640  000257                          CCC                     ;CC=0100: JUST Z-BIT
002642  000264                          SEZ
002644  001001                          BNE     BRZ1            ;CHECK OPPOSITE CONDITION
002646  001404                          BEQ     BRZ2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 774           <====
002650                          BRZ1:
002650  012742  000043                  MOV     #43,-(R2)       ;MOVE TO MAILBOX #  *******  43  *******
002654  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002656  000000                          HALT                    ;IMPROPER BR W/ Z=1
                                        ;CHECK WITH Z-BIT OFF
002660  000277                  BRZ2:   SCC                     ;CC=1011: ALL BUT Z-BIT
002662  000244                          CLZ
002664  001401                          BEQ     BRZ3
002666  001004                          BNE     .+TST40-.-4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
002670                          BRZ3:
002670  012742  000044                  MOV     #44,-(R2)       ;MOVE TO MAILBOX #  *******  44  *******
002674  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002676  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
                                ;*****************************************************************************************
002700  005212                  TST40:  INC     (R2)            ;UPDATE TEST NUMBER
002702  022712  000040                  CMP     #40,(R2)        ;SEQUENCE ERROR?
002706  001014                          BNE     .+TST41-.-10    ;BR TO ERROR HALT ON SEQ ERROR
                                        ;FIRST WITH N-BIT ON
002710  000257                          CCC                     ;CC=1000: JUST N-BIT
002712  000270                          SEN
002714  100001                          BPL     BRN1            ;CHECK OPPOSITE CONDITION
002716  100404                          BMI     BRN2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 774           <====
002720                          BRN1:
002720  012742  000045                  MOV     #45,-(R2)       ;MOVE TO MAILBOX #  *******  45  *******
002724  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002726  000000                          HALT                    ;IMPROPER BR W/ N=1
                                        ;CHECK WITH N-BIT OFF
002730  000277                  BRN2:   SCC                     ;CC=0111
002732  000250                          CLN
002734  100401                          BMI     BRN3            ;CHECK OPPOSITE CONDITION
002736  100004                          BPL     .+TST41-.-4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 764           <====
002740                          BRN3:
002740  012742  000046                  MOV     #46,-(R2)       ;MOVE TO MAILBOX #  *******  46  *******
002744  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002746  000000                          HALT                    ;IMPROPER BR W/ N=0
                                                                ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.
                        ;THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
                        ;BVS AND BVC ARE TESTED FOR PROPER EXECUTION.  THEN THE V-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 41        TEST BRANCHES AROUND V-BIT
                        ;*****************************************************************************
002750  005212         TST41:  INC     (R2)            ;UPDATE TEST NUMBER
002752  022712  000041         CMP     #41,(R2)        ;SEQUENCE ERROR?
002756  001014                 BNE     TST42-10        ;BR TO ERROR HALT ON SEQ ERROR
                                ;FIRST WITH V-BIT ON
002760  000257                 CCC                     ;CC=0010: JUST V-BIT
002762  000262                 SEV
002764  102001                 BVC     BRV1            ;CHECK OPPOSITE CONDITION
002766  102404                 BVS     BRV2
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 774           <====
002770                  BRV1:
002770  012742  000047         MOV     #47,-(R2)       ;MOVE TO MAILBOX #  *******  47  *******
002774  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
002776  000000                 HALT                    ;IMPROPER BR W/ V=1
                                ;CHECK WITH V-BIT OFF
003000  000277         BRV2:   SCC
003002  000242                 CLV                     ;CC=1101: ALL BVT V-BIT
003004  102401                 BVS     BRV3            ;CHECK OPPOSITE CONDITION
003006  102004                 BVC     TST42
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 764           <====
003010                  BRV3:
003010  012742  000050         MOV     #50,-(R2)       ;MOVE TO MAILBOX #  *******  50  *******
003014  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003016  000000                 HALT                    ;IMPROPER BR W/ V=0
                                                       ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.
                        ;THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
                        ;BCS AND BCC ARE TESTED FOR PROPER EXECUTION.  THEN THE C-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 42        TEST BRANCHES AROUND C-BIT
                        ;*****************************************************************************
003020  005212         TST42:  INC     (R2)            ;UPDATE TEST NUMBER
003022  022712  000042         CMP     #42,(R2)        ;SEQUENCE ERROR?
003026  001014                 BNE     TST43-10        ;BR TO ERROR HALT ON SEQ ERROR
                                ;FIRST WITH C-BIT ON
003030  000257                 CCC                     ;CC=0001: JUST C-BIT
003032  000261                 SEC
003034  103001                 BCC     BRC1            ;CHECK OPPOSITE CONDITION
003036  103404                 BCS     BRC2
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 774           <====
003040                  BRC1:
003040  012742  000051         MOV     #51,-(R2)       ;MOVE TO MAILBOX #  *******  51  *******
003044  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003046  000000                 HALT                    ;IMPROPER BR W/ C=1
003050  000277                 ;CHECK WITH C-BIT OFF
003052  000241         BRC2:   SCC
003054  103401                 CLC                     ;CC=1110
003056  100404                 BCS     BRC3            ;CHECK OPPOSITE CONDITION
                               BMI     TST43
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 764           <====
003060                  BRC3:
003060  012742  000052         MOV     #52,-(R2)       ;MOVE TO MAILBOX #  *******  52  *******
003064  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003066  000000                 HALT                    ;IMPROPER BR W/ C=0
                                                       ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************************
                        .SBTTL  MICROCODE TESTS
                        ;
                        ;       THE MICROCODE TESTS ARE USED TO VERIFY THE MICROPROGRAMM
                        ;FLOW.  THE GOAL OF THESE TESTS IS TO EXERCISE EVERY POSSIBLE
                        ;BRANCH IN THE MICROPROGRAM FLOW.
                        ;       THE TEST EXERCISES EVERY BRANCH IN THE MICROCODE BY
                        ;TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
                        ;ALL POSSIBLE MODES.  FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
                        ;AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
                        ;ADDRESSING MODES.  BYTE MODES ARE ALSO TESTED.  AS EACH NEW
                        ;MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
                        ;A SMALL LOOP CONVENIENT FOR SCOPING.  THE TEST IS SET UP USING
                        ;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
                        ;VERIFIED.
                        ;       IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
                        ;FAULT.
                        ;
                        ;*****************************************************************************************



                        ;*****************************************************************************************
                        ;
                        ;       THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
                        ;MODE WITH THE SINGLE OPERAND INSTRUCTION.  FOLLOWING THE SEQUENCE CHECK,
                        ;THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
                        ;CHECKS THAT THE Z-BIT WAS PROPERLY SET.  THIS SMALL TEST IS SELF-SUFFICIENT
                        ;AND CAN BE SCOPED TO TROUBLE SHOOT ALL OF THE IR DECODE LOGIC 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.
                        ;*****************************************************************************************
003070  005212          TST43:  INC     (R2)            ;UPDATE TEST NUMBER
003072  022712  000043          CMP     #43,(R2)        ;SEQUENCE ERROR?
003076  001020                  BNE     TST44-10        ;BR TO ERROR HALT ON SEQ ERROR
003100  005000                  CLR     R0              ;TRY THE CLEAR INST.
003102  001404                  BEQ     SOP0A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 776           <====
003104  012742  000053          MOV     #53,-(R2)       ;MOVE TO MAILBOX #  *******  53  *******
003110  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003112  000000                  HALT                    ;CLR DID NOT SET Z-BIT
003114  005200          SOP0A:  INC     R0              ;TRY THE INCREMENT INST.
003116  005100                  COM     R0              ;TRY COMPLEMENT
003120  005200                  INC     R0
003122  100404                  BMI     SOP0B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```



```
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
003124  012742  000054          MOV     #54,-(R2)       ;MOVE TO MAILBOX #  *******  54  *******
003130  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003132  000000                  HALT                    ;NEGATE DID NOT SET N-BIT
003134  005100          SOP0B:  COM     R0              ;TRY COMPLEMENT INST.
003136  001404                  BEQ     TST44
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
003140  012742  000055          MOV     #55,-(R2)       ;MOVE TO MAILBOX #  *******  55  *******
003144  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003146  000000                  HALT                    ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
                                                        ;  OR SEQUENCE ERROR



                        ;*****************************************************************************************
                        ;
                        ;       THIS TEST INTRODUCES THE REMAINING SOP INSTRUCIONS AND TESTS
                        ;THEM IN MODE 0.  THE PURPOSE IS TO PROVIDE A BASELINE OF
                        ;INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS.  SINCE THE MICROCODE FOR
                        ;THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
                        ;SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
                        ;FUNCTIONING.
                        ;
                        ;*****************************************************************************************
                        ;TEST 44        TEST REMAINDER OF SOP INSTS IN MODE 0
                        ;*****************************************************************************************
003150  005212          TST44:  INC     (R2)            ;UPDATE TEST NUMBER
003152  022712  000044          CMP     #44,(R2)        ;SEQUENCE ERROR?
003156  001021                  BNE     TST45-10        ;BR TO ERROR HALT ON SEQ ERROR
003160  005000                  CLR     R0              ;INITIALIZE
003162  005300                  DEC     R0              ;TRY DECREMENT INST.
003164  100404                  BMI     SOP0C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 775           <====
003166  012742  000056          MOV     #56,-(R2)       ;MOVE TO MAILBOX #  *******  56  *******
003172  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003174  000000                  HALT                    ;N-BIT NOT SET ON DEC
003176  000261          SOP0C:  SEC                     ;INITIALIZE CARRY
003200  005500                  ADC     R0              ;TRY ADD CARRY INST
003202  001007                  BNE     SOP0D
003204  000261                  SEC                     ;INITIALIZE CARRY
003206  005600                  SBC     R0              ;TRY SUBTRACT-CARRY INST
003210  100004                  BPL     SOP0D
003212  005100                  COM     R0
003214  005200                  INC     R0
003216  005300                  DEC     R0
003220  001404                  BEQ     TST45
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
1175                                                ;           WHICH FOLLOWS W/ 757          <====
1176  003222                        SOPOD:
1177  003222  012742  000057                MOV     #57,-(R2)   ;MOVE TO MAILBOX #  *******  57  *******
1178  003226  005242                        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
1179  003230  000000                        HALT                ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F
1180                                                            ;  OR SEQUENCE ERROR
```



```
1181
1182                                                ;*******************************************************************************
1183                                                ;
1184                                                ;       THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
1185                                                ;THE MODE 0 BYTE MICROCODE IS TESTED.  THE METHOD AND SEQUENCE
1186                                                ;OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
1187                                                ;
1188                                                ;*******************************************************************************
1189                                                ;TEST 45        TEST MODE 0 EVEN BYTE USING SOP INST
1190                                                ;*******************************************************************************
1191  003232  005212                        TST45:  INC     (R2)        ;UPDATE TEST NUMBER
1192  003234  022712  000045                        CMP     #45,(R2)    ;SEQUENCE ERROR?
1193  003240  001012                        BNE     TST46-10    ;BR TO ERROR HALT ON SEQ ERROR
1194  003242  105000                        CLRB    R0          ;TRY CLEARING EVEN BYTE OF REGISTER
1195  003244  001404                        BEQ     SOPB0A
1196                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
1197                                                            ;           CONDITIONAL BRANCH INST. AND    <====
1198                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
1199                                                            ;           WHICH FOLLOWS W/ 776            <====
1200  003246  012742  000060                        MOV     #60,-(R2)   ;MOVE TO MAILBOX #  *******  60  *******
1201  003252  005242                        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
1202  003254  000000                        HALT                ;CLRB DID NOT SET Z-BIT
1203  003256  105100                SOPB0A: COMB    R0          ;TRY SETTING EVEN BYTE OF REGISTER
1204  003260  100002                        BPL     SOPB0B
1205  003262  105200                        INCB    R0          ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
1206  003264  001404                        BEQ     TST46
1207                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
1208                                                            ;           CONDITIONAL BRANCH INST. AND    <====
1209                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
1210                                                            ;           WHICH FOLLOWS W/ 766            <====
1211  003266                        SOPB0B:
1212  003266  012742  000061                MOV     #61,-(R2)   ;MOVE TO MAILBOX #  *******  61  *******
1213  003272  005242                        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
1214  003274  000000                        HALT                ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
1215                                                            ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
                        ;SINGLE OPERAND MODE 1 INSTRUCTIONS.  AGAIN, THE CLR INSTRUCTION
                        ;IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
                        ;CONDITION CODES ARE SET.  OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
                        ;COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
                        ;
                        ;*****************************************************************************
                        ;TEST 46        TEST MODE 1 USING SOP INST.
                        ;*****************************************************************************
003276  005212          TST46:  INC     (R2)            ;UPDATE TEST NUMBER
003300  022712  000046          CMP     #46,(R2)        ;SEQUENCE ERROR?
003304  001014                  BNE     TST47-10        ;BR TO ERROR HALT ON SEQ ERROR
003306  005000                  CLR     R0              ;INITIALIZE R0
003310  005010                  CLR     (R0)            ;TRY CLEAR INST W/MODE 1
003312  001404                  BEQ     SOP1A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 775           <====
003314  012742  000062          MOV     #62,-(R2)       ;MOVE TO MAILBOX #  *******  62  *******
003320  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003322  000000                  HALT                    ;CLR DID NOT SET Z-BIT
003324  005310          SOP1A:  DEC     (R0)            ;TRY DECREMENT INST W/MODE 1
003326  100003                  BPL     SOP1B
003330  000261                  SEC                     ;INITIALIZE CARRY
003332  005510                  ADC     (R0)            ;TRY ADD-CARRY W/MODE 1
003334  001404                  BEQ     TST47
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
003336          SOP1B:
003336  012742  000063          MOV     #63,-(R2)       ;MOVE TO MAILBOX #  *******  63  *******
003342  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003344  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
                        ;SINGLE OPERAND INSTRUCTIONS.
                        ;       THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
                        ;AND VERIFIED.
                        ;
                        ;*****************************************************************************
                        ;TEST 47        TEST MODE 1 EVEN BYTE USING SOP INST
                        ;*****************************************************************************
003346  005212          TST47:  INC     (R2)            ;UPDATE TEST NUMBER
003350  022712  000047          CMP     #47,(R2)        ;SEQUENCE ERROR?
003354  001020                  BNE     TST50-10        ;BR TO ERROR HALT ON SEQ ERROR
003356  005000                  CLR     R0              ;INITIALIZE R0
003360  005010                  CLR     (R0)            ;INITIALIZE LOC. 0
003362  005110                  COM     (R0)
003364  105010                  CLRB    (R0)            ;TRY TO CLEAR BYTE 0
003366  001404                  BEQ     SOPB1A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 773           <====
003370  012742  000064          MOV     #64,-(R2)       ;MOVE TO MAILBOX #  *******  64  *******
003374  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003376  000000                  HALT                    ;CLRB DID NOT SET Z-BIT
003400  005210          SOPB1A: INC     (R0)            ;INCREMENT TO TEST WORD
003402  100005                  BPL     SOPB1B
003404  105110                  COMB    (R0)            ;COMPLEMENT:  ODD BYTE = 376
003406  105210                  INCB    (R0)            ;INC:  ODD BYTE = 377
003410  100002                  BPL     SOPB1B
003412  105210                  INCB    (R0)            ;INCREMENT ODD BYTE=0
003414  001404                  BEQ     TST50
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
003416          SOPB1B:
003416  012742  000065          MOV     #65,-(R2)       ;MOVE TO MAILBOX #  *******  65  *******
003422  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003424  000000                  HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       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
                                ;*****************************************************************************
003426  005212          TST50:  INC     (R2)            ;UPDATE TEST NUMBER
003430  022712  000050          CMP     #50,(R2)        ;SEQUENCE ERROR?
003434  001022                  BNE     TST51-10        ;BR TO ERROR HALT ON SEQ ERROR
003436  005000                  CLR     R0              ;INITIALIZE R0
003440  005010                  CLR     (R0)            ;INITIALIZE LOC. 0
003442  005110                  COM     (R0)
003444  005200                  INC     R0              ;R0=ODD BYTE
003446  105010                  CLRB    (R0)            ;TRY TO CLEAR BYTE 1
003450  001404                  BEQ     SOPB1C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
003452  012742  000066          MOV     #66,-(R2)       ;MOVE TO MAILBOX #  *******  66  *******
003456  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003460  000000                  HALT                    ;CLRB DID NOT SET Z-BIT
003462  005300          SOPB1C: DEC     R0              ;R0=WORD ADDR.
003464  005210                  INC     (R0)            ;INCREMENT TO TEST WORD
003466  005200                  INC     R0              ;R0=ODD BYTE
003470  105110                  COMB    (R0)            ;TRY TO COMPLEMENT BYTE 1
003472  105210                  INCB    (R0)
003474  100002                  BPL     SOPB1D
003476  105210                  INCB    (R0)            ;TRY TO INCREMENT BYTE 1
003500  001404                  BEQ     TST51
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
003502          SOPB1D:
003502  012742  000067          MOV     #67,-(R2)       ;MOVE TO MAILBOX #  *******  67  *******
003506  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003510  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.
                                ;*****************************************************************************
003512  005212          TST51:  INC     (R2)            ;UPDATE TEST NUMBER
003514  022712  000051          CMP     #51,(R2)        ;SEQUENCE ERROR?
003520  001023                  BNE     TST52-10        ;BR TO ERROR HALT ON SEQ ERROR
003522  005000                  CLR     R0              ;SET R0=400
003524  105100                  COMB    R0
003526  005200                  INC     R0
003530  005010                  CLR     (R0)            ;CLEAR 400
003532  005110                  COM     (R0)            ;INITIALIZE: 400=-1
003534  005020                  CLR     (R0)+           ;TRY CLEARING WITH MODE 2
003536  001404                  BEQ     SOPZA
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
003540  012742  000070          MOV     #70,-(R2)       ;MOVE TO MAILBOX #  *******  70  *******
003544  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003546  000000                  HALT                    ;CLR INST DID NOT SET Z-BIT
003550  005300          SOPZA:  DEC     R0              ;RESET R0
003552  005300                  DEC     R0
003554  005120                  COM     (R0)+           ;TRY COMPLEMENTING WITH MODE 2
003556  100004                  BPL     SOP2B
003560  005300                  DEC     R0              ;RESET R0
003562  005300                  DEC     R0
003564  005220                  INC     (R0)+           ;TRY INCREMENTING WITH MODE 2
003566  001404                  BEQ     TST52
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 755           <====
003570          SOP2B:
003570  012742  000071          MOV     #71,-(R2)       ;MOVE TO MAILBOX #  *******  71  *******
003574  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003576  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.
                                ;*****************************************************************************
003600  005212                  TST52:  INC     (R2)            ;UPDATE TEST NUMBER
003602  022712  000052                  CMP     #52,(R2)        ;SEQUENCE ERROR?
003606  001023                          BNE     TST53-10        ;BR TO ERROR HALT ON SEQ ERROR
003610  005000                          CLR     R0              ;SET R0=400
003612  105100                          COMB    R0
003614  005200                          INC     R0
003616  005010                          CLR     (R0)            ;CLEAR 400
003620  005110                          COM     (R0)            ;INITIALIZE: 400=-1
003622  105020                          CLRB    (R)+            ;TRY TO CLEAT 400 W/MODE 2
003624  001404                          BEQ     SOPB2A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
003626  012742  000072                          MOV     #72,-(R2)       ;MOVE TO MAILBOX #  *******  72  *******
003632  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003634  000000                          HALT                    ;CLR DID NOT SET Z-BIT
003636  005300                  SOPB2A: DEC     R0              ;RESULT R0=400
003640  005210                          INC     (R0)            ;INC 400 TO TEST WORD
003642  105110                          COMB    (R0)
003644  105220                          INCB    (R)+            ;TRY TO INC EVEN BYTE
003646  100003                          BPL     SOPB2B
003650  005300                          DEC     R0              ;RESET R0=400
003652  105220                          INCB    (R)+            ;TRY INCREMENT OF EVEN BYTE
003654  001404                          BEQ     TST53
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 755           <====
003656                          SOPB2B:
003656  012742  000073                  MOV     #73,-(R2)       ;MOVE TO MAILBOX #  *******  73  *******
003662  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003664  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
                                                                ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
                                ;TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
                                ;
                                ;*****************************************************************************
                                ;TEST 53         TEST MODE 2 ODD BYTE USING SOP INST.
                                ;*****************************************************************************
003666  005212                  TST53:  INC     (R2)            ;UPDATE TEST NUMBER
003670  022712  000053                  CMP     #53,(R2)        ;SEQUENCE ERROR?
003674  001026                          BNE     TST54-10        ;BR TO ERROR HALT ON SEQ ERROR
003676  005000                          CLR     R0              ;SET R0=400
003700  105100                          COMB    R0
003702  005200                          INC     R0
003704  005010                          CLR     (R0)            ;CLEAR LOC 400
003706  005110                          COM     (R0)            ;INITIALIZE: 400=-1
003710  005200                          INC     R0              ;R0=ODD BYTE
003712  105020                          CLRB    (R)+            ;TRY TO CLEAR ODD BYTE
003714  001404                          BEQ     SOPB2C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
003716  012742  000074                          MOV     #74,-(R2)       ;MOVE TO MAILBOX #  *******  74  *******
003722  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003724  000000                          HALT                    ;CLRB DID NOT SET Z-BIT
003726  005300                  SOPB2C: DEC     R0              ;R0=WORD ADDR.
003730  005300                          DEC     R0
003732  005220                          INC     (R0)+           ;INCREMENT WORD
003734  005300                          DEC     R0              ;POINT TO ODD BYTE
003736  105110                          COMB    (R0)            ;COMPLEMENT ODD BYTE
003740  105220                          INCB    (R)+            ;TRY TO INCREMENT ODD BYTE
003742  100003                          BPL     SOPB2D
003744  005300                          DEC     R0              ;RESET R0 TO ODD BYTE
003746  105220                          INCB    (R)+            ;TRY TO INCREMENT ODD BYTE
003750  001404                          BEQ     TST54
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 752           <====
003752                          SOPB2D:
003752  012742  000075                  MOV     #75,-(R2)       ;MOVE TO MAILBOX #  *******  75  *******
003756  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
003760  000000                          HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INST.
                                                                ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES.  PREVIOUSLY
                        ;TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
                        ;
                        ;*****************************************************************************
                        ;TEST 54         TEST MODE 0 USING NEGATE INSTRUCTION
                        ;*****************************************************************************
003762  005212          TST54:  INC     (R2)            ;UPDATE TEST NUMBER
003764  022712  000054          CMP     #54,(R2)        ;SEQUENCE ERROR?
003770  001035                  BNE     TST55-10        ;BR TO ERROR HALT ON SEQ ERROR
003772  005000                  CLR     R0              ;SET R0=0
003774  005200                  INC     R0              ;   R0=1
003776  005400                  NEG     R0              ;TRY NEGATE MODE 0:  R0=-1
004000  100003                  BPL     NEG00           ;CC=1001?
004002  001402                  BEQ     NEG00
004004  102401                  BVS     NEG00
004006  103404                  BCS     NEG01

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====

004010                  NEG00:
004010  012742  000076          MOV     #76.,-(R2)      ;MOVE TO MAILBOX #  *******  76  *******
004014  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004016  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY

004020  005200          NEG01:  INC     R0              ;TEST DATA RESULT
004022  001404                  BEQ     NEG02

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====

004024  012742  000077          MOV     #77.,-(R2)      ;MOVE TO MAILBOX #  *******  77  *******
004030  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004032  000000                  HALT                    ;DATA RESULT OF NEGATE INCORRECT

004034  105100          NEG02:  COMB    R0              ;R0=377
004036  105400                  NEGB    R0              ;R0=1
004040  100403                  BMI     NEG03           ;CC=0001?
004042  001402                  BEQ     NEG03
004044  102401                  BVS     NEG03
004046  103404                  BCS     NEG04

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 751           <====

004050                  NEG03:
004050  012742  000100          MOV     #100.,-(R2)     ;MOVE TO MAILBOX #  *******  100  *******
004054  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004056  000000                  HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
004060  005300          NEG04:  DEC     R0              ;TEST DATA RESULT
004062  001404                  BEQ     TST55

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
```



```
                                                        ;           WHICH FOLLOWS W/ 743           <====
004064  012742  000101          MOV     #101.,-(R2)     ;MOVE TO MAILBOX #  *******  101  *******
004070  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004072  000000                  HALT                    ;DATA RESULT OF NEGB INCORRECT
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 55         TEST MODE 1 USING NEGATE INST.
                        ;*****************************************************************************
004074  005212          TST55:  INC     (R2)            ;UPDATE TEST NUMBER
004076  022712  000055          CMP     #55.,(R2)       ;SEQUENCE ERROR?
004102  001040                  BNE     TST56-10        ;BR TO ERROR HALT ON SEQ ERROR
004104  005000                  CLR     R0              ;POINT TO LOC. 0
004106  005010                  CLR     (R0)            ;CLEAR LOC. 0
004110  005210                  INC     (R0)            ;LOC. 0=1
004112  005410                  NEG     (R0)            ;TRY NEG. LOC. 0=-1
004114  100003                  BPL     NEG10           ;CC=1001
004116  001402                  BEQ     NEG10
004120  102401                  BVS     NEG10
004122  103404                  BCS     NEG11

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====

004124                  NEG10:
004124  012742  000102          MOV     #102.,-(R2)     ;MOVE TO MAILBOX #  *******  102  *******
004130  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004132  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY

004134  005237  000000  NEG11:  INC     @#0             ;TEST DATA RESULT
004140  001404                  BEQ     NEG12

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====

004142  012742  000103          MOV     #103.,-(R2)     ;MOVE TO MAILBOX #  *******  103  *******
004146  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004150  000000                  HALT                    ;DATA RESULT OF NEGATE INCORRECT
004152  105110          NEG12:  COMB    (R0)            ;LOC. 0=377
004154  105410                  NEGB    (R0)            ;TRY NEGB LOC. 0=1
004156  100403                  BMI     NEG13           ;CC=0001?
004160  001402                  BEQ     NEG13
004162  102401                  BVS     NEG13
004164  103404                  BCS     NEG14

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 747           <====

004166                  NEG13:
004166  012742  000104          MOV     #104.,-(R2)     ;MOVE TO MAILBOX #  *******  104  *******
004172  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004174  000000                  HALT                    ;NEGB DID NOT SET CC'S CORRECTLY
004176  005337  000000  NEG14:  DEC     @#0             ;TEST DATA RESULT
004202  001404                  BEQ     TST56

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
1599
1600  004204  012742  000105          MOV     #105,-(R2)      ;          WHICH FOLLOWS W/ 740      <====
1601  004210  005242                  INC     -(R2)           ;MOVE TO MAILBOX #  *******  105  *******
1602  004212  000000                  HALT                    ;SET MSGTYP TO FATAL ERROR
1603                                                          ;DATA RESULT OF NEGB INCORRECT
                                                              ;  OR SEQUENCE ERROR
```



```
1604                                  ;*************************************************************************************
1605                                  ;TEST 56          TEST MODE 2 USING NEGATE INSTRUCTION
1606                                  ;*************************************************************************************
1607  004214  005212          TST56:  INC     (R2)            ;UPDATE TEST NUMBER
1608  004216  022712  000056          CMP     #56,(R2)        ;SEQUENCE ERROR?
1609  004222  001032                  BNE     .+TST57-10      ;BR TO ERROR HALT ON SEQ ERROR
1610  004224  005000                  CLR     R0              ;POINT TO LOC. 0
1611  004226  005010                  CLR     (R0)            ;CLEAR LOC. 0
1612  004230  005210                  INC     (R0)            ;LOC. 0=1
1613  004232  005420                  NEG     (R0)+           ;TRY NEG.: LOC. 0=-1
1614  004234  100003                  BPL     NEG20           ;CC=1001?
1615  004236  001402                  BEQ     NEG20
1616  004240  102401                  BVS     NEG20
1617  004242  103404                  BCS     NEG21
1618
1619                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1620                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1621                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1622  004244                          NEG20:                  ;           WHICH FOLLOWS W/ 770           <====
1623  004244  012742  000106                  MOV     #106,-(R2)      ;MOVE TO MAILBOX #  *******  106  *******
1624  004250  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1625  004252  000000                  HALT                    ;NEGATE DID NOT SET CC'S CORRECTLY
1626  004254  105300          NEG21:  DECB    R0              ;R0=LOC. 0
1627  004256  105300                  DECB    R0
1628  004260  105420                  NEGB    (R0)+           ;BYTE 0=1    R0=1
1629  004262  105420                  NEGB    (R0)+           ;BYTE 1=1    R0=2
1630  004264  105340                  DECB    -(R0)           ;R0=1  LOC. 0=01
1631  004266  005300                  DEC     R0              ;R0=0
1632  004270  001404                  BEQ     NEG22
1633
1634                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1635                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1636                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1637  004272  012742  000107                                ;           WHICH FOLLOWS W/ 755           <====
1638  004276  005242                  MOV     #107,-(R2)      ;MOVE TO MAILBOX #  *******  107  *******
1639  004300  000000                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1640  004302  005337  000000          HALT                    ;REGISTER NOT INCREMENTED CORRECTLY
1641  004306  001404          NEG22:  DEC     @#0             ;LOC. 0=0
1642                                  BEQ     TST57
1643
1644                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1645                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1646  004310  012742  000110                                ;           REPLACE THE MOVE INSTRUCTION   <====
1647  004314  005242                  MOV     #110,-(R2)      ;           WHICH FOLLOWS W/ 746           <====
1648  004316  000000                  INC     -(R2)           ;MOVE TO MAILBOX #  *******  110  *******
1649                                  HALT                    ;SET MSGTYP TO FATAL ERROR
                                                              ;NEG BYTE INSTRUCTIONS FAILED
                                                              ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS.  IT
                        ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 400
                        ;THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
                        ;INSTRUCTIONS UNDER TEST.
                        ;       R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
                        ;INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0.  THEN R0
                        ;IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
                        ;LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST.  THE PROPER INCREMENTING
                        ;OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
                        ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
                        ;(LOC. 400-402) HAS THE PROPER VALUES (0).
                        ;
                        ;*****************************************************************************
                        ;TEST 57        TEST MODE 3 USING SOP INST.
                        ;*****************************************************************************
004320  005212          TST57:  INC     (R2)            ;UPDATE TEST NUMBER
004322  022712  000057          CMP     #57,(R2)        ;SEQUENCE ERROR?
004326  001020                  BNE     TST60-10        ;BR TO ERROR HALT ON SEQ ERROR
004330  005000                  CLR     R0              ;SET R0=400
004332  105100                  COMB    R0
004334  005200                  INC     R0
004336  005010                  CLR     (R0)            ;CLEAR LOC 400
004340  005030                  CLR     @(R0)+          ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402
004342  001404                  BEQ     SOP3A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
004344  012742  000111                  MOV     #111,-(R2)      ;MOVE TO MAILBOX #  *******  111  *******
004350  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004352  000000                  HALT                    ;CLR DID NOT SET Z-BIT
004354  005300          SOP3A:  DEC     R0              ;RESET R0=400
004356  005300                  DEC     R0
004360  005130                  COM     @(R0)+          ;TRY TO COMPLEMENT LOC 0 OF MODE 3  ;R0=402
004362  100002                  BPL     SOP3B
004364  005230                  INC     @(R0)+          ;TRY TO INCREMENT LOC 0 W/MODE 3  ;R0=404
004366  001404                  BEQ     TST60
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
004370          SOP3B:
004370  012742  000112                  MOV     #112,-(R2)      ;MOVE TO MAILBOX #  *******  112  *******
004374  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004376  000000                  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.
                        ;*****************************************************************************
004400  005212          TST60:  INC     (R2)            ;UPDATE TEST NUMBER
004402  022712  000060          CMP     #60,(R2)        ;SEQUENCE ERROR?
004406  001026                  BNE     TST61-10        ;BR TO ERROR HALT ON SEQ ERROR
004410  005004                  CLR     R4              ;SET R4=400
004412  105104                  COMB    R4
004414  005204                  INC     R4
004416  005000                  CLR     R0              ;INITIALIZE LOC. 0=-1
004420  005010                  CLR     (R0)
004422  005110                  COM     (R0)            ;LOC. 0=-1
004424  105034                  CLRB    @(R4)+          ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400  R4=402
004426  001404                  BEQ     SOPB3A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
004430  012742  000113                  MOV     #113,-(R2)      ;MOVE TO MAILBOX #  *******  113  *******
004434  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004436  000000                  HALT                    ;CLRB DID NOT SET Z-BIT
004440  005304          SOPB3A: DEC     R4              ;RESET POINTER  R4=400
004442  005304                  DEC     R4
004444  005234                  INC     @(R4)+          ;TRY INCREMENTING WORD  LOC.0=177401  R4=402
004446  100006                  BPL     SOPB3B
004450  105434                  NEGB    @(R4)+          ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
004452  100004                  BPL     SOPB3B
004454  005304                  DEC     R4              ;R4=402
004456  005304                  DEC     R4
004460  105234                  INCB    @(R4)+          ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
004462  001404                  BEQ     TST61
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
004464          SOPB3B:
004464  012742  000114                  MOV     #114,-(R2)      ;MOVE TO MAILBOX #  *******  114  *******
004470  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004472  000000                  HALT                    ;CUMMULATIVE RESULT OF ABOVE INST FAILED
                                                        ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
                        ;WHICH ADDRESS ODD BYTES.  THE TARGET IS BYTE 1.  A TABLE AT
                        ;LOC. 400-406 IS USED.  R0 SERVES AS THE TABLE POINTER.
                        ;       R0 IS INITIALIZED TO 400.  LOC. 0 IS SET TO -1 USING THE
                        ;FIRST TWO TABLE ENTRIES.  A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING
                        ;TABLE ADDRESS AT 404.  R0 IS DECREMENTED TO 402 AND SEVERAL SOP
                        ;MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER
                        ;REGISTER INCREMENTING.
                        ;       THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND
                        ;AFTER THE TEST IS RUN.
                        ;
                        ;*****************************************************************************
                        ;TEST 61         TEST MODE 3 ODD BYTE USING SOP INST.
                        ;*****************************************************************************
004474  005212         TST61:  INC     (R2)            ;UPDATE TEST NUMBER
004476  022712  000061         CMP     #61,(R2)        ;SEQUENCE ERROR?
004502  001024                 BNE     TST62-10        ;BR TO ERROR HALT ON SEQ ERROR
004504  005000                 CLR     R0              ;SET R0=400
004506  105100                 COMB    R0
004510  005200                 INC     R0
004512  005030                 CLR     @(R0)+          ;INITIALIZE
004514  005130                 COM     @(R0)+          ;LOC 0=-1 R0=404
004516  105030                 CLRB    @(R0)+          ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406
004520  001404                 BEQ     SOPB3C

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 771            <====

004522  012742  000115         MOV     #115,-(R2)      ;MOVE TO MAILBOX #  *******  115  *******
004526  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004530  000000                 HALT                    ;CLRB DID NOT SET Z-BIT
004532  005300         SOPB3C: DEC     R0              ;RESET R0=402
004534  005300                 DEC     R0
004536  005300                 DEC     R0              ;POINT TO EVEN BYTE ADDR.
004540  005300                 DEC     R0
004542  005230                 INC     @(R0)+          ;INCREMENT WORD LOC. 0=400 R0=404
004544  105430                 NEGB    @(R0)+          ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406
004546  100002                 BPL     SOPB3D
004550  105230                 INCB    @(R0)+          ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410
004552  001404                 BEQ     TST62

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 754            <====

004554         SOPB3D:
004554  012742  000116         MOV     #116,-(R2)      ;MOVE TO MAILBOX #  *******  116  *******
004560  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004562  000000                 HALT                    ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED
                                                       ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;TEST 62         TEST MODE 3 USING NEGATE INSTRUCTION
                        ;*****************************************************************************
004564  005212         TST62:  INC     (R2)            ;UPDATE TEST NUMBER
004566  022712  000062         CMP     #62,(R2)        ;SEQUENCE ERROR?
004572  001054                 BNE     TST63-10        ;BR TO ERROR HALT ON SEQ ERROR
004574  005000                 CLR     R0              ;R0=400
004576  105100                 COMB    R0
004600  005200                 INC     R0
004602  005010                 CLR     (R0)            ;LOC. 400=0
004604  005004                 CLR     R4              ;R4=0
004606  005014                 CLR     (R4)            ;LOC. 0=0
004610  005214                 INC     (R4)            ;LOC. 0=1
004612  005430                 NEG     @(R0)+          ;TRY NEGATE   LOC. 0=-1  R0=402
004614  100003                 BPL     NEG30           ;CC=1001?
004616  001402                 BEQ     NEG30
004620  102401                 BVS     NEG30
004622  103404                 BCS     NEG31

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 764            <====

004624         NEG30:
004624  012742  000117         MOV     #117,-(R2)      ;MOVE TO MAILBOX #  *******  117  *******
004630  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004632  000000                 HALT                    ;NEG DID NOT SET CC'S CORRECTLY
004634  005214         NEG31:  INC     (R4)            ;LOC. 0=0
004636  001404                 BEQ     NEG32

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 756            <====

004640  012742  000120         MOV     #120,-(R2)      ;MOVE TO MAILBOX #  *******  120  *******
004644  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004646  000000                 HALT                    ;DATA RESULT OF NEG INCORRECT
004650  105137  000001 NEG32:  COMB    @#1             ;LOC 0=177400
004654  005237  000000         INC     @#0             ;LOC. 0=177401
004660  105430                 NEGB    @(R0)+          ;TRY NEGB LOC. 0=177777  R0=404
004662  100404                 BMI     NEG33

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 744            <====

004664  012742  000121         MOV     #121,-(R2)      ;MOVE TO MAILBOX #  *******  121  *******
004670  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004672  000000                 HALT                    ;NEGB FAILED WITH EVEN BYTE
004674  105430         NEG33:  NEGB    @(R0)+          ;TRY NEGB  LOC.0=777  R0=406
004676  100004                 BPL     NEG34

                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                       ;           CONDITIONAL BRANCH INST. AND    <====
                                                       ;           REPLACE THE MOVE INSTRUCTION    <====
                                                       ;           WHICH FOLLOWS W/ 736            <====

004700  012742  000122         MOV     #122,-(R2)      ;MOVE TO MAILBOX #  *******  122  *******
004704  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
004706  000000                 HALT                    ;NEGB FAILED WITH ODD BYTE
004710  105137  000001 NEG34:  COMB    @#1             ;LOC. 0=177377
```

```
1860  004714  105237  000001          INCB    @#1             ;LOC. 0=177777
1861  004720  005214                  INC     (R4)            ;LOC. 0=0
1862  004722  001404                  BEQ     TST63
1863                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1864                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1865                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1866                                                          ;           WHICH FOLLOWS W/ 724           <====
1867  004724  012742  000123          MOV     #123,-(R2)      ;MOVE TO MAILBOX #  *******  123  *******
1868  004730  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1869  004732  000000                  HALT                    ;DATA RESULT OF NEGB'S INCORRECT
1870                                                          ;  OR SEQUENCE ERROR
```



```
1871                                ;*****************************************************************************
1872                                ;
1873                                ;       THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
1874                                ;R0 IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
1875                                ;LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
1876                                ;COMPLEMENTS LOC.376.
1877                                ;       TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
1878                                ;TO COMPLETE THE TEST.
1879                                ;*****************************************************************************
1880                                ;TEST 63        TEST MODE 4 USING SOP INSTS
1881                                ;*****************************************************************************
1882
1883  004734  005212          TST63:  INC     (R2)            ;UPDATE TEST NUMBER
1884  004736  022712  000063          CMP     #63,(R2)        ;SEQUENCE ERROR?
1885  004742  001021                  BNE     TST64-10        ;BR TO ERROR HALT ON SEQ ERROR
1886  004744  005000                  CLR     R0              ;SET R0=400
1887  004746  105100                  COMB    R0
1888  004750  005200                  INC     R0
1889  004752  005040                  CLR     -(R0)           ;TRY TO CLEAR USING MODE 4
1890  004754  001404                  BEQ     SOP4A
1891                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1892                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1893                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1894                                                          ;           WHICH FOLLOWS W/ 773           <====
1895  004756  012742  000124          MOV     #124,-(R2)      ;MOVE TO MAILBOX #  *******  124  *******
1896  004762  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1897  004764  000000                  HALT                    ;CLR DID NOT SET Z-BIT
1898  004766  005200          SOP4A:  INC     R0              ;RESET R0
1899  004770  005200                  INC     R0
1900  004772  005140                  COM     -(R0)           ;TRY TO COMPLEMENT USING MODE 4
1901  004774  100004                  BPL     SOP4B
1902  004776  005200                  INC     R0              ;MOVE POINTER
1903  005000  005200                  INC     R0
1904  005002  005240                  INC     -(R0)
1905  005004  001404                  BEQ     TST64
1906                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
1907                                                          ;           CONDITIONAL BRANCH INST. AND   <====
1908                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
1909                                                          ;           WHICH FOLLOWS W/ 757           <====
1910  005006          SOP4B:
1911  005006  012742  000125          MOV     #125,-(R2)      ;MOVE TO MAILBOX #  *******  125  *******
1912  005012  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
1913  005014  000000                  HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
1914                                                          ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS.  IT
                                ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 372
                                ;THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
                                ;INSTRUCTIONS UNDER TEST.
                                ;       R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
                                ;AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
                                ;LOC. 0.  THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
                                ;INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
                                ;THE TEST.  THE PROPER DECREMENTING OF THE REGISTER IS ALSO
                                ;VERIFIED IN THIS MANNER.
                                ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
                                ;(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
                                ;
                                ;*****************************************************************************
                                ;TEST 64         TEST MODE 5 USING SOP INSTS
                                ;*****************************************************************************
005016  005212          TST64:  INC     (R2)            ;UPDATE TEST NUMBER
005020  022712  000064          CMP     #64,(R2)        ;SEQUENCE ERROR?
005024  001017                  BNE     TST65-10        ;BR TO ERROR HALT ON SEQ ERROR
005026  005000                  CLR     R0              ;SET R0=376
005030  005020                  CLR     (R0)+
005032  105400                  NEGB    R0
005034  005050                  CLR     @-(R0)          ;TRY TO CLEAR LOC 0 W/MODE 5
005036  001404                  BEQ     SOP5A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 773           <====
005040  012742  000126          MOV     #126,-(R2)      ;MOVE TO MAILBOX #  *******  126  *******
005044  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005046  000000                  HALT                    ;CLR DID NOT SET Z-BIT
005050  005200          SOP5A:  INC     R0              ;RESET R0
005052  005200                  INC     R0
005054  005150                  COM     @-(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
005056  100002                  BPL     SOP5B
005060  005250                  INC     @-(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 5
005062  001404                  BEQ     TST65
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
005064          SOP5B:
005064  012742  000127          MOV     #127,-(R2)      ;MOVE TO MAILBOX #  *******  127  *******
005070  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005072  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
                                                        ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 SINGLE OPERAND INSTRUCTIONS.  IT
                                ;USES LOCATION 0 AS ITS TARGET DATA.  R0 IS SET TO 400 USING
                                ;PREVIOUSLY TESTED INSTRUCTIONS AND A MODE 6 CLR INSTRUCTION IS
                                ;EXECUTED ON LOC. 0 USING R0 AND A -400 OFFSET.  COM AND INC
                                ;INSTRUCTIONS ARE THEN USED TO VERIFY THE DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 65         TEST MODE 6 USING SOP INSTS
                                ;*****************************************************************************
005074  005212          TST65:  INC     (R2)            ;UPDATE TEST NUMBER
005076  022712  000065          CMP     #65,(R2)        ;SEQUENCE ERROR?
005102  001020                  BNE     TST66-10        ;BR TO ERROR HALT ON SEQ ERROR
005104  005000                  CLR     R0              ;SET R0=400
005106  105100                  COMB    R0
005110  005200                  INC     R0
005112  005060  177400          CLR     -400(R0)        ;TRY TO CLEAR LOCATION 0 W/MODE 6
005116  001404                  BEQ     SOP6A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
005120  012742  000130          MOV     #130,-(R2)      ;MOVE TO MAILBOX #  *******  130  *******
005124  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005126  000000                  HALT                    ;CLR DID NOT SET Z-BIT
005130  005160  177400  SOP6A:  COM     -400(R0)        ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
005134  100003                  BPL     SOP6B
005136  005260  177400          INC     -400(R0)        ;TRY TO INCREMENT LOCATION 0 W/MODE 6
005142  001404                  BEQ     TST66
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
005144          SOP6B:
005144  012742  000131          MOV     #131,-(R2)      ;MOVE TO MAILBOX #  *******  131  *******
005150  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005152  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.
                        ;*****************************************************************************
005154  005212          TST66:  INC     (R2)            ;UPDATE TEST NUMBER
005156  022712  000066          CMP     #66,(R2)        ;SEQUENCE ERROR?
005162  001021                  BNE     TST67-10        ;BR TO ERROR HALT ON SEQ ERROR
005164  005000                  CLR     R0              ;SET R0=400
005166  105100                  COMB    R0
005170  005200                  INC     R0
005172  005210                  INC     (R0)            ;R0=1
005174  005070  000002          CLR     @2(R0)          ;TRY TO CLEAR LOC. 0 W/MODE 7
005200  001404                  BEQ     SOP7A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 771            <====
005202  012742  000132          MOV     #132,-(R2)      ;MOVE TO MAILBOX #  *******  132  *******
005206  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005210  000000                  HALT                    ;CLR DID NOT SET Z-BIT
005212  005170  000002  SOP7A:  COM     @2(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
005216  100003                  BPL     SOP7B
005220  005270  000002          INC     @2(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 7
005224  001404                  BEQ     TST67
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 757            <====
005226          SOP7B:
005226  012742  000133          MOV     #133,-(R2)      ;MOVE TO MAILBOX #  *******  133  *******
005232  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005234  000000                  HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;TEST 67         TEST MODE 4 WITH NEGATE INSTRUCTION
                        ;*****************************************************************************
005236  005212          TST67:  INC     (R2)            ;UPDATE TEST NUMBER
005240  022712  000067          CMP     #67,(R2)        ;SEQUENCE ERROR?
005244  001024                  BNE     TST70-10        ;BR TO ERROR HALT ON SEQ ERROR
005246  005000                  CLR     R0
005250  005010                  CLR     (R0)
005252  005120                  COM     (R0)+           ;LOC. 0=177777,  R0=2
005254  005440                  NEG     -(R0)           ;TRY NEGATE, LOC. 0=1
005256  100403                  BMI     NEG40           ;CC=0001?
005260  001402                  BEQ     NEG40
005262  102401                  BVS     NEG40
005264  103404                  BCS     NEG41
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 770            <====
005266          NEG40:
005266  012742  000134          MOV     #134,-(R2)      ;MOVE TO MAILBOX #  *******  134  *******
005272  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005274  000000                  HALT
005276  005400          NEG41:  NEG     R0              ;NEG DID NOT SET CC'S CORRECTLY
005300  001404                  BEQ     NEG42           ;TST R0 WITH A NEG.
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 762            <====
005302  012742  000135          MOV     #135,-(R2)      ;MOVE TO MAILBOX #  *******  135  *******
005306  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005310  000000                  HALT
005312  005310          NEG42:  DEC     (R0)            ;R0 NOT DECREMENTED PROPERLY
005314  001404                  BEQ     TST70           ;TEST DTA RESULT OF NEG
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 754            <====
005316  012742  000136          MOV     #136,-(R2)      ;MOVE TO MAILBOX #  *******  136  *******
005322  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005324  000000                  HALT                    ;DATA RESULT OF NEG INCORRECT
                                                        ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;TEST 70          TEST MODE 5 WITH NEGATE INSTRUCTION
                        ;*****************************************************************************
005326  005212          TST70:  INC     (R2)            ;UPDATE TEST NUMBER
005330  022712  000070          CMP     #70,(R2)        ;SEQUENCE ERROR?
005334  001031                  BNE     TST71-10        ;BR TO ERROR HALT ON SEQ ERROR
005336  005000                  CLR     R0              ;R0=0
005340  005010                  CLR     (R0)            ;LOC. 0=0
005342  105100                  COMB    R0              ;R0=377
005344  005200                  INC     R0              ;R0=400
005346  005010                  CLR     (R0)            ;SET 400 = 0
005350  005004                  CLR     R4              ;R4=0
005352  005314                  DEC     (R4)            ;LOC. 0=177777
005354  005450                  NEG     @-(R0)          ;TRY NEGATE:  LOC. 0=1
005356  100403                  BMI     NEG50           ;CC=0001?
005360  001402                  BEQ     NEG50
005362  102401                  BVS     NEG50
005364  103404                  BCS     NEG51

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
005366                  NEG50:
005366  012742  000137          MOV     #137,-(R2)      ;MOVE TO MAILBOX #  *******  137  *******
005372  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005374  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005376  005314          NEG51:  DEC     (R4)
005400  001404                  BEQ     NEG52

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
005402  012742  000140          MOV     #140,-(R2)      ;MOVE TO MAILBOX #  *******  140  *******
005406  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005410  000000                  HALT                    ;DATA RESULT OF NEG INCORRECT
005412  105100          NEG52:  COMB    R0
005414  005300                  DEC     R0
005416  001404                  BEQ     TST71

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 747           <====
005420  012742  000141          MOV     #141,-(R2)      ;MOVE TO MAILBOX #  *******  141  *******
005424  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005426  000000                  HALT                    ;REGISTER NOT DECREMENTED PROPERLY
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;TEST 71          TEST MODE 6 WITH NEGATE
                        ;*****************************************************************************
005430  005212          TST71:  INC     (R2)            ;UPDATE TEST NUMBER
005432  022712  000071          CMP     #71,(R2)        ;SEQUENCE ERROR?
005436  001022                  BNE     TST72-10        ;BR TO ERROR HALT ON SEQ ERROR
005440  005000                  CLR     R0              ;R0=0
005442  005004                  CLR     R4              ;R4=0
005444  105100                  COMB    R0              ;R0=377
005446  005014                  CLR     (R4)            ;LOC. 0=0
005450  105024                  CLRB    (R4)+           ;LOC. 0=177777,  R4=1
005452  105114                  COMB    (R4)            ;LOC. 0=177400
005454  005460  177401          NEG     -377(R0)        ;LOC. 0=400
005460  100403                  BMI     NEG60           ;CC=0001
005462  001402                  BEQ     NEG60
005464  102401                  BVS     NEG60
005466  103404                  BCS     NEG61

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 764           <====
005470                  NEG60:
005470  012742  000142          MOV     #142,-(R2)      ;MOVE TO MAILBOX #  *******  142  *******
005474  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005476  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005500  105314          NEG61:  DECB    (R4)
005502  001404                  BEQ     TST72

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
005504  012742  000143          MOV     #143,-(R2)      ;MOVE TO MAILBOX #  *******  143  *******
005510  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005512  000000                  HALT                    ;DATA RESULT OF NEG INCORRECT
                                                        ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;TEST 72          TEST MODE 7 W/ NEGATE
                        ;*****************************************************************************
005514  005212          TST72:  INC     (R2)            ;UPDATE TEST NUMBER
005516  022712  000072          CMP     #72,(R2)        ;SEQUENCE ERROR?
005522  001024                  BNE     TST73-10        ;BR TO ERROR HALT ON SEQ ERROR
005524  005000                  CLR     R0              ;R0=0
005526  005010                  CLR     (R0)            ;LOC. 0=0
005530  005110                  COM     (R0)            ;LOC. 0=177777
005532  105100                  COMB    R0              ;R0=377
005534  105470  000005          NEGB    @5(R0)          ;R0+5=404,  404=1,  LOC. 0=777
005540  100403                  BMI     NEG70           ;CC=0001?
005542  001402                  BEQ     NEG70
005544  102401                  BVS     NEG70
005546  103404                  BCS     NEG71
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
005550                  NEG70:
005550  012742  000144          MOV     #144,-(R2)      ;MOVE TO MAILBOX #  *******  144  *******
005554  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005556  000000                  HALT                    ;NEG DID NOT SET CC'S CORRECTLY
005560  105100          NEG71:  COMB    R0              ;R0=0
005562  105120                  COMB    (R0)+           ;LOC. 0=400,  R0=1
005564  105310                  DECB    (R0)            ;LOC. 0=0
005566  005467  172206          NEG     0               ;USE NEG MODE 67 TO TST FOR ZERO
005572  001404                  BEQ     TST73
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
005574  012742  000145          MOV     #145,-(R2)      ;MOVE TO MAILBOX #  *******  145  *******
005600  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005602  000000                  HALT                    ;DATA RESULT OF NEG WAS INCORRECT
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
                        ;INSTRUCTIONS.  CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
                        ;INSTRUCTION (SOPX).  THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
                        ;77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
                        ;OF THESE INSTRUCTIONS.
                        ;
                        ;*****************************************************************************
                        ;TEST 73          TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
                        ;*****************************************************************************
005604  005212          TST73:  INC     (R2)            ;UPDATE TEST NUMBER
005606  022712  000073          CMP     #73,(R2)        ;SEQUENCE ERROR?
005612  001017                  BNE     SOPB            ;BR TO ERROR HALT ON SEQ ERROR
005614  005027                  CLR     (R7)+           ;CLEAR NEXT LOCATION: (SOPX)
005616  177777          SOPX:   -1                      ;USE MODE 27
005620  001404                  BEQ     SOPA
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 775           <====
005622  012742  000146          MOV     #146,-(R2)      ;MOVE TO MAILBOX #  *******  146  *******
005626  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005630  000000                  HALT                    ;CLR DID NOT SET Z-BIT
005632  005237  005616          SOPA:   INC     @#SOPX          ;INC SOPX W/MODE 37
005636  005467  177754          NEG     SOPX            ;NEGATE SOPX W/MODE 67
005642  100003                  BPL     SOPB
005644  005277  000012          INC     @SOPXAD         ;INC SOPX W/MODE 77
005650  001405                  BEQ     TST74
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
005652                  SOPB:
005652  012742  000147          MOV     #147,-(R2)      ;MOVE TO MAILBOX #  *******  147  *******
005656  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
005660  000000                  HALT                    ;INC DID NOT SET Z-BIT
                                                        ;  OR SEQUENCE ERROR
005662  005616          SOPXAD: SOPX                    ;INDIRECT ADDRESS OF SOPX
```

```
2245
2246                                    ;*****************************************************************************
2247                                    ;
2248                                    ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
2249                                    ;USING MODE 0.  R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
2250                                    ;TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION.  A TST INSTRUCTION
2251                                    ;IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION
2252                                    ;CODES.
2253                                    ;
2254                                    ;*****************************************************************************
2255                                    ;TEST 74        TEST MODE 0 SOP NON-MODIFYING
2256                                    ;*****************************************************************************
2257  005664  005212                    TST74:  INC     (R2)            ;UPDATE TEST NUMBER
2258  005666  022712  000074                    CMP     #74,(R2)        ;SEQUENCE ERROR?
2259  005672  001010                            BNE     TST75-10        ;BR TO ERROR HALT ON SEQ ERROR
2260  005674  005000                            CLR     R0              ;INITIALIZE R0=0
2261  005676  000277                            SCC                     ;SET CC=1011
2262  005700  000244                            CLZ
2263  005702  005700                            TST     R0              ;TRY TST W/ MODE 0
2264  005704  102403                            BVS     SNM0A           ;CHECK THAT CC=0100
2265  005706  100402                            BMI     SNM0A
2266  005710  103401                            BCS     SNM0A
2267  005712  001404                            BEQ     TST75
2268                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2269                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2270                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2271                                                                    ;           WHICH FOLLOWS W/ 770           <====
2272  005714                            SNM0A:
2273  005714  012742  000150                   MOV     #150,-(R2)      ;MOVE TO MAILBOX #  *******  150  *******
2274  005720  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2275  005722  000000                            HALT                    ;CONDITION CODES NOT SET PROPERLY
2276                                                                    ;  OR SEQUENCE ERROR
```



```
2277
2278                                    ;*****************************************************************************
2279                                    ;
2280                                    ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0.
2281                                    ;R0 IS SET TO 377 AND COMPLEMENT OF THE EXPECTED CONDITION CODES
2282                                    ;IS LOADED IN PSW.  A TSTB INSTRUCTION IS EXECUTED AND THE RESULTS
2283                                    ;ARE CHECKED WITH SEVERAL CONDITIONAL BRANCH INSTRUCTIONS.
2284                                    ;       THIS VERIFIES THAT THE PROPER BYTE WAS TESTED.
2285                                    ;
2286                                    ;*****************************************************************************
2287                                    ;TEST 75        TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING
2288                                    ;*****************************************************************************
2289  005724  005212                    TST75:  INC     (R2)            ;UPDATE TEST NUMBER
2290  005726  022712  000075                    CMP     #75,(R2)        ;SEQUENCE ERROR?
2291  005732  001010                            BNE     TST76-10        ;BR TO ERROR HALT ON SEQ ERROR
2292  005734  005000                            CLR     R0              ;INITIALIZE
2293  005736  105100                            COMB    R0              ;R0=377
2294  005740  000277                            SCC                     ;SET CC=0111
2295  005742  000250                            CLN
2296  005744  105700                            TSTB    R0              ;TRY TST EVEN BYTE
2297  005746  102402                            BVS     SNMB0A          ;CHECK CC=1000
2298  005750  101401                            BLOS    SNMB0A
2299  005752  100404                            BMI     TST76
2300                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2301                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
2302                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
2303                                                                    ;           WHICH FOLLOWS W/ 770           <====
2304  005754                            SNMB0A:
2305  005754  012742  000151                   MOV     #151,-(R2)      ;MOVE TO MAILBOX #  *******  151  *******
2306  005760  005242                            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
2307  005762  000000                            HALT                    ;CONDITION CODES NOT SET PROPERLY
2308                                                                    ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
                        ;R0 IS USED TO POINT TO AND CLEAR LOC. 0.  THE COMPLEMENT OF THE
                        ;EXPECTED CONDITION CODES ARE LOADED IN THE PSW.  A TST INSTRUCTION
                        ;IS THEN EXECUTED ON LOC. 0 USING R0 AND CONDITIONAL BRANCHES TEST
                        ;THE RESULTS.
                        ;
                        ;*****************************************************************************
                        ;TEST 76         TEST MODE 1 SOP NON-MODIFYING
                        ;*****************************************************************************
005764  005212          TST76:  INC     (R2)            ;UPDATE TEST NUMBER
005766  022712  000076          CMP     #76,(R2)        ;SEQUENCE ERROR?
005772  001011                  BNE     TST77-10        ;BR TO ERROR HALT ON SEQ ERROR
005774  005000                  CLR     R0              ;POINT TO LOC 0
005776  005010                  CLR     (R0)            ;CLEAR LOC 0
006000  000277                  SCC                     ;INITIALIZE
006002  000244                  CLZ                     ;CC=1011
006004  005710                  TST     (R0)            ;TRY TST W/ MODE 1
006006  102403                  BVS     SNM1A           ;CHECK CC=0100
006010  103402                  BCS     SNM1A
006012  100401                  BMI     SNM1A
006014  001404                  BEQ     TST77

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
006016                  SNM1A:
006016  012742  000152          MOV     #152,-(R2)      ;MOVE TO MAILBOX #  *******  152  *******
006022  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006024  000000                  HALT                    ;CC'S NOT SET PROPERLY
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
                        ;THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
                        ;AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
                        ;PROPER CONDITION CODE BITS.
                        ;
                        ;*****************************************************************************
                        ;TEST 77         TEST MODE 1 BYTE INST. NON-MODIFYING
                        ;*****************************************************************************
006026  005212          TST77:  INC     (R2)            ;UPDATE TEST NUMBER
006030  022712  000077          CMP     #77,(R2)        ;SEQUENCE ERROR?
006034  001026                  BNE     TST100-10       ;BR TO ERROR HALT ON SEQ ERROR
006036  005000                  CLR     R0              ;POINT TO LOC 0
006040  005010                  CLR     (R0)            ;CLEAR LOC 0
006042  105110                  COMB    (R0)            ;COMPLEMENT BYTE 0
006044  000277                  SCC                     ;SET CC=0111
006046  000250                  CLN
006050  105710                  TSTB    (R0)            ;TRY TST ON EVEN BYTE
006052  102402                  BVS     SNMB1A
006054  101401                  BLOS    SNMB1A
006056  100404                  BMI     SNMB1B

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
006060                  SNMB1A:
006060  012742  000153          MOV     #153,-(R2)      ;MOVE TO MAILBOX #  *******  153  *******
006064  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006066  000000                  HALT                    ;CC'S NOT CORRECT
006070  005000          SNMB1B: CLR     R0
006072  005200                  INC     R0
006074  000277                  SCC                     ;SET CC=1011
006076  000244                  CLZ
006100  105710                  TSTB    (R0)            ;TRY TO TST AN ODD BYTE
006102  102403                  BVS     SNMB1C          ;CHECK CC=0100
006104  103402                  BCS     SNMB1C
006106  100401                  BMI     SNMB1C
006110  001404                  BEQ     TST100

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
006112                  SNMB1C:
006112  012742  000154          MOV     #154,-(R2)      ;MOVE TO MAILBOX #  *******  154  *******
006116  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006120  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
                                ;*****************************************************************************
006122  005212                TST100: INC     (R2)            ;UPDATE TEST NUMBER
006124  022712  000100                CMP     #100,(R2)       ;SEQUENCE ERROR?
006130  001020                        BNE     TST101-10       ;BR TO ERROR HALT ON SEQ ERROR
006132  005000                        CLR     R0              ;INITIALIZE R0=0
006134  005010                        CLR     (R0)            ;CLEAR LOC 0
006136  000277                        SCC                     ;SET CC=1011
006140  000244                        CLZ
006142  005720                        TST     (R0)+           ;TRY TST W/ MODE 2
006144  102403                        BVS     SNM2A           ;CHECK CC=0100
006146  103402                        BCS     SNM2A
006150  100401                        BMI     SNM2A
006152  001404                        BEQ     SNM2B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 767            <====
006154                          SNM2A:
006154  012742  000155                MOV     #155,-(R2)      ;MOVE TO MAILBOX #  *******  155  *******
006160  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006162  000000                        HALT                    ;CC'S NOT CORRECT
006164  005300                SNM2B:  DEC     R0              ;RESET R0
006166  005300                        DEC     R0
006170  001404                        BEQ     TST101
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 760            <====
006172  012742  000156                MOV     #156,-(R2)      ;MOVE TO MAILBOX #  *******  156  *******
006176  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006200  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
                                ;*****************************************************************************
006202  005212                TST101: INC     (R2)            ;UPDATE TEST NUMBER
006204  022712  000101                CMP     #101,(R2)       ;SEQUENCE ERROR?
006210  001042                        BNE     TST102-10       ;BR TO ERROR HALT ON SEQ ERROR
006212  005000                        CLR     R0              ;CLEAR R0
006214  005010                        CLR     (R0)            ;CLEAR LOC 0
006216  105110                        COMB    (R0)            ;SET LOC 0=377
006220  000277                        SCC                     ;SET CC=0111
006222  000250                        CLN
006224  105720                        TSTB    (R0)+           ;TRY TST OF EVEN BYTE
006226  102402                        BVS     SNMB2A
006230  101401                        BLOS    SNMB2A
006232  100404                        BMI     SNMB2B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 767            <====
006234                          SNMB2A:
006234  012742  000157                MOV     #157,-(R2)      ;MOVE TO MAILBOX #  *******  157  *******
006240  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006242  000000                        HALT                    ;CC'S NOT SET CORRECTLY
006244  005300                SNMB2B: DEC     R0              ;DECREMENT R0
006246  001404                        BEQ     SNMB2C
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 761            <====
006250  012742  000160                MOV     #160,-(R2)      ;MOVE TO MAILBOX #  *******  160  *******
006254  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006256  000000                        HALT                    ;MODE 2 DID NOT INC REG CORRECTLY
006260  005200                SNMB2C: INC     R0              ;POINT TO ODD BYTE
006262  000277                        SCC                     ;SET CC=1011
006264  000244                        CLZ
006266  105720                        TSTB    (R0)+           ;TRY TST OF ODD BYTE
006270  102403                        BVS     SNMB2D          ;CHECK CC'S=0100
006272  103402                        BCS     SNMB2D
006274  100401                        BMI     SNMB2D
006276  001404                        BEQ     SNMB2E
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 745            <====
006300                          SNMB2D:
006300  012742  000161                MOV     #161,-(R2)      ;MOVE TO MAILBOX #  *******  161  *******
006304  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
  2489  006306  000000                    HALT                    ;CC'S NOT CORRECT
  2490  006310  005300            SNMB2E: DEC     R0
  2491  006312  005300                    DEC     R0
  2492  006314  001404                    BEQ     TST102
  2493                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
  2494                                                            ;           CONDITIONAL BRANCH INST. AND   <====
  2495                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
  2496                                                            ;           WHICH FOLLOWS W/ 736           <====
  2497  006316  012742  000162          MOV     #162,-(R2)      ;MOVE TO MAILBOX #  *******  162  *******
  2498  006322  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
  2499  006324  000000                    HALT                    ;R0 DID NOT INCREMENT PROPERLY
  2500                                                            ;  OR SEQUENCE ERROR
```



```
  2501
  2502                                  ;*****************************************************************************
  2503                                  ;
  2504                                  ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
  2505                                  ;A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
  2506                                  ;THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
  2507                                  ;TST MODE 3 INSTRUCTION.
  2508                                  ;
  2509                                  ;*****************************************************************************
  2510                                  ;TEST 102       TEST MODE 3 W/ SOP NON-MODIFYING INSTS
  2511                                  ;*****************************************************************************
  2512  006326  005212            TST102: INC     (R2)            ;UPDATE TEST NUMBER
  2513  006330  022712  000102            CMP     #102,(R2)       ;SEQUENCE ERROR?
  2514  006334  001022                    BNE     TST103-10       ;BR TO ERROR HALT ON SEQ ERROR
  2515  006336  005000                    CLR     R0              ;R0=0
  2516  006340  005010                    CLR     (R0)            ;CLEAR LOC 0
  2517  006342  105100                    COMB    R0              ;R0=376
  2518  006344  005300                    DEC     R0
  2519  006346  000277                    SCC                     ;SET CC=1011
  2520  006350  000244                    CLZ
  2521  006352  005730                    TST     @(R0)+          ;TRY TST W/ MODE 3
  2522  006354  102403                    BVS     SNM3A           ;CHECK CC=0100
  2523  006356  103402                    BCS     SNM3A
  2524  006360  100401                    BMI     SNM3A
  2525  006362  001404                    BEQ     SNM3B
  2526                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
  2527                                                            ;           CONDITIONAL BRANCH INST. AND   <====
  2528                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
  2529                                                            ;           WHICH FOLLOWS W/ 765           <====
  2530  006364                    SNM3A:
  2531  006364  012742  000163          MOV     #163,-(R2)      ;MOVE TO MAILBOX #  *******  163  *******
  2532  006370  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
  2533  006372  000000                    HALT                    ;CC'S NOT CORRECT
  2534  006374  005300            SNM3B:  DEC     R0              ;R0=377
  2535  006376  105100                    COMB    R0              ;R0=0
  2536  006400  001404                    BEQ     TST103
  2537                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
  2538                                                            ;           CONDITIONAL BRANCH INST. AND   <====
  2539                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
  2540                                                            ;           WHICH FOLLOWS W/ 756           <====
  2541  006402  012742  000164          MOV     #164,-(R2)      ;MOVE TO MAILBOX #  *******  164  *******
  2542  006406  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
  2543  006410  000000                    HALT                    ;MODE 3 DID NOT INC REG CORRECTLY
  2544                                                            ;  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.
                                ;*****************************************************************************
006412  005212                TST103: INC     (R2)            ;UPDATE TEST NUMBER
006414  022712  000103                CMP     #103,(R2)       ;SEQUENCE ERROR?
006420  001036                        BNE     TST104-10       ;BR TO ERROR HALT ON SEQ ERROR
006422  005000                        CLR     R0              ;R0=0
006424  005010                        CLR     (R0)            ;CLEAR LOC 0
006426  105110                        COMB    (R0)            ;LOC. 0 =377
006430  105100                        COMB    R0
006432  005200                        INC     R0
006434  005720                        TST     (R0)+           ;R0=402
006436  000277                        SCC                     ;CC=0111
006440  000250                        CLN
006442  105730                        TSTB    @(R0)+          ;TRY TST OF EVEN BYTE
006444  102402                        BVS     SNMB3A          ;CHECK CC=1000
006446  101401                        BLOS    SNMB3A
006450  100404                        BMI     SNMB3B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 764           <====
006452                  SNMB3A:
006452  012742  000165                MOV     #165,-(R2)      ;MOVE TO MAILBOX #  *******  165  *******
006456  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006460  000000                        HALT                    ;CC'S NOT CORRECT
006462  000277          SNMB3B:       SCC                     ;SET CC=1011
006464  000244                        CLZ
006466  105730                        TSTB    @(R0)+          ;TRY TST OF ODD BYTE
006470  102403                        BVS     SNMB3C          ;CHECK CC=0100
006472  103402                        BCS     SNMB3C
006474  100401                        BMI     SNMB3C
006476  001404                        BEQ     SNMB3D
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 751           <====
006500                  SNMB3C:
006500  012742  000166                MOV     #166,-(R2)      ;MOVE TO MAILBOX #  *******  166  *******
006504  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006506  000000                        HALT                    ;CC'S NOT CORRECT
006510  005720          SNMB3D:       TST     (R0)+           ;R0=410
006512  005710                        TST     (R0)
006514  100404                        BMI     TST104
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
```



```
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 742           <====
006516  012742  000167                MOV     #167,-(R2)      ;MOVE TO MAILBOX #  *******  167  *******
006522  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006524  000000                        HALT                    ;TSTB DID NOT INCREMENT R0 CORRETCLY
                                                            ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
                                ;LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
                                ;EXPECTED RESULTS.  R0 AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
                                ;THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
                                ;IS CHECKED FOR PROPER DECREMENTING.
                                ;
                                ;*****************************************************************************
                                ;TEST 104       TEST MODE 4 W/ SOP NON-MODIFYING INSTS
                                ;*****************************************************************************
006526  005212                TST104: INC     (R2)            ;UPDATE TEST NUMBER
006530  022712  000104                CMP     #104,(R2)       ;SEQUENCE ERROR?
006534  001017                        BNE     TST105-10       ;BR TO ERROR HALT ON SEQ ERROR
006536  005000                        CLR     R0              ;R0=0
006540  005010                        CLR     (R0)            ;LOC 0=0
006542  005120                        COM     (R0)+           ;LOC 0=-1
006544  000277                        SCC                     ;SET CC=1011
006546  000244                        CLZ
006550  005740                        TST     -(R0)           ;TRY TST W/ MODE 4
006552  102402                        BVS     SNM4A           ;CHECK CC=0100
006554  101401                        BLOS    SNM4A
006556  100404                        BMI     SNM4B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 767           <====
006560                  SNM4A:
006560  012742  000170                MOV     #170,-(R2)      ;MOVE TO MAILBOX #  *******  170  *******
006564  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006566  000000                        HALT                    ;CC'S NOT CORRECT
006570  005700          SNM4B:        TST     R0
006572  001404                        BEQ     TST105
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                            ;           CONDITIONAL BRANCH INST. AND   <====
                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
                                                            ;           WHICH FOLLOWS W/ 761           <====
006574  012742  000171                MOV     #171,-(R2)      ;MOVE TO MAILBOX #  *******  171  *******
006600  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006602  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
                        ;*****************************************************************************
006604  005212         TST105: INC     (R2)            ;UPDATE TEST NUMBER
006606  022712  000105         CMP     #105,(R2)       ;SEQUENCE ERROR?
006612  001022                 BNE     TST106-10       ;BR TO ERROR HALT ON SEQ ERROR
006614  005000                 CLR     R0              ;R0=0
006616  005010                 CLR     (R0)            ;LOC 0=0
006620  005110                 COM     (R0)            ;LOC 0=-1
006622  105100                 COMB    R0              ;R0=377
006624  005200                 INC     R0              ;R0=400
006626  000277                 SCC                     ;SET CC=0111
006630  000250                 CLN
006632  005750                 TST     @-(R0)          ;TRY TST W/ MODE 5
006634  102402                 BVS     SNM5A           ;CHECK CC=1000
006636  101401                 BLOS    SNM5A
006640  100404                 BMI     SNM5B
                        ;
                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                        ;           CONDITIONAL BRANCH INST. AND    <====
                        ;           REPLACE THE MOVE INSTRUCTION    <====
                        ;           WHICH FOLLOWS W/ 765            <====
006642                  SNM5A:
006642  012742  000172         MOV     #172,-(R2)      ;MOVE TO MAILBOX #  *******  172  *******
006646  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006650  000000                 HALT                    ;CC'S NOT SET PROPERLY
006652  005200                 INC     R0              ;R0=377
006654  105100                 COMB    R0              ;R0=0
006656  001404                 BEQ     TST106
                        ;
                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                        ;           CONDITIONAL BRANCH INST. AND    <====
                        ;           REPLACE THE MOVE INSTRUCTION    <====
                        ;           WHICH FOLLOWS W/ 756            <====
006660  012742  000173         MOV     #173,-(R2)      ;MOVE TO MAILBOX #  *******  173  *******
006664  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006666  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
                        ;*****************************************************************************
006670  005212         TST106: INC     (R2)            ;UPDATE TEST NUMBER
006672  022712  000106         CMP     #106,(R2)       ;SEQUENCE ERROR?
006676  001021                 BNE     TST107-10       ;BR TO ERROR HALT ON SEQ ERROR
006700  005000                 CLR     R0              ;R0=0
006702  005010                 CLR     (R0)            ;LOC 0=0
006704  005110                 COM     (R0)            ;LOC 0=-1
006706  105100                 COMB    R0              ;R0=377
006710  000277                 SCC                     ;SET CC=0111
006712  000250                 CLN
006714  005760  177401         TST     -377(R0)        ;TRY TST W/ MODE 6
006720  102402                 BVS     SNM6A           ;CHECK CC=1000
006722  101401                 BLOS    SNM6A
006724  100404                 BMI     SNM6B
                        ;
                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                        ;           CONDITIONAL BRANCH INST. AND    <====
                        ;           REPLACE THE MOVE INSTRUCTION    <====
                        ;           WHICH FOLLOWS W/ 765            <====
006726                  SNM6A:
006726  012742  000174         MOV     #174,-(R2)      ;MOVE TO MAILBOX #  *******  174  *******
006732  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006734  000000                 HALT                    ;CC'S INCORRECT
006736  105100                 COMB    R0              ;R0=0
006740  001404                 BEQ     TST107
                        ;
                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                        ;           CONDITIONAL BRANCH INST. AND    <====
                        ;           REPLACE THE MOVE INSTRUCTION    <====
                        ;           WHICH FOLLOWS W/ 757            <====
006742  012742  000175         MOV     #175,-(R2)      ;MOVE TO MAILBOX #  *******  175  *******
006746  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
006750  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.
                                ;*****************************************************************************
006752  005212                  TST107: INC     (R2)            ;UPDATE TEST NUMBER
006754  022712  000107                  CMP     #107,(R2)       ;SEQUENCE ERROR?
006760  001021                          BNE     TST110-10       ;BR TO ERROR HALT ON SEQ ERROR
006762  005000                          CLR     R0              ;R0=0
006764  005010                          CLR     (R0)            ;LOC 0=0
006766  005110                          COM     (R0)            ;LOC 0=-1
006770  105100                          COMB    R0              ;R0=377
006772  000277                          SCC                     ;CC=0111
006774  000250                          CLN
006776  005770  000001                  TST     @1(R0)          ;TRY TST W/ MODE 7
007002  102402                          BVS     SNM7A           ;CHECK CC=1000
007004  101401                          BLOS    SNM7A
007006  100404                          BMI     SNM7B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 765           <====
007010                          SNM7A:
007010  012742  000176                  MOV     #176,-(R2)      ;MOVE TO MAILBOX #  *******  176  *******
007014  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007016  000000                          HALT                    ;CC'S NOT CORRECT
007020  105100                  SNM7B:  COMB    R0              ;R0=0
007022  001404                          BEQ     TST110
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
007024  012742  000177                  MOV     #177,-(R2)      ;MOVE TO MAILBOX #  *******  177  *******
007030  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007032  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.
                                ;*****************************************************************************
007034  005212                  TST110: INC     (R2)            ;UPDATE TEST NUMBER
007036  022712  000110                  CMP     #110,(R2)       ;SEQUENCE ERROR?
007042  001006                          BNE     TST111-10       ;BR TO ERROR HALT ON SEQ ERROR
007044  005000                          CLR     R0              ;R0=0
007046  005100                          COM     R0              ;R0=-1
007050  005004                          CLR     R4              ;R4=0
007052  060004                          ADD     R0,R4           ;TRY ADD: R4=-1
007054  005204                          INC     R4              ;R4=0
007056  001404                          BEQ     TST111
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 772           <====
007060  012742  000200                  MOV     #200,-(R2)      ;MOVE TO MAILBOX #  *******  200  *******
007064  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007066  000000                          HALT                    ;ADD INST. FAILED W/ MODE 0
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.
                                ;THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES UNIQUE
                                ;MICROCODE.
                                ;
                                ;*****************************************************************************
                                ;TEST 111        MOV MODE 0 TO MODE 0
                                ;*****************************************************************************
007070  005212                  TST111: INC     (R2)            ;UPDATE TEST NUMBER
007072  022712  000111                  CMP     #111,(R2)       ;SEQUENCE ERROR?
007076  001006                          BNE     TST112-10       ;BR TO ERROR HALT ON SEQ ERROR
007100  005000                          CLR     R0              ;R0=0
007102  005004                          CLR     R4              ;R4=0
007104  005100                          COM     R0              ;R0=-1
007106  010004                          MOV     R0,R4           ;TRY MOVE -1 TO R4
007110  005204                          INC     R4              ;INC R4
007112  001404                          BEQ     TST112
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 772           <====
007114  012742  000201                  MOV     #201,-(R2)      ;MOVE TO MAILBOX #  *******  201  *******
007120  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007122  000000                          HALT                    ;MOVE FAILED MODE 0 TO MODE 0
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
```

```
                                ;
                                ;       THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
                                ;THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES SOME
                                ;UNIQUE MICROCODE.
                                ;
                                ;*****************************************************************************
                                ;TEST 112        TEST SUB MODE 0,0
                                ;*****************************************************************************
007124  005212                  TST112: INC     (R2)            ;UPDATE TEST NUMBER
007126  022712  000112                  CMP     #112,(R2)       ;SEQUENCE ERROR?
007132  001016                          BNE     TST113-10       ;BR TO ERROR HALT ON SEQ ERROR
007134  005000                          CLR     R0              ;R0=0
007136  005004                          CLR     R4              ;R4=0
007140  005204                          INC     R4              ;R4=1
007142  160400                          SUB     R4,R0           ;TRY SUB 0,0   R0=-1
007144  100003                          BPL     SUB0            ;CC=1001
007146  001402                          BEQ     SUB0
007150  102401                          BVS     SUB0
007152  103404                          BCS     SUB0A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
007154                          SUB0:
007154  012742  000202                  MOV     #202,-(R2)      ;MOVE TO MAILBOX #  *******  202  *******
007160  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007162  000000                          HALT                    ;CONDITION CODE FAILED ON SUB
007164  005200                  SUB0A:  INC     R0
007166  001404                          BEQ     TST113
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 762           <====
007170  012742  000203                  MOV     #203,-(R2)      ;MOVE TO MAILBOX #  *******  203  *******
007174  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007176  000000                          HALT                    ;DATA RESULT OF SUB FAILED
                                                                ;  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
                                ;*****************************************************************************
007200  005212                  TST113: INC     (R2)            ;UPDATE TEST NUMBER
007202  022712  000113                  CMP     #113,(R2)       ;SEQUENCE ERROR?
007206  001051                          BNE     TST114-10       ;BR TO ERROR HALT ON SEQ ERROR
007210  005000                          CLR     R0              ;R0=0
007212  010004                          MOV     R0,R4           ;TRY MOVE MODE 0,0
007214  001404                          BEQ     DOP0A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 775           <====
007216  012742  000204                  MOV     #204,-(R2)      ;MOVE TO MAILBOX #  *******  204  *******
007222  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007224  000000                          HALT                    ;Z-BIT NOT SET
007226  005200                  DOP0A:  INC     R0              ;R0=1
007230  005100                          COM     R0              ;R0=177776
007232  005104                          COM     R4              ;R4=177777
007234  040004                          BIC     R0,R4           ;TRY BIC: R4=1
007236  005304                          DEC     R4              ;R4=0
007240  001404                          BEQ     DOP0B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 763           <====
007242  012742  000205                  MOV     #205,-(R2)      ;MOVE TO MAILBOX #  *******  205  *******
007246  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007250  000000                          HALT                    ;BIC CLEAR RESULT INCORRECT
007252  050004                  DOP0B:  BIS     R0,R4           ;TRY BIS: R4=177777
007254  005204                          INC     R4
007256  005204                          INC     R4              ;R4=0
007260  001404                          BEQ     DOP0C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 753           <====
007262  012742  000206                  MOV     #206,-(R2)      ;MOVE TO MAILBOX #  *******  206  *******
007266  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007270  000000                          HALT                    ;RESULT OF BIS INCORRECT
007272  005000                  DOP0C:  CLR     R0              ;R0=0
007274  105100                          COMB    R0              ;R0=377
007276  005004                          CLR     R4              ;R4=0
007300  005104                          COM     R4              ;R4=177777
007302  040004                          BIC     R0,R4           ;R4=177400
007304  060004                          ADD     R0,R4           ;TRY ADD: R4=177777
007306  005204                          INC     R4              ;R4=0
```

```
2925  007310  001404                    BEQ     DOP0D
2926                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2927                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2928                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2929                                                  ;           WHICH FOLLOWS W/ 737           <====
2930  007312  012742  000207                    MOV     #207,-(R2)  ;MOVE TO MAILBOX #  *******  207  *******
2931  007316  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2932  007320  000000                            HALT                ;RESULT OF ADD INCORRECT
2933  007322  160004                    DOP0D:  SUB     R0,R4       ;177401=R4
2934  007324  105404                            NEGB    R4          ;R4=177777
2935  007326  005204                            INC     R4          ;R0=0
2936  007330  001404                            BEQ     TST114
2937                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2938                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2939                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2940                                                  ;           WHICH FOLLOWS W/ 727           <====
2941  007332  012742  000210                    MOV     #210,-(R2)  ;MOVE TO MAILBOX #  *******  210  *******
2942  007336  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2943  007340  000000                            HALT                ;RESULT OF SUB INCORRECT
2944                                                  ;  OR SEQUENCE ERROR
2945
```



```
2946                                        ;*****************************************************************************
2947                                        ;
2948                                        ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
2949                                        ;DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
2950                                        ;
2951                                        ;*****************************************************************************
2952                                        ;TEST 114       TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
2953                                        ;*****************************************************************************
2954  007342  005212                    TST114: INC     (R2)        ;UPDATE TEST NUMBER
2955  007344  022712  000114                    CMP     #114,(R2)   ;SEQUENCE ERROR?
2956  007350  001024                            BNE     TST115-10   ;BR TO ERROR HALT ON SEQ ERROR
2957  007352  005000                            CLR     R0          ;R0=0
2958  007354  005010                            CLR     (R0)        ;LOC. 0=0
2959  007356  105110                            COMB    (R0)        ;LOC. 0=377
2960  007360  005220                            INC     (R0)+       ;LOC. 0=400   R0=2
2961  007362  005400                            NEG     R0          ;R0=-2
2962  007364  060037  000000                    ADD     R0,@#0      ;TRY ADD 0,3;  LOC. 0=376
2963  007370  100403                            BMI     DOP03A      ;CC=0001?
2964  007372  001402                            BEQ     DOP03A
2965  007374  102401                            BVS     DOP03A
2966  007376  103404                            BCS     DOP03B
2967                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2968                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2969                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2970                                                  ;           WHICH FOLLOWS W/ 765           <====
2971  007400                            DOP03A:
2972  007400  012742  000211                    MOV     #211,-(R2)  ;MOVE TO MAILBOX #  *******  211  *******
2973  007404  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2974  007406  000000                            HALT                ;CC'S NOT SET CORRECTLY
2975  007410  105137  000000            DOP03B: COMB    @#0         ;LOC. 0=1
2976  007414  005337  000000                    DEC     @#0         ;LOC. 0=0
2977  007420  001404                            BEQ     TST115
2978                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
2979                                                  ;           CONDITIONAL BRANCH INST. AND   <====
2980                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
2981                                                  ;           WHICH FOLLOWS W/ 754           <====
2982  007422  012742  000212                    MOV     #212,-(R2)  ;MOVE TO MAILBOX #  *******  212  *******
2983  007426  005242                            INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
2984  007430  000000                            HALT                ;DATA RESULT INCORRECT
2985                                                  ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
                        ;R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY.  COMPARE INSTRUCTIONS ARE
                        ;THEN EXECUTED AND CHECKED.  FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
                        ;
                        ;*****************************************************************************
                        ;TEST 115       TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
                        ;*****************************************************************************
007432  005212          TST115: INC     (R2)            ;UPDATE TEST NUMBER
007434  022712  000115          CMP     #115,(R2)       ;SEQUENCE ERROR?
007440  001042                  BNE     TST116-10       ;BR TO ERROR HALT ON SEQ ERROR
007442  005000                  CLR     R0              ;R0=0
007444  005004                  CLR     R4              ;R4=0
007446  005204                  INC     R4              ;R4=1
007450  020400                  CMP     R4,R0           ;TRY COMPARE R4 TO R0
007452  003004                  BGT     DNM1
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 773           <====
007454  012742  000213          MOV     #213,-(R2)      ;MOVE TO MAILBOX #  *******  213  *******
007460  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007462  000000                  HALT                    ;CC'S NOT CORRECT FOR CMP
007464  020004          DNM1:   CMP     R0,R4           ;TRY COMPARE R0 TO R4
007466  002404                  BLT     DNM2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
007470  012742  000214          MOV     #214,-(R2)      ;MOVE TO MAILBOX #  *******  214  *******
007474  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007476  000000                  HALT                    ;CC'S NOT CORRECT FOR CMP
007500  005200          DNM2:   INC     R0              ;R0=1
007502  020400                  CMP     R4,R0           ;TRY COMPARE R4=1 TO R0=1
007504  001404                  BEQ     DNM3
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
007506  012742  000215          MOV     #215,-(R2)      ;MOVE TO MAILBOX #  *******  215  *******
007512  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007514  000000                  HALT                    ;CC'S NOT CORRECT (Z=1) FOR CMP
007516  005000          DNM3:   CLR     R0              ;R0=0
007520  005100                  COM     R0              ;R0=177777
007522  005004                  CLR     R4              ;R4=0
007524  030004                  BIT     R0,R4           ;TRY BIT R0 TO R4
007526  001404                  BEQ     DNM4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 745           <====
007530  012742  000216          MOV     #216,-(R2)      ;MOVE TO MAILBOX #  *******  216  *******
007534  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007536  000000                  HALT                    ;CC'S NOT CORRECT FOR BIT
007540  005304          DNM4:   DEC     R4              ;R4=177777
```



```
007542  030004                  BIT     R0,R4           ;TRY BIT AGAIN
007544  100404                  BMI     TST116
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 736           <====
007546  012742  000217          MOV     #217,-(R2)      ;MOVE TO MAILBOX #  *******  217  *******
007552  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007554  000000                  HALT                    ;CC'S NOT CORRECT FOR BIT
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
                        ;IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
                        ;
                        ;*****************************************************************************
                        ;TEST 116       TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.
                        ;*****************************************************************************
007556  005212          TST116: INC     (R2)            ;UPDATE TEST NUMBER
007560  022712  000116          CMP     #116,(R2)       ;SEQUENCE ERROR?
007564  001022                  BNE     TST117-10       ;BR TO ERROR HALT ON SEQ ERROR
007566  005000                  CLR     R0              ;R0=0
007570  005010                  CLR     (R0)            ;LOC. 0=0
007572  005110                  COM     (R0)            ;LOC. 0=177777
007574  005200                  INC     R0              ;R0=1
007576  020037  000000          CMP     R0,@#0          ;TRY CMP MODE 0,3
007602  100403                  BMI     DNM03A          ;CC=0001
007604  001402                  BEQ     DNM03A
007606  102401                  BVS     DNM03A
007610  103404                  BCS     DNM03B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
007612          DNM03A:
007612  012742  000220          MOV     #220,-(R2)      ;MOVE TO MAILBOX #  *******  220  *******
007616  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007620  000000                  HALT                    ;CC'S NOT SET CORRECTLY
007622  005300          DNM03B: DEC     R0
007624  001002                  BNE     DNM03C
007626  005210                  INC     (R0)
007630  001404                  BEQ     TST117
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
007632          DNM03C:
007632  012742  000221          MOV     #221,-(R2)      ;MOVE TO MAILBOX #  *******  221  *******
007636  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007640  000000                  HALT                    ;DATA INCORRECTLY MODIFIED BY CMP
                                                        ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS.  R0 IS SET TO -1
                        ;AND LOC 0 TO 1.  R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.
                        ;IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE
                        ;RESULTS VERIFIED.
                        ;
                        ;*****************************************************************************
                        ;TEST 117        TEST MODE 1 W/ DOP INST.
                        ;*****************************************************************************
007642  005212          TST117: INC     (R2)            ;UPDATE TEST NUMBER
007644  022712  000117          CMP     #117,(R2)       ;SEQUENCE ERROR?
007650  001007                  BNE     TST120-10       ;BR TO ERROR HALT ON SEQ ERROR
007652  005000                  CLR     R0              ;R0=0
007654  005100                  COM     R0              ;R0=177777
007656  005004                  CLR     R4              ;R4=0
007660  005014                  CLR     (R4)            ;LOC 0=0
007662  005214                  INC     (R4)            ;LOC 0=1
007664  061400                  ADD     (R4),R0         ;TRY ADD SOURCE MODE 1
007666  001404                  BEQ     TST120
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 771            <====
007670  012742  000222          MOV     #222,-(R2)      ;MOVE TO MAILBOX #  *******  222  *******
007674  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007676  000000                  HALT                    ;RESULT OF ADD INCORRECT
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       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.
                        ;*****************************************************************************
007700  005212          TST120: INC     (R2)            ;UPDATE TEST NUMBER
007702  022712  000120          CMP     #120,(R2)       ;SEQUENCE ERROR?
007706  001007                  BNE     TST121-10       ;BR TO ERROR HALT ON SEQ ERROR
007710  005000                  CLR     R0              ;R0=0
007712  005010                  CLR     (R0)            ;LOC. 0=0
007714  005110                  COM     (R0)            ;LOC. 0=177777
007716  005004                  CLR     R4              ;R4=0
007720  151004                  BISB    (R0),R4         ;TRY MODE 1- EVEN BYTE W/ DOP
007722  105104                  COMB    R4              ;R4=0
007724  001404                  BEQ     TST121
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 771            <====
007726  012742  000223          MOV     #223,-(R2)      ;MOVE TO MAILBOX #  *******  223  *******
007732  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007734  000000                  HALT                    ;RESULT OF BISB IS INCORRECT
                                                        ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 1 DOP NON-MODIFYING INSTRUCTIONS
                        ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO -1 AND R0 IS CLEARED
                        ;AND USED AS THE ADDRESSING REGISTER.  R4 IS SET TO 377 AND A
                        ;MODE 1,0 CMPB INSTRUCTION IS USED THE RESULTS VERIFIED.
                        ;
                        ;*****************************************************************************************
                        ;TEST 121       TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
                        ;*****************************************************************************************
007736  005212         TST121: INC     (R2)            ;UPDATE TEST NUMBER
007740  022712  000121         CMP     #121,(R2)       ;SEQUENCE ERROR?
007744  001007                 BNE     TST122-10       ;BR TO ERROR HALT ON SEQ ERROR
007746  005000                 CLR     R0              ;R0=0
007750  005010                 CLR     (R0)            ;LOC 0=0
007752  005110                 COM     (R0)            ;LOC 0=177777
007754  005004                 CLR     R4              ;R4=0
007756  105104                 COMB    R4              ;R4=377
007760  121004                 CMPB    (R0),R4         ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
007762  001404                 BEQ     TST122
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 771           <====
007764  012742  000224         MOV     #224,-(R2)      ;MOVE TO MAILBOX #  *******  224  *******
007770  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
007772  000000                 HALT                    ;RESULT OF CMPB INCORRECT
                                                       ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
                        ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
                        ;R4 IS SET TO -1.  MOVB ARE USED TO MOVE BYTE 0 TO R4.  THIS
                        ;VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
                        ;FUNCTION WITH MODE 0.
                        ;       THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
                        ;THE LOGIC FOR COMPLEMENTARY DATA.
                        ;       THIS TEST EXERCISES UNIQUE MICROCODE.
                        ;
                        ;*****************************************************************************************
                        ;TEST 122       TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
                        ;*****************************************************************************************
007774  005212         TST122: INC     (R2)            ;UPDATE TEST NUMBER
007776  022712  000122         CMP     #122,(R2)       ;SEQUENCE ERROR?
010002  001020                 BNE     TST123-10       ;BR TO ERROR HALT ON SEQ ERROR
010004  005000                 CLR     R0              ;R0=0
010006  005010                 CLR     (R0)            ;LOC 0=0
010010  105110                 COMB    (R0)            ;LOC 0=177400
010012  005110                 COM     (R0)
010014  005004                 CLR     R4              ;R4=0
010016  005104                 COM     R4              ;R4=177777
010020  111004                 MOVB    (R0),R4         ;R4=0
010022  005704                 TST     R4              ;CHECK SIGN OF WORD
010024  001404                 BEQ     DOP1
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 767           <====
010026  012742  000225         MOV     #225,-(R2)      ;MOVE TO MAILBOX #  *******  225  *******
010032  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010034  000000                 HALT                    ;MOVB SHOULD SIGN X-TEND
010036  005110         DOP1:   COM     (R0)            ;LOC 0=177777
010040  111004                 MOVB    (R0),R4         ;DO MOVB W/ EVEN BYTE
010042  100404                 BMI     TST123
                                                       ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                       ;           CONDITIONAL BRANCH INST. AND   <====
                                                       ;           REPLACE THE MOVE INSTRUCTION   <====
                                                       ;           WHICH FOLLOWS W/ 760           <====
010044  012742  000226         MOV     #226,-(R2)      ;MOVE TO MAILBOX #  *******  226  *******
010050  005242                 INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010052  000000                 HALT                    ;MOVB SHOULD SIGN X-TEND
                                                       ;  OR SEQUENCE ERROR
```

```
                        ;******************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE
                        ;ODD BYTES.  LOC. 0 IS SET TO 177400.  R0 IS SET TO 0 AND R4 IS
                        ;SET TO 1.  THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.
                        ;THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.
                        ;
                        ;******************************************************************************
                        ;TEST 123        TEST MODE 1-ODD BYTE W/ DOP INSTS.
                        ;******************************************************************************
010054  005212          TST123: INC     (R2)            ;UPDATE TEST NUMBER
010056  022712  000123          CMP     #123,(R2)       ;SEQUENCE ERROR?
010062  001010                  BNE     TST124-10       ;BR TO ERROR HALT ON SEQ ERROR
010064  005000                  CLR     R0              ;R0=0
010066  005010                  CLR     (R0)            ;LOC. 0=0
010070  005004                  CLR     R4              ;R4=0
010072  005204                  INC     R4              ;R4=1
010074  105114                  COMB    (R4)            ;LOC. 0=177400
010076  151410                  BISB    (R4),(R0)       ;TRY TO BIS LOW ORDER BITS W/ MODE 1
010100  005210                  INC     (R0)            ;CHECK RESULT
010102  001404                  BEQ     TST124
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
010104  012742  000227          MOV     #227,-(R2)      ;MOVE TO MAILBOX #  *******  227  *******
010110  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010112  000000                  HALT                    ;RESULT OF BISB INCORRECT
                                                        ;  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.
                        ;******************************************************************************
010114  005212          TST124: INC     (R2)            ;UPDATE TEST NUMBER
010116  022712  000124          CMP     #124,(R2)       ;SEQUENCE ERROR?
010122  001015                  BNE     TST125-10       ;BR TO ERROR HALT ON SEQ ERROR
010124  005000                  CLR     R0              ;R0=0
010126  005010                  CLR     (R0)            ;LOC. 0=0
010130  005110                  COM     (R0)            ;LOC. 0=177777
010132  012004                  MOV     (R0)+,R4        ;TRY MOVE MODE 2,0
010134  005204                  INC     R4              ;CHECK R4
010136  001404                  BEQ     DOP2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 772           <====
010140  012742  000230          MOV     #230,-(R2)      ;MOVE TO MAILBOX #  *******  230  *******
010144  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010146  000000                  HALT                    ;RESULT OF MOV INST INCORRECT
010150  005300          DOP2:   DEC     R0              ;TEST R0 AFTER MODE 2
010152  005300                  DEC     R0
010154  001404                  BEQ     TST125
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
010156  012742  000231          MOV     #231,-(R2)      ;MOVE TO MAILBOX #  *******  231  *******
010162  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010164  000000                  HALT                    ;REGISTER NOT INCREMENTED IN MODE 2
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS
                                ;EVEN BYTES.  LOC. 0 IS SET TO -1.  R0 IS CLEARED AND USED AS THE
                                ;ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING
                                ;BYTE 0 DATA AND A BICB.  UNIQUE IN THIS TEST IS USE OF THE
                                ;SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION.  THE SOURCE AND
                                ;DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.
                                ;
                                ;*****************************************************************************
                                ;TEST 125        TEST MODE 2 - EVEN BYTE W/ DOP INST.
                                ;*****************************************************************************
010166  005212                TST125: INC     (R2)            ;UPDATE TEST NUMBER
010170  022712  000125                CMP     #125,(R2)       ;SEQUENCE ERROR?
010174  001016                       BNE     TST126-10       ;BR TO ERROR HALT ON SEQ ERROR
010176  005000                       CLR     R0              ;R0=0
010200  010010                       MOV     R0,(R0)         ;LOC.  0=0
010202  005110                       COM     (R0)            ;LOC.  0=177777
010204  142010                       BICB    (R0)+,(R0)      ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
010206  105737  000001                TSTB    @#1             ;CHECK RESULT
010212  001404                       BEQ     DOPB2A
                                                             ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                             ;           CONDITIONAL BRANCH INST. AND   <====
                                                             ;           REPLACE THE MOVE INSTRUCTION   <====
                                                             ;           WHICH FOLLOWS W/ 771           <====
010214  012742  000232                MOV     #232,-(R2)      ;MOVE TO MAILBOX #  *******  232  *******
010220  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010222  000000                       HALT                    ;BICB DESTINATION INCORRECT
010224  105137  000000        DOPB2A: COMB    @#0             ;CHECK BICB SOURCE
010230  001404                       BEQ     TST126
                                                             ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                             ;           CONDITIONAL BRANCH INST. AND   <====
                                                             ;           REPLACE THE MOVE INSTRUCTION   <====
                                                             ;           WHICH FOLLOWS W/ 762           <====
010232  012742  000233                MOV     #233,-(R2)      ;MOVE TO MAILBOX #  *******  233  *******
010236  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010240  000000                       HALT                    ;BICB SOURCE INCORRECTLY CHANGED
                                                             ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE
                                ;ODD BYTES.  R0 IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.
                                ;A MODE 2 MOVB USES R0 TO MOVE BYTE 1 TO R4.  AN INCREMENT
                                ;IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.
                                ;
                                ;*****************************************************************************
                                ;TEST 126        TEST MODE 2 - ODD BYTE W/ DOP INST.
                                ;*****************************************************************************
010242  005212                TST126: INC     (R2)            ;UPDATE TEST NUMBER
010244  022712  000126                CMP     #126,(R2)       ;SEQUENCE ERROR?
010250  001017                       BNE     TST127-10       ;BR TO ERROR HALT ON SEQ ERROR
010252  005000                       CLR     R0              ;R0=0
010254  005004                       CLR     R4              ;R4=0
010256  005010                       CLR     (R0)            ;LOC.  0=0
010260  005110                       COM     (R0)            ;LOC.  0=177777
010262  105120                       COMB    (R0)+           ;LOC 0=177400; R0=1
010264  112004                       MOVB    (R0)+,R4        ;TRY DOP MODE 2 W/ ODD BYTE
010266  005204                       INC     R4              ;CHECK RESULT OF MOVB
010270  001404                       BEQ     DOPB2B
                                                             ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                             ;           CONDITIONAL BRANCH INST. AND   <====
                                                             ;           REPLACE THE MOVE INSTRUCTION   <====
                                                             ;           WHICH FOLLOWS W/ 770           <====
010272  012742  000234                MOV     #234,-(R2)      ;MOVE TO MAILBOX #  *******  234  *******
010276  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010300  000000                       HALT                    ;RESULT OF MOVB INCORRECT
010302  005740                DOPB2B: TST     -(R0)           ;BUMP R0 DOWN BY 2
010304  005700                       TST     R0              ;CHECK R0
010306  001404                       BEQ     TST127
                                                             ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                             ;           CONDITIONAL BRANCH INST. AND   <====
                                                             ;           REPLACE THE MOVE INSTRUCTION   <====
                                                             ;           WHICH FOLLOWS W/ 761           <====
010310  012742  000235                MOV     #235,-(R2)      ;MOVE TO MAILBOX #  *******  235  *******
010314  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010316  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.
                                ;*****************************************************************************
010320  005212                  TST127: INC     (R2)            ;UPDATE TEST NUMBER
010322  022712  000127                  CMP     #127,(R2)       ;SEQUENCE ERROR?
010326  001011                          BNE     TST130-10       ;BR TO ERROR HALT ON SEQ ERROR
010330  012737  052525  000000          MOV     #052525,@#0     ;MOVE 52525 TO LOC. 0
010336  012700  125252                  MOV     #125252,R0      ;SET ALT. ONE AND ZERO IN R0
010342  053700  000000                  BIS     @#0,R0          ;TRY TO SET ALL OTHER BITS W/ MODE 3
010346  005200                          INC     R0              ;TEST RESULT
010350  001404                          BEQ     TST130
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
010352  012742  000236                  MOV     #236,-(R2)      ;MOVE TO MAILBOX #  *******  236  *******
010356  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010360  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.
                                ;*****************************************************************************
010362  005212                  TST130: INC     (R2)            ;UPDATE TEST NUMBER
010364  022712  000130                  CMP     #130,(R2)       ;SEQUENCE ERROR?
010370  001011                          BNE     TST131-10       ;BR TO ERROR HALT ON SEQ ERROR
010372  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0' PATTERN TO LOC. 0
010400  005000                          CLR     R0              ;R0=0
010402  153700  000000                  BISB    @#0,R0          ;TRY R0=252 W/ MODE 3 - EVEN BYTE
010406  022700  000252                  CMP     #252,R0         ;BISB W/ EVEN BYTE SUCCESSFUL?
010412  001404                          BEQ     TST131
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
010414  012742  000237                  MOV     #237,-(R2)      ;MOVE TO MAILBOX #  *******  237  *******
010420  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010422  000000                          HALT                    ;BISB W/ MODE 3 - EVEN BYTE FAILED
                                                                ;  OR SEQUENCE ERROR
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
                                ;WHICH ADDRESS ODD BYTES.  THE SAME PROCEDURE USED IN PREVIOUS
                                ;TEST IS USED HERE.  THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
                                ;THE EXPECTED RESULT IS:  R0 = 125.
                                ;
                                ;*****************************************************************************
                                ;TEST 131        TEST MODE 3 - ODD BYTE W/ DOP INSTS.
                                ;*****************************************************************************
010424  005212                  TST131: INC     (R2)            ;UPDATE TEST NUMBER
010426  022712  000131                  CMP     #131,(R2)       ;SEQUENCE ERROR?
010432  001011                          BNE     TST132-10       ;BR TO ERROR HALT ON SEQ ERROR
010434  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0'S PATTERN TO LOC 0
010442  005000                          CLR     R0              ;R0=0
010444  153700  000001                  BISB    @#1,R0          ;TRY R0=152 W/ MODE 3 - ODD BYTE
010450  022700  000125                  CMP     #125,R0         ;R0=125?
010454  001404                          BEQ     TST132
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
010456  012742  000240                  MOV     #240,-(R2)      ;MOVE TO MAILBOX #  *******  240  *******
010462  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010464  000000                          HALT                    ;BISB W/ MODE 3 - ODD BYTE FAILED
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 132        TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
                                ;*****************************************************************************
010466  005212                  TST132: INC     (R2)            ;UPDATE TEST NUMBER
010470  022712  000132                  CMP     #132,(R2)       ;SEQUENCE ERROR?
010474  001017                          BNE     TST133-10       ;BR TO ERROR HALT ON SEQ ERROR
010476  005000                          CLR     R0              ;R0=0
010500  105100                          COMB    R0              ;R0=377
010502  000263                          +SEC!SEV                ;SET C AND V BITS
010504  132700  000200                  BITB    #200,R0         ;TRY DOPNM DEST. MODE 0-BYTE
010510  001403                          BEQ     DNMB0A          ;BR TO ERROR IF Z BIT SET
010512  102402                          BVS     DNMB0A          ;BR TO ERROR IF V BIT SET
010514  103001                          BCC     DNMB0A          ;BR TO ERROR IF C BIT CLEAR.
010516  100404                          BMI     DNMB0B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
010520                          DNMB0A:
010520  012742  000241                  MOV     #241,-(R2)      ;MOVE TO MAILBOX #  *******  241  *******
010524  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
010526  000000                          HALT                    ;CC'S INCORRECT
010530  105100                  DNMB0B: COMB    R0              ;CHECK DESTINATION DATA
010532  001404                          BEQ     TST133
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 761           <====
010534  012742  000242                  MOV     #242,-(R2)      ;MOVE TO MAILBOX #  *******  242  *******
```

```
3478  010540  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3479  010542  000000                        HALT                    ;DEST. DATA MODIFIED
3480                                                                  ;  OR SEQUENCE ERROR
3481
3482                                  ;*************************************************************************
3483                                  ;TEST 133       TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3484                                  ;*************************************************************************
3485  010544  005212                  TST133: INC     (R2)            ;UPDATE TEST NUMBER
3486  010546  022712  000133                 CMP     #133,(R2)       ;SEQUENCE ERROR?
3487  010552  001017                         BNE     TST134-10       ;BR TO ERROR HALT ON SEQ ERROR
3488  010554  005000                         CLR     R0              ;R0=0
3489  010556  005010                         CLR     (R0)            ;LOC. 0=0
3490  010560  000241                         CLC                     ;CLEAR C BIT
3491  010562  032710  177777                BIT     #177777,(R0)    ;TRY DOPNM DEST. MODE 1
3492  010566  100403                         BMI     DNM1A           ;BR TO ERROR IF N BIT SET
3493  010570  102402                         BVS     DNM1A           ;BR TO ERROR IF V BIT SET
3494  010572  103401                         BCS     DNM1A           ;BR TO ERROR IF C BIT SET
3495  010574  001404                         BEQ     DNM1B
3496                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3497                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3498                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3499                                                                  ;           WHICH FOLLOWS W/ 767           <====
3500  010576                          DNM1A:
3501  010576  012742  000243                 MOV     #243,-(R2)      ;MOVE TO MAILBOX #  *******  243  *******
3502  010602  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3503  010604  000000                         HALT                    ;COND. CODES INCORRECT
3504  010606  005710                  DNM1B: TST     (R0)            ;CHECK TEST DATA
3505  010610  001404                         BEQ     TST134
3506                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3507                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3508                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3509                                                                  ;           WHICH FOLLOWS W/ 761           <====
3510  010612  012742  000244                 MOV     #244,-(R2)      ;MOVE TO MAILBOX #  *******  244  *******
3511  010616  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3512  010620  000000                         HALT                    ;DESTINATION DATA MODIFIED
3513                                                                  ;  OR SEQUENCE ERROR
3514
3515                                  ;*************************************************************************
3516                                  ;TEST 134       TEST DEST. MODE 2 W/ DOP NON-MODIFYING INST.
3517                                  ;*************************************************************************
3518  010622  005212                  TST134: INC     (R2)            ;UPDATE TEST NUMBER
3519  010624  022712  000134                 CMP     #134,(R2)       ;SEQUENCE ERROR?
3520  010630  001027                         BNE     TST135-10       ;BR TO ERROR HALT ON SEQ ERROR
3521  010632  005000                         CLR     R0              ;R0=0
3522  010634  005010                         CLR     (R0)            ;LOC. 0=0
3523  010636  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125252
3524  010642  032720  077777                BIT     #77777,(R0)+    ;TRY DOPNM INST W/ MODE 2
3525  010646  102402                         BVS     DNM2A           ;BR TO ERROR IF V BIT SET
3526  010650  001401                         BEQ     DNM2A           ;BR TO ERROR IF Z-BIT SET
3527  010652  100004                         BPL     DNM2B
3528                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3529                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3530                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3531                                                                  ;           WHICH FOLLOWS W/ 767           <====
3532  010654                          DNM2A:
3533  010654  012742  000245                 MOV     #245,-(R2)      ;MOVE TO MAILBOX #  *******  245  *******
```



```
3534  010660  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3535  010662  000000                        HALT                    ;COND. CODES INCORRECT
3536  010664  005300                  DNM2B: DEC     R0              ;DECREMENT R0 TO CHECK IT.
3537  010666  005300                         DEC     R0
3538  010670  001404                         BEQ     DNM2D
3539                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3540                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3541                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3542                                                                  ;           WHICH FOLLOWS W/ 760           <====
3543  010672                          DNM2C:
3544  010672  012742  000246                 MOV     #246,-(R2)      ;MOVE TO MAILBOX #  *******  246  *******
3545  010676  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3546  010700  000000                         HALT                    ;MODE 2 REGISTER NOT INCREMENTED BY 2
3547  010702  022710  125252          DNM2D: CMP     #125252,(R0)    ;CHECK DEST. DATA
3548  010706  001404                         BEQ     TST135
3549                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3550                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3551                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3552                                                                  ;           WHICH FOLLOWS W/ 751           <====
3553  010710  012742  000247                 MOV     #247,-(R2)      ;MOVE TO MAILBOX #  *******  247  *******
3554  010714  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3555  010716  000000                         HALT                    ;DEST. DATA MODIFIED
3556                                                                  ;  OR SEQUENCE ERROR
3557
3558                                  ;*************************************************************************
3559                                  ;TEST 135       TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
3560                                  ;*************************************************************************
3561  010720  005212                  TST135: INC     (R2)            ;UPDATE TEST NUMBER
3562  010722  022712  000135                 CMP     #135,(R2)       ;SEQUENCE ERROR?
3563  010726  001051                         BNE     TST136-10       ;BR TO ERROR HALT ON SEQ ERROR
3564  010730  005000                         CLR     R0              ;R0=0
3565  010732  005010                         CLR     (R0)            ;LOC. 0=0
3566  010734  052710  052652                BIS     #52652,(R0)     ;LOC. 0=52652
3567  010740  000263                         .SEC!SEV                ;SET C AND V BITS
3568  010742  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
3569  010746  001403                         BEQ     DNMB2A          ;BR TO ERROR IF Z-BIT SET
3570  010750  103002                         BCC     DNMB2A          ;BR TO ERROR IF C-BIT CLEAR
3571  010752  102401                         BVS     DNMB2A          ;BR TO ERROR IF V-BIT SET
3572  010754  100404                         BMI     DNMB2B
3573                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3574                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3575                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3576                                                                  ;           WHICH FOLLOWS W/ 765           <====
3577  010756                          DNMB2A:
3578  010756  012742  000250                 MOV     #250,-(R2)      ;MOVE TO MAILBOX #  *******  250  *******
3579  010762  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3580  010764  000000                         HALT                    ;COND. CODES INCORRECT
3581  010766  005300                  DNMB2B: DEC     R0              ;CHECK DEST. REGISTER.
3582  010770  001404                         BEQ     DNMB2C
3583                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
3584                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
3585                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
3586                                                                  ;           WHICH FOLLOWS W/ 757           <====
3587  010772  012742  000251                 MOV     #251,-(R2)      ;MOVE TO MAILBOX #  *******  251  *******
3588  010776  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3589  011000  000000                         HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
```

```
3590  011002  005200            DNMB2C: INC     R0              ;R0=1
3591  011004  132720  000201            BITB    #201,(R0)+      ;TRY DOPNM INST. W/MODE 2-ODD BYTE
3592  011010  001402                    BEQ     DNMB2D  ;BR TO ERROR IF Z-BIT SET
3593  011012  102401                    BVS     DNMB2D          ;BR TO ERROR IF V-BIT SET
3594  011014  100004                    BPL     DNMB2E
3595                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3596                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3597                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3598                                                            ;           WHICH FOLLOWS W/ 745            <====
3599  011016                    DNMB2D:
3600  011016  012742  000252            MOV     #252,-(R2)      ;MOVE TO MAILBOX #  *******  252  *******
3601  011022  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3602  011024  000000                    HALT                    ;COND. CODES INCORRECT
3603  011026  005300            DNMB2E: DEC     R0              ;DEC R0 TO CHECK IT.
3604  011030  005300                    DEC     R0
3605  011032  001404                    BEQ     DNMB2F
3606                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3607                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3608                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3609                                                            ;           WHICH FOLLOWS W/ 736            <====
3610  011034  012742  000253            MOV     #253,-(R2)      ;MOVE TO MAILBOX #  *******  253  *******
3611  011040  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3612  011042  000000                    HALT                    ;DEST. REGISTER NOT INCREMENTED BY 1
3613  011044  022710  052652    DNMB2F: CMP     #52652,(R0)     ;CHECK DEST. DATA IS UNMODIFIED
3614  011050  001404                    BEQ     TST136
3615                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3616                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3617                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3618                                                            ;           WHICH FOLLOWS W/ 727            <====
3619  011052  012742  000254            MOV     #254,-(R2)      ;MOVE TO MAILBOX #  *******  254  *******
3620  011056  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3621  011060  000000                    HALT                    ;DEST. DATA WAS MODIFIED.
3622                                                            ;  OR SEQUENCE ERROR
3623
3624
3625                                ;*****************************************************************************
3626                                ;TEST 136       TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
3627                                ;*****************************************************************************
3628  011062  005212            TST136: INC     (R2)            ;UPDATE TEST NUMBER
3629  011064  022712  000136            CMP     #136,(R2)       ;SEQUENCE ERROR?
3630  011070  001050                    BNE     TST137-10       ;BR TO ERROR HALT ON SEQ ERROR
3631  011072  005000                    CLR     R0              ;R0=0
3632  011074  005010                    CLR     (R0)            ;LOC. 0=0
3633  011076  052710  125125            BIS     #125125,(R0)    ;LOC. 0=125125
3634  011102  105100                    COMB    R0              ;R0=377
3635  011104  005200                    INC     R0              ;R0=400
3636  011106  005010                    CLR     (R0)            ;LOC. 400=0
3637  011110  000263                    +SEC!SEV                ;C-BIT=V-BIT=1
3638  011112  132730  000201            BITB    #201,@(R0)+     ;TRY DOPNM W/MODE 3-EVEN BYTE
3639  011116  001403                    BEQ     DNMB3A          ;BR TO ERROR IF Z BIT SET
3640  011120  102402                    BVS     DNMB3A          ;BR TO ERROR IF V BIT SET
3641  011122  103001                    BCC     DNMB3A          ;BR TO ERROR IF C BIT CLEAR
3642  011124  100004                    BPL     DNMB3B
3643                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3644                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3645                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
```



```
3646                                                            ;           WHICH FOLLOWS W/ 762            <====
3647  011126                    DNMB3A:
3648  011126  012742  000255            MOV     #255,-(R2)      ;MOVE TO MAILBOX #  *******  255  *******
3649  011132  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3650  011134  000000                    HALT                    ;COND. CODES INCORRECT
3651  011136  022700  000402    DNMB3B: CMP     #402,R0         ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
3652  011142  001404                    BEQ     DNMB3C
3653                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3654                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3655                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3656                                                            ;           WHICH FOLLOWS W/ 753            <====
3657  011144  012742  000256            MOV     #256,-(R2)      ;MOVE TO MAILBOX #  *******  256  *******
3658  011150  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3659  011152  000000                    HALT                    ;DEST. REGISTER NOT INCREMENTED BY 2
3660  011154  005200            DNMB3C: INC     R0              ;R0=404
3661  011156  005200                    INC     R0
3662  011160  132730  000201            BITB    #201,@(R0)+     ;TRY DOPNM DEST MODE 3-BYTE(ODD)
3663  011164  001402                    BEQ     DNMB3D          ;BR TO ERROR IF Z BIT SET
3664  011166  102401                    BVS     DNMB3D          ;BR TO ERROR IF V BIT SET
3665  011170  100404                    BMI     DNMB3E
3666                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3667                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3668                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3669                                                            ;           WHICH FOLLOWS W/ 740            <====
3670  011172                    DNMB3D:
3671  011172  012742  000257            MOV     #257,-(R2)      ;MOVE TO MAILBOX #  *******  257  *******
3672  011176  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3673  011200  000000                    HALT                    ;COND. CODES INCORRECT
3674  011202  005004            DNMB3E: CLR     R4              ;R4=0
3675  011204  022714  125125            CMP     #125125,(R4)    ;CHECK DEST. DATA
3676  011210  001404                    BEQ     TST137
3677                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
3678                                                            ;           CONDITIONAL BRANCH INST. AND    <====
3679                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
3680                                                            ;           WHICH FOLLOWS W/ 730            <====
3681  011212  012742  000260            MOV     #260,-(R2)      ;MOVE TO MAILBOX #  *******  260  *******
3682  011216  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
3683  011220  000000                    HALT                    ;DEST. DATA MODIFIED
3684                                                            ;  OR SEQUENCE ERROR
3685
3686                                ;*****************************************************************************
3687                                ;TEST 137       TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
3688                                ;*****************************************************************************
3689  011222  005212            TST137: INC     (R2)            ;UPDATE TEST NUMBER
3690  011224  022712  000137            CMP     #137,(R2)       ;SEQUENCE ERROR?
3691  011230  001033                    BNE     TST140-10       ;BR TO ERROR HALT ON SEQ ERROR
3692  011232  005000                    CLR     R0              ;R0=0
3693  011234  005010                    CLR     (R0)            ;LOC. 0=0
3694  011236  052710  125252            BIS     #125252,(R0)    ;LOC. 0=125125
3695  011242  052700  000002            BIS     #2,R0           ;R0=2
3696  011246  000277                    SCC                     ;SET ALL COND. CODE BITS
3697  011250  032740  020000            BIT     #20000,-(R0)    ;TRY DOPNM W/ MODE 4
3698  011254  100403                    BMI     DNM4A           ;BR TO ERROR IF N-BIT SET
3699  011256  102402                    BVS     DNM4A           ;BR TO ERROR IF V-BIT SET
3700  011260  103001                    BCC     DNM4A           ;BR TO ERROR IF C-BIT CHAR
3701  011262  001004                    BNE     DNM4B
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
011264                          DNM4A:
011264  012742  000261                MOV     #261,-(R2)      ;MOVE TO MAILBOX #  *******  261  *******
011270  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011272  000000                  HALT                    ;COND. CODES INCORRECT
011274  005700                  DNM4B:  TST     R0              ;CHECK DEST. REGISTER
011276  001404                  BEQ     DNM4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 755           <====
011300  012742  000262                MOV     #262,-(R2)      ;MOVE TO MAILBOX #  *******  262  *******
011304  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011306  000000                  HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
011310  022737  125252  000000  DNM4C:  CMP     #125252,@#0     ;CHECK DEST. DATA
011316  001404                  BEQ     TST140
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 745           <====
011320  012742  000263                MOV     #263,-(R2)      ;MOVE TO MAILBOX #  *******  263  *******
011324  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011326  000000                  HALT                    ;DEST. DATA MODIFIED
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 140       TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
                        ;*****************************************************************************
011330  005212          TST140: INC     (R2)            ;UPDATE TEST NUMBER
011332  022712  000140          CMP     #140,(R2)       ;SEQUENCE ERROR?
011336  001051                  BNE     TST141-10       ;BR TO ERROR HALT ON SEQ ERROR
011340  005000                  CLR     R0              ;R0=0
011342  005010                  CLR     (R0)            ;LOC. 0=0
011344  052710  052652          BIS     #52652,(R0)     ;LOC. 0=52652
011350  052700  000002          BIS     #2,R0           ;R0=2
011354  000257                  CCC                     ;COND. CODES=0
011356  132740  000201          BITB    #201,-(R0)      ;TRY DOPNM INST W/MODE 4 ODD BYTE
011362  102403                  BVS     DNMB4A          ;BR TO ERROR IF V BIT SET
011364  001402                  BEQ     DNMB4A  ;BR TO ERROR IF Z BIT SET
011366  103401                  BCS     DNMB4A          ;BR TO ERROR IF C BIT SET
011370  001004                  BNE     DNMB4B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
011372                          DNMB4A:
011372  012742  000264                MOV     #264,-(R2)      ;MOVE TO MAILBOX #  *******  264  *******
011376  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011400  000000                  HALT                    ;COND. CODES INCORRECT
011402  022700  000001          DNMB4B: CMP     #1,R0           ;CHECK DEST. REGISTER
011406  001404                  BEQ     DNMB4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
```



```
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
011410  012742  000265                MOV     #265,-(R2)      ;MOVE TO MAILBOX #  *******  265  *******
011414  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011416  000000                  HALT                    ;DEST REG. NOT DECREMENTED BY 1
011420  132740  000201          DNMB4C: BITB    #201,-(R0)      ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
011424  001401                  BEQ     DNMB4D          ;BR TO ERROR IF Z-BIT SET
011426  100404                  BMI     DNMB4E
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 744           <====
011430                          DNMB4D:
011430  012742  000266                MOV     #266,-(R2)      ;MOVE TO MAILBOX #  *******  266  *******
011434  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011436  000000                  HALT                    ;COND. CODES INCORRECT
011440  005700                  DNMB4E: TST     R0              ;CHECK DEST. REGISTER
011442  001404                  BEQ     DNMB4F
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 736           <====
011444  012742  000267                MOV     #267,-(R2)      ;MOVE TO MAILBOX #  *******  267  *******
011450  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011452  000000                  HALT                    ;DEST. REG. NOT DECREMENTED BY 1
011454  022710  052652          DNMB4F: CMP     #52652,(R0)     ;CHECK DESTINATION DATA
011460  001404                  BEQ     TST141
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 727           <====
011462  012742  000270                MOV     #270,-(R2)      ;MOVE TO MAILBOX #  *******  270  *******
011466  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011470  000000                  HALT                    ;DEST. DATA MODIFIED
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 141       TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
                        ;*****************************************************************************
011472  005212          TST141: INC     (R2)            ;UPDATE TEST NUMBER
011474  022712  000141          CMP     #141,(R2)       ;SEQUENCE ERROR?
011500  001034                  BNE     TST142-10       ;BR TO ERROR HALT ON SEQ ERROR
011502  005000                  CLR     R0              ;R0=0
011504  005010                  CLR     (R0)            ;LOC 0=0
011506  052710  100000          BIS     #100000,(R0)    ;LOC. 0=100000
011512  052700  000402          BIS     #402,R0         ;R0=2
011516  000277                  SCC                     ;SET ALL COND. CODE BITS
011520  032750  100000          BIT     #100000,@-(R0)  ;TRY DOPNM W/MODE 5
011524  102403                  BVS     DNM5A           ;BR TO ERROR IF V-BIT SET
011526  103002                  BCC     DNM5A           ;BR TO ERROR IF C-BIT CLEAR
011530  001401                  BEQ     DNM5A           ;BR TO ERROR IF Z-BIT SET
011532  100404                  BMI     DNM5B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
```

```
011534                  DNM5A:
011534  012742  000271          MOV     #271,-(R2)      ;MOVE TO MAILBOX #  *******  271  *******
011540  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011542  000000                  HALT                    ;COND. CODES INCORRECT
011544  022700  000400  DNM5B:  CMP  #400,R0            ;CHECK DEST. REGISTER
011550  001404                  BEQ     DNM5C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 754            <====
011552  012742  000272          MOV     #272,-(R2)      ;MOVE TO MAILBOX #  *******  272  *******
011556  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011560  000000                  HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
011562  022737  100000  000000  DNM5C:  CMP  #100000,@#0  ;CHECK DESTINATION DATA
011570  001404                  BEQ     TST142
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 744            <====
011572  012742  000273          MOV     #273,-(R2)      ;MOVE TO MAILBOX #  *******  273  *******
011576  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011600  000000                  HALT                    ;DEST. DATA INCORRECTLY MODIFIED
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 142       TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
                        ;*****************************************************************************
011602  005212          TST142: INC     (R2)            ;UPDATE TEST NUMBER
011604  022712  000142          CMP     #142,(R2)       ;SEQUENCE ERROR?
011610  001033                  BNE     TST143-10       ;BR TO ERROR HALT ON SEQ ERROR
011612  005000                  CLR     R0              ;R0=0
011614  005010                  CLR     (R0)            ;LOC> 0=0
011616  052710  000001          BIS     #1,(R0)         ;LOC. 0=1
011622  005100                  COM     R0              ;R0=-1 C-BIT=1
011624  032760  000001  000001  BIT     #1,1(R0)        ;TRY DOPNM W/MODE 6
011632  001403                  BEQ     DNM6A           ;BR TO ERROR IF Z-BIT SET
011634  102402                  BVS     DNM6A           ;BR TO ERROR IF V-BIT SET
011636  103001                  BCC     DNM6A           ;BR TO ERROR IF C-BIT CLEAR
011640  100004                  BPL     DNM6B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 764            <====
011642                  DNM6A:
011642  012742  000274          MOV     #274,-(R2)      ;MOVE TO MAILBOX #  *******  274  *******
011646  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011650  000000                  HALT                    ;COND CODES INCORRECT
011652  022700  177777  DNM6B:  CMP     #-1,R0          ;CHECK DEST. REGISTER
011656  001404                  BEQ     DNM6C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 755            <====
011660  012742  000275          MOV     #275,-(R2)      ;MOVE TO MAILBOX #  *******  275  *******
011664  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011666  000000                  HALT                    ;DEST. REGISTER MODIFIED
```



```
011670  022737  000001  000000  DNM6C:  CMP     #1,@#0          ;CHECK DEST. DATA
011676  001404                  BEQ     TST143
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 745            <====
011700  012742  000276          MOV     #276,-(R2)      ;MOVE TO MAILBOX #  *******  276  *******
011704  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011706  000000                  HALT                    ;DEST. DATA MODIFIED
                                                        ;  OR SEQUENCE ERROR

                        ;*****************************************************************************
                        ;TEST 143       TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
                        ;*****************************************************************************
011710  005212          TST143: INC     (R2)            ;UPDATE TEST NUMBER
011712  022712  000143          CMP     #143,(R2)       ;SEQUENCE ERROR?
011716  001034                  BNE     TST144-10       ;BR TO ERROR HALT ON SEQ ERROR
011720  005000                  CLR     R0              ;R0=0
011722  005010                  CLR     (R0)            ;LOC. 0=0 C-BIT=0
011724  052710  125125          BIS     #125125,(R0)    ;LOC. 0=125125
011730  052700  000001          BIS     #1,R0           ;R0=1
011734  132770  000125  000403  BITB    #125,@403(R0)   ;TRY DOPNM W/MODE 7
011742  102403                  BVS     DNM7A           ;BR TO ERROR IF V-BIT SET
011744  100402                  BMI     DNM7A           ;BR TO ERROR IF N-BIT SET
011746  103401                  BCS     DNM7A           ;BR TO ERROR IF C-BIT SET
011750  001404                  BEQ     DNM7B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 763            <====
011752                  DNM7A:
011752  012742  000277          MOV     #277,-(R2)      ;MOVE TO MAILBOX #  *******  277  *******
011756  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011760  000000                  HALT                    ;COND. CODES INCORRECT
011762  022700  000001  DNM7B:  CMP     #1,R0           ;CHECK DEST. REGISTER
011766  001404                  BEQ     DNM7C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 754            <====
011770  012742  000300          MOV     #300,-(R2)      ;MOVE TO MAILBOX #  *******  300  *******
011774  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
011776  000000                  HALT                    ;DESTINATION REGISTER MODIFIED
012000  022737  125125  000000  DNM7C:  CMP     #125125,@#0     ;CHECK DEST. DATA
012006  001404                  BEQ     TST144
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 744            <====
012010  012742  000301          MOV     #301,-(R2)      ;MOVE TO MAILBOX #  *******  301  *******
012014  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012016  000000                  HALT                    ;DEST. DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR

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

```
                        ;       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
                        ;*****************************************************************************
012020  005212          TST144: INC     (R2)            ;UPDATE TEST NUMBER
012022  022712  000144          CMP     #144,(R2)       ;SEQUENCE ERROR?
012026  001016                  BNE     TST145-10       ;BR TO ERROR HALT ON SEQ ERROR
012030  005000                  CLR     R0              ;R0=0
012032  005010                  CLR     (R0)            ;LOC. 0=0
012034  005100                  COM     R0              ;R0=-1
012036  005004                  CLR     R4              ;R4 POINTS TO LOC. 0
012040  010014                  MOV     R0,(R4)         ;TRY MOVE MODE 0,1
012042  102402                  BVS     MDM1A           ;BR TO ERROR IF V SET
012044  001401                  BEQ     MDM1A           ;BR TO ERROR IF Z SET
012046  100404                  BMI     MDM1B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
012050                  MDM1A:
012050  012742  000302          MOV     #302,-(R2)      ;MOVE TO MAILBOX #  *******  302  *******
012054  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012056  000000                  HALT                    ;CONDITION CODE NOT CORRECT
012060  005704          MDM1B:  TST     R4
012062  001404                  BEQ     TST145
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
012064  012742  000303          MOV     #303,-(R2)      ;MOVE TO MAILBOX #  *******  303  *******
012070  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012072  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
                        ;*****************************************************************************
012074  005212          TST145: INC     (R2)            ;UPDATE TEST NUMBER
012076  022712  000145          CMP     #145,(R2)       ;SEQUENCE ERROR?
012102  001025                  BNE     TST146-10       ;BR TO ERROR HALT ON SEQ ERROR
012104  005000                  CLR     R0              ;R0=0
012106  005010                  CLR     (R0)            ;LOC.0=0
012110  005110                  COM     (R0)            ;LOC. 0= 1
012112  010020                  MOV     R0,(R0)+        ;TRY MOVE MODE 0,2
012114  100402                  BMI     MDM2A           ;BR TO ERROR IF N SET
012116  102401                  BVS     MDM2A           ;BR TO ERROR IF V SET
012120  001404                  BEQ     MDM2B
```



```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
012122                  MDM2A:
012122  012742  000304          MOV     #304,-(R2)      ;MOVE TO MAILBOX #  *******  304  *******
012126  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012130  000000                  HALT                    ;CC'S INCORRECT
012132  005300          MDM2B:  DEC     R0
012134  005300                  DEC     R0
012136  001404                  BEQ     MDM2D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
012140                  MDM2C:
012140  012742  000305          MOV     #305,-(R2)      ;MOVE TO MAILBOX #  *******  305  *******
012144  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012146  000000                  HALT                    ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
012150  005737  000000  MDM2D:  TST     @#0
012154  001404                  BEQ     TST146
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 753           <====
012156  012742  000306          MOV     #306,-(R2)      ;MOVE TO MAILBOX #  *******  306  *******
012162  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012164  000000                  HALT                    ;DESTINATION DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
                        ;INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
                        ;
                        ;*****************************************************************************
                        ;TEST 146       TEST MOV-BYTE DESTINATION MODE 2
                        ;*****************************************************************************
012166  005212          TST146: INC     (R2)            ;UPDATE TEST NUMBER
012170  022712  000146          CMP     #146,(R2)       ;SEQUENCE ERROR?
012174  001046                  BNE     TST147-10       ;BR TO ERROR HALT ON SEQ ERROR
012176  005000                  CLR     R0              ;R0=0
012200  005010                  CLR     (R0)            ;LOC. 0=0
012202  112720  000125          MOVB    #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
012206  102402                  BVS     MBDM2A          ;BR TO ERROR IF V SET
012210  001401                  BEQ     MBDM2A          ;BR TO ERROR IF Z SET
012212  100004                  BPL     MBDM2B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
012214                  MBDM2A:
012214  012742  000307          MOV     #307,-(R2)      ;MOVE TO MAILBOX #  *******  307  *******
012220  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012222  000000                  HALT                    ;CC'S INCORRECT
012224  022700  000001  MBDM2B: CMP     #1,R0
```

```
4038  012230  001404                        BEQ     MBDM2C
4039                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4040                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4041                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4042                                                                ;           WHICH FOLLOWS W/ 762           <====
4043  012232  012742  000310                  MOV     #310,-(R2)    ;MOVE TO MAILBOX #  *******  310  *******
4044  012236  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4045  012240  000000                        HALT                  ;REGISTER NOT INCREMENTED BY ONE
4046  012242  112720  000252          MBDM2C: MOVB    #252,(R0)+    ;TRY DESTINATION MODE 2 W/ODD BYTE
4047  012246  102402                        BVS     MBDM2D
4048  012250  001401                        BEQ     MBDM2D
4049  012252  100404                        BMI     MBDM2E
4050                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4051                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4052                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4053                                                                ;           WHICH FOLLOWS W/ 751           <====
4054  012254                          MBDM2D:
4055  012254  012742  000311                  MOV     #311,-(R2)    ;MOVE TO MAILBOX #  *******  311  *******
4056  012260  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4057  012262  000000                        HALT                  ;CC'S NOT SET CORRECT
4058  012264  022700  000002          MBDM2E: CMP     #2,R0
4059  012270  001404                        BEQ     MBDM2F
4060                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4061                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4062                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4063                                                                ;           WHICH FOLLOWS W/ 742           <====
4064  012272  012742  000312                  MOV     #312,-(R2)    ;MOVE TO MAILBOX #  *******  312  *******
4065  012276  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4066  012300  000000                        HALT                  ;REGISTER NOT INCREMENTED BY ONE
4067  012302  022737  125125  000000  MBDM2F: CMP     #125125,@#0   ;CHECK DATA
4068  012310  001404                        BEQ     TST147
4069                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4070                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4071                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4072                                                                ;           WHICH FOLLOWS W/ 732           <====
4073  012312  012742  000313                  MOV     #313,-(R2)    ;MOVE TO MAILBOX #  *******  313  *******
4074  012316  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4075  012320  000000                        HALT                  ;DESTINATION DATA INCORRECT
4076                                                                ;  OR SEQUENCE ERROR
4077
4078                                ;*************************************************************************
4079                                ;
4080                                ;       THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
4081                                ;AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB
4082                                ;INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
4083                                ;*************************************************************************
4084                                ;TEST 147       TEST MOV(B) DESTINATION MODE 3
4085                                ;*************************************************************************
4086  012322  005212                TST147: INC     (R2)          ;UPDATE TEST NUMBER
4087  012324  022712  000147                CMP     #147,(R2)     ;SEQUENCE ERROR?
4088  012330  001057                        BNE     TST150-10     ;BR TO ERROR HALT ON SEQ ERROR
4089  012332  012700  000400                MOV     #400,R0       ;R0=400
4090  012336  005010                        CLR     (R0)          ;LOC. 400 POINTS TO LOC. 0
4091  012340  005037  000000                CLR     @#0           ;LOC. 0=0
4092  012344  012730  125252                MOV     #125252,@(R0)+ ;TRY MOV DESTINATION MODE 2
4093  012350  102402                        BVS     MDM3A         ;BR TO ERROR IF V SET
```



```
4094  012352  001401                        BEQ     MDM3A         ;BR TO ERROR IF Z SET
4095  012354  100404                        BMI     MDM3B
4096                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4097                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4098                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4099                                                                ;           WHICH FOLLOWS W/ 766           <====
4100  012356                          MDM3A:
4101  012356  012742  000314                  MOV     #314,-(R2)    ;MOVE TO MAILBOX #  *******  314  *******
4102  012362  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4103  012364  000000                        HALT                  ;CC'S INCORRECT
4104  012366  022700  000402          MDM3B: CMP     #402,R0       ;CHECK DEST. MODE REGISTER
4105  012372  001404                        BEQ     MDM3C
4106                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4107                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4108                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4109                                                                ;           WHICH FOLLOWS W/ 757           <====
4110  012374  012742  000315                  MOV     #315,-(R2)    ;MOVE TO MAILBOX #  *******  315  *******
4111  012400  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4112  012402  000000                        HALT                  ;REGISTER NOT INCREMENTED BY 2
4113  012404  022737  125252  000000  MDM3C: CMP     #125252,@#0   ;CHECK DESTINATION DATA
4114  012412  001404                        BEQ     MDM3D
4115                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4116                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4117                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4118                                                                ;           WHICH FOLLOWS W/ 747           <====
4119  012414  012742  000316                  MOV     #316,-(R2)    ;MOVE TO MAILBOX #  *******  316  *******
4120  012420  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4121  012422  000000                        HALT                  ;DESTINATION DATA INCORRECT
4122  012424  112737  000125  000000  MDM3D: MOVB    #125,@#0      ;TRY MOVB DESTINATION MODE Z EVEN BYTE
4123  012432  022737  125125  000000          CMP     #125125,@#0   ;CHECK DATA
4124  012440  001404                        BEQ     MDM3E
4125                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4126                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4127                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4128                                                                ;           WHICH FOLLOWS W/ 734           <====
4129  012442  012742  000317                  MOV     #317,-(R2)    ;MOVE TO MAILBOX #  *******  317  *******
4130  012446  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4131  012450  000000                        HALT                  ;DESTINATION DATA INCORRECT
4132  012452  112737  000525  000001  MDM3E: MOVB    #525,@#1      ;TRY MOVB DESTINATION MODE 2 ODD BYTE
4133  012460  022737  052525  000000          CMP     #52525,@#0    ;CHECK DATA
4134  012466  001404                        BEQ     TST150
4135                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4136                                                                ;           CONDITIONAL BRANCH INST. AND   <====
4137                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
4138                                                                ;           WHICH FOLLOWS W/ 721           <====
4139  012470  012742  000320                  MOV     #320,-(R2)    ;MOVE TO MAILBOX #  *******  320  *******
4140  012474  005242                        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4141  012476  000000                        HALT                  ;
4142
4143                                ;*************************************************************************
4144                                ;
4145                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4146                                ;SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4147                                ;R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4148                                ;CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
4149                                ;
```

```
                                ;*****************************************************************************
                                ;TEST 150        TEST MOV DESTINATION MODE 4
                                ;*****************************************************************************
012500  005212          TST150: INC     (R2)            ;UPDATE TEST NUMBER
012502  022712  000150          CMP     #150,(R2)       ;SEQUENCE ERROR?
012506  001026                  BNE     TST151-10       ;BR TO ERROR HALT ON SEQ ERROR
012510  005000                  CLR     R0              ;R0=0
012512  005010                  CLR     (R0)            ;LOC 0=0
012514  012704  000002          MOV     #2,R4           ;R4=2
012520  012744  012345          MOV     #12345,-(R4)    ;TRY MOV DEST. MODE 4
012524  102402                  BVS     MDM4A           ;BR TO ERROR IF V-BIT SET
012526  001401                  BEQ     MDM4A           ;BR TO ERROR IF Z-BIT SET
012530  100004                  BPL     MDM4B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
012532                  MDM4A:
012532  012742  000321          MOV     #321,-(R2)      ;MOVE TO MAILBOX #  *******  321  *******
012536  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012540  000000                  HALT                    ;CC'S NOT CORRECT
012542  005704          MDM4B:  TST     R4              ;CHECK DECREMENTING OF MODE 4 REG.
012544  001404                  BEQ     MDM4C
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 761           <====
012546  012742  000322          MOV     #322,-(R2)      ;MOVE TO MAILBOX #  *******  322  *******
012552  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012554  000000                  HALT                    ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
012556  022710  012345  MDM4C:  CMP     #12345,(R0)     ;CHECK DESTINATION DATA
012562  001404                  BEQ     TST151
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
012564  012742  000323          MOV     #323,-(R2)      ;MOVE TO MAILBOX #  *******  323  *******
012570  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012572  000000                  HALT                    ;DESTINATION DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOVB DESTINATION MODE 4 INSTRUCTION
                                ;       ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
                                ;       USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
                                ;       ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH
                                ;       INSTRUCTIONS ARE USED TO VERIFY THE DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 151        TEST MOVB DESTINATION MODE 4
                                ;*****************************************************************************
012574  005212          TST151: INC     (R2)            ;UPDATE TEST NUMBER
012576  022712  000151          CMP     #151,(R2)       ;SEQUENCE ERROR?
012602  001046                  BNE     TST152-10       ;BR TO ERROR HALT ON SEQ ERROR
012604  005004                  CLR     R4              ;R4=0
```



```
012606  005014                  CLR     (R4)            ;LOC. 0=0
012610  012700  000002          MOV     #2,R0           ;R0 = 2
012614  112740  125125          MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4-ODD BYTE
012620  020027  000001          CMP     R0,#1           ;CHECK THAT DEST. REG. WAS DECREMENTED
012624  001404                  BEQ     MBDM4A
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
012626  012742  000324          MOV     #324,-(R2)      ;MOVE TO MAILBOX #  *******  324  *******
012632  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012634  000000                  HALT                    ;DESTINATION REG. NOT DECREMENTED BY 1
012636  021427  052400  MBDM4A: CMP     (R4),#52400     ;CHECK DEST. DATA
012642  001404                  BEQ     MBDM4B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
012644  012742  000325          MOV     #325,-(R2)      ;MOVE TO MAILBOX #  *******  325  *******
012650  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012652  000000                  HALT                    ;DEST. DATA NOT CORRECT
012654  112740  125125  MBDM4B: MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4--EVEN BYTE
012660  102402                  BVS     MBDM4C          ;BR. TO ERROR IF V-BIT SET
012662  001401                  BEQ     MBDM4C          ;BR TO ERROR IF Z-BIT SET
012664  100004                  BPL     MBDM4D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 747           <====
012666                  MBDM4C:
012666  012742  000326          MOV     #326,-(R2)      ;MOVE TO MAILBOX #  *******  326  *******
012672  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012674  000000                  HALT                    ;COND. CODES INCORRECT
012676  005700  MBDM4D: TST     R0                      ;CHECK MODE 4 DEST. REGISTER
012700  001404                  BEQ     MBDM4E
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 741           <====
012702  012742  000327          MOV     #327,-(R2)      ;MOVE TO MAILBOX #  *******  327  *******
012706  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012710  000000                  HALT                    ;DESTINATION REG NOT DECREMENTED BY 1
012712  021427  052525  MBDM4E: CMP     (R4),#52525     ;CHECK DEST. DAT^A
012716  001404                  BEQ     TST152
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 732           <====
012720  012742  000330          MOV     #330,-(R2)      ;MOVE TO MAILBOX #  *******  330  *******
012724  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012726  000000                  HALT                    ;DESTINATION DATA INCORRECT
                                                        ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOVB
```

```
                                ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
                                ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
                                ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
                                ;THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
                                ;PROPER ADDRESSING AND DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 152        TEST MOV DESTINATION MODE 5
                                ;*****************************************************************************
012730  005212                  TST152: INC     (R2)            ;UPDATE TEST NUMBER
012732  022712  000152                  CMP     #152,(R2)       ;SEQUENCE ERROR?
012736  001051                          BNE     TST153-10       ;BR TO ERROR HALT ON SEQ ERROR
012740  005004                          CLR     R4              ;R4=0
012742  005014                          CLR     (R4)            ;LOC. 0 = 0
012744  012700  000400                  MOV     #400,R0         ;R0=400
012750  012750  004321                  MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
012754  102402                          BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
012756  001401                          BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
012760  100004                          BPL     MDM5B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
012762                          MDM5A:
012762  012742  000331                  MOV     #331,-(R2)      ;MOVE TO MAILBOX #  *******  331  *******
012766  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
012770  000000                          HALT                    ;COND. CODES INCORRECT
012772  022700  000376          MDM5B:  CMP     #376,R0         ;CHECK MODE 5 REG. WAS DECREMENTED
012776  001404                          BEQ     MDM5C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
013000  012742  000332                  MOV     #332,-(R2)      ;MOVE TO MAILBOX #  *******  332  *******
013004  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013006  000000                          HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
013010  022714  004321          MDM5C:  CMP     #4321,(R4)      ;CHECK DEST. DATA
013014  001404                          BEQ     MDM5D
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 751           <====
013016  012742  000333                  MOV     #333,-(R2)      ;MOVE TO MAILBOX #  *******  333  *******
013022  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013024  000000                          HALT                    ;DEST. DATA INCORRECT
013026  012700  000406          MDM5D:  MOV     #406,R0         ;R0=406
013032  112750  000377                  MOVB    #377,@-(R0)     ;TRY MOV DEST. MODE 5 --EVEN BYTE
013036  022700  000404                  CMP     #404,R0         ;CHECK MODE 5 REG.
013042  001404                          BEQ     MDM5E
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 736           <====
013044  012742  000334                  MOV     #334,-(R2)      ;MOVE TO MAILBOX #  *******  334  *******
013050  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013052  000000                          HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
```



```
013054  022714  177721          MDM5E:  CMP     #177721,(R4)    ;CHECK DEST. DATA
013060  001404                          BEQ     TST153
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 727           <====
013062  012742  000335                  MOV     #335,-(R2)      ;MOVE TO MAILBOX #  *******  335  *******
013066  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013070  000000                          HALT                    ;DEST. DATA INCORRECT
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE
                                ;DESTINATION MODE 6 INDTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0
                                ;FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0
                                ;BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY
                                ;PROPER ADDRESSING AND DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 153        TEST MOV DESTINATION MODE 6
                                ;*****************************************************************************
013072  005212                  TST153: INC     (R2)            ;UPDATE TEST NUMBER
013074  022712  000153                  CMP     #153,(R2)       ;SEQUENCE ERROR?
013100  001054                          BNE     TST154-10       ;BR TO ERROR HALT ON SEQ ERROR
013102  005000                          CLR     R0              ;R0=0
013104  005010                          CLR     (R0)            ;LOC. 0=0
013106  005200                          INC     R0              ;R0=1
013110  012760  052525  177777          MOV     #052525,-1(R0)  ;TRY MOV DEST. MODE 6
013116  102402                          BVS     MDM6A           ;BR TO ERROR IF V-BIT SET
013120  001401                          BEQ     MDM6A           ;BR TO ERROR IF Z-BIT SET
013122  100004                          BPL     MDM6B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
013124                          MDM6A:
013124  012742  000336                  MOV     #336,-(R2)      ;MOVE TO MAILBOX #  *******  336  *******
013130  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013132  000000                          HALT                    ;COND. CODES INCORRECT
013134  022700  000001          MDM6B:  CMP     #1,R0           ;CHECK DEST. REGISTER UNALTERED
013140  001404                          BEQ     MDM6C
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 760           <====
013142  012742  000337                  MOV     #337,-(R2)      ;MOVE TO MAILBOX #  *******  337  *******
013146  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
013150  000000                          HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
013152  022737  052525  000000  MDM6C:  CMP     #52525,@#0      ;CHECK DEST. DATA
013160  001404                          BEQ     MDM6D
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 750           <====
013162  012742  000340                  MOV     #340,-(R2)      ;MOVE TO MAILBOX #  *******  340  *******
```

```
 4374  013166  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4375  013170  000000                    HALT                    ;DEST. DATA INCORRECT
 4376  013172  012700  000002  MDM6D:    MOV     #2,R0           ;R0=2
 4377  013176  112760  000377  177777    MOVB    #377,-1(R0)     ;TRY MOVB DEST. MODE 6
 4378  013204  022700  000002            CMP     #2,R0           ;CHECK DEST. REGISTER UNALTERED
 4379  013210  001404                    BEQ     MDM6E
 4380                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4381                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4382                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4383                                                            ;           WHICH FOLLOWS W/ 734           <====
 4384  013212  012742  000341            MOV     #341,-(R2)      ;MOVE TO MAILBOX #  *******  341  *******
 4385  013216  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4386  013220  000000                    HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
 4387  013222  022737  177525  000000  MDM6E:  CMP     #177525,@#0     ;CHECK DEST. DATA
 4388  013230  001404                    BEQ     TST154
 4389                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4390                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4391                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4392                                                            ;           WHICH FOLLOWS W/ 724           <====
 4393  013232  012742  000342            MOV     #342,-(R2)      ;MOVE TO MAILBOX #  *******  342  *******
 4394  013236  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4395  013240  000000                    HALT                    ;DEST. DATA INCORRECT
 4396                                                            ;  OR SEQUENCE ERROR
 4397
 4398                            ;*****************************************************************************
 4399                            ;
 4400                            ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
 4401                            ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
 4402                            ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
 4403                            ;USED TO VERIFY PROPER ADDRESSING AND DATA.
 4404                            ;
 4405                            ;*****************************************************************************
 4406                            ;TEST 154       TEST MOV DESTINATION MODE 7
 4407                            ;*****************************************************************************
 4408  013242  005212  TST154:  INC     (R2)            ;UPDATE TEST NUMBER
 4409  013244  022712  000154            CMP     #154,(R2)       ;SEQUENCE ERROR?
 4410  013250  001053                    BNE     TST155-10       ;BR TO ERROR HALT ON SEQ ERROR
 4411  013252  005004                    CLR     R4              ;R4=0
 4412  013254  005014                    CLR     (R4)            ;LOC.0=0
 4413  013256  012700  000403            MOV     #403,R0         ;R0=403
 4414  013262  012770  070707  177777    MOV     #70707,@-1(R0)  ;TRY MOV W/DEST MODE 7
 4415  013270  102402                    BVS     MDM7A           ;BR. TO ERROR IF V-BIT SET
 4416  013272  001401                    BEQ     MDM7A           ;BR TO ERROR IF Z-BIT SET
 4417  013274  100004                    BPL     MDM7B
 4418                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4419                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4420                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4421                                                            ;           WHICH FOLLOWS W/ 766           <====
 4422  013276            MDM7A:
 4423  013276  012742  000343            MOV     #343,-(R2)      ;MOVE TO MAILBOX #  *******  343  *******
 4424  013302  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4425  013304  000000                    HALT                    ;COND. CODES INCORRECT
 4426  013306  022700  000403  MDM7B:    CMP     #403,R0         ;CHECK DEST. REGISTER
 4427  013312  001404                    BEQ     MDM7C
 4428                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4429                                                            ;           CONDITIONAL BRANCH INST. AND   <====
```



```
 4430                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4431                                                            ;           WHICH FOLLOWS W/ 757           <====
 4432  013314  012742  000344            MOV     #344,-(R2)      ;MOVE TO MAILBOX #  *******  344  *******
 4433  013320  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4434  013322  000000                    HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
 4435  013324  022737  070707  000000  MDM7C:  CMP     #70707,@#0      ;CHECK DEST. DATA
 4436  013332  001404                    BEQ     MDM7D
 4437                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4438                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4439                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4440                                                            ;           WHICH FOLLOWS W/ 747           <====
 4441  013334  012742  000345            MOV     #345,-(R2)      ;MOVE TO MAILBOX #  *******  345  *******
 4442  013340  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4443  013342  000000                    HALT                    ;DEST. DATA INCORRECT
 4444  013344  112770  107070  000001  MDM7D:  MOVB    #107070,@1(R0)  ;TRY MOVB W/DEST MODE 7--ODD BYTE
 4445  013352  022700  000403            CMP     #403,R0         ;CHECK MODE 7 DEST. REG.
 4446  013356  001404                    BEQ     MDM7E
 4447                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4448                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4449                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4450                                                            ;           WHICH FOLLOWS W/ 735           <====
 4451  013360  012742  000346            MOV     #346,-(R2)      ;MOVE TO MAILBOX #  *******  346  *******
 4452  013364  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4453  013366  000000                    HALT                    ;DEST. DATA INCORRECT
 4454  013370  022737  034307  000000  MDM7E:  CMP     #34307,@#0      ;CHECK DEST. DATA
 4455  013376  001404                    BEQ     TST155
 4456                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4457                                                            ;           CONDITIONAL BRANCH INST. AND   <====
 4458                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
 4459                                                            ;           WHICH FOLLOWS W/ 725           <====
 4460  013400  012742  000347            MOV     #347,-(R2)      ;MOVE TO MAILBOX #  *******  347  *******
 4461  013404  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 4462  013406  000000                    HALT                    ;DESTINATION DATA INCORRECT
 4463                                                            ;  OR SEQUENCE ERROR
 4464
 4465                            ;*****************************************************************************
 4466                            ;
 4467                            ;       THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
 4468                            ;THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
 4469                            ;TABLE OF OPERANDS.  THE TABLE OF OPERANDS AND THE WORK LOCATION IS
 4470                            ;STORED FOLLOWING THE TEST CODE.  A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
 4471                            ;THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
 4472                            ;VALUE.  THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
 4473                            ;GO UNDETECTED.  WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
 4474                            ;ODD ADDRESSES ARE USED IN THE TEST.  THE LISTING SHOWS THE
 4475                            ;EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
 4476                            ;
 4477                            ;*****************************************************************************
 4478                            ;TEST 155       TEST MODE 4 W/ DOP INSTS.
 4479                            ;*****************************************************************************
 4480  013410  005212  TST155:  INC     (R2)            ;UPDATE TEST NUMBER
 4481  013412  022712  000155            CMP     #155,(R2)       ;SEQUENCE ERROR?
 4482  013416  001015                    BNE     DOP4            ;BR TO ERROR HALT ON SEQ ERROR
 4483  013420  012700  013472            MOV     #TBL1,R0        ;INITIALIZE R0
 4484  013424  014037  013472            MOV     -(R0),@#TBL1    ;TBL1=125252
 4485  013430  064037  013472            ADD     -(R0),@#TBL1    ;TBL1=000377
```

```
4486  013434  144037  013472          BICB    -(R0),@#TBL1    ;TBL1=000252
4487  013440  154037  013473          BISB    -(R0),@#TBL1+1  ;TBL1=125252
4488  013444  024037  013472          CMP     -(R0),@#TBL1    ;CHECK RESULT
4489  013450  001411                  BEQ     TST156
4490                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4491                                                          ;           CONDITIONAL BRANCH INST. AND    <====
4492                                                          ;           REPLACE THE MOVE INSTRUCTION    <====
4493                                                          ;           WHICH FOLLOWS W/ 763            <====
4494  013452                  DOP4:
4495  013452  012742  000350          MOV     #350,-(R2)      ;MOVE TO MAILBOX #  *******  350  *******
4496  013456  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4497  013460  000000                  HALT                    ;RESULT OF MODE 4 INSTS. INCORRECT
4498                                                          ;  OR SEQUENCE ERROR
4499
4500  013462  125252                  125252
4501  013464  052652                  52652
4502  013466  053125                  53125
4503  013470  125252                  125252
4504  013472  000000          TBL1:   0
4505
4506                          ;*****************************************************************************
4507                          ;
4508                          ;       THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
4509                          ;THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
4510                          ;THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
4511                          ;THE DATA TABLE USED IN THE PREVIOUS TEST.  THE TEST IS IDENTICAL TO
4512                          ;THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
4513                          ;TABLE AND MODE 5 ADDRESSING.  (SEE PREVIOUS TEST).
4514                          ;
4515                          ;*****************************************************************************
4516                          ;TEST 156       TEST MODE 5 W/ DOP INSTS.
4517                          ;*****************************************************************************
4518  013474  005212          TST156: INC     (R2)            ;UPDATE TEST NUMBER
4519  013476  022712  000156          CMP     #156,(R2)       ;SEQUENCE ERROR?
4520  013502  001015                  BNE     DOP5            ;BR TO ERROR HALT ON SEQ ERROR
4521  013504  012700  013560          MOV     #TBL2+2,R0      ;INITIALIZE R0
4522  013510  015037  013472          MOV     @-(R0),@#TBL1   ;TBL1=125252
4523  013514  065037  013472          ADD     @-(R0),@#TBL1   ;TBL1=000377
4524  013520  145037  013472          BICB    @-(R0),@#TBL1   ;TBL1=000252
4525  013524  155037  013473          BISB    @-(R0),@#TBL1+1 ;TBL1=125252
4526  013530  025037  013472          CMP     @-(R0),@#TBL1   ;CHECK RESULT
4527  013534  001411                  BEQ     TST157
4528                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4529                                                          ;           CONDITIONAL BRANCH INST. AND    <====
4530                                                          ;           REPLACE THE MOVE INSTRUCTION    <====
4531                                                          ;           WHICH FOLLOWS W/ 763            <====
4532  013536                  DOP5:
4533  013536  012742  000351          MOV     #351,-(R2)      ;MOVE TO MAILBOX #  *******  351  *******
4534  013542  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4535  013544  000000                  HALT                    ;RESULT OF MODE 5 INSTS. INCORRECT
4536                                                          ;  OR SEQUENCE ERROR
4537  013546  013462                  TBL1-10
4538  013550  013464                  TBL1-6
4539  013552  013465                  TBL1-5
4540  013554  013466                  TBL1-4
4541  013556  013470          TBL2:   TBL1-2
```



```
4542                          ;*****************************************************************************
4543                          ;
4544                          ;       THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
4545                          ;IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
4546                          ;THIS TIME THE DATA IS ACCESSED USING MODE 6.  R0 IS SET
4547                          ;TO POINT TO THE MIDDLE OF THE TABLE.  THE TABLE IS ACCESSED FROM
4548                          ;BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
4549                          ;THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
4550                          ;TESTS.
4551                          ;
4552                          ;*****************************************************************************
4553                          ;TEST 157       TEST MODE 6 W/ DOP INSTS.
4554                          ;*****************************************************************************
4555  013560  005212          TST157: INC     (R2)            ;UPDATE TEST NUMBER
4556  013562  022712  000157          CMP     #157,(R2)       ;SEQUENCE ERROR?
4557  013566  001022                  BNE     TST160-10       ;BR TO ERROR HALT ON SEQ ERROR
4558  013570  012700  013466          MOV     #TBL1-4,R0      ;INITIALIZE R0
4559  013574  016037  000002  013472  MOV     2(R0),@#TBL1    ;TBL1=125252
4560  013602  066037  000000  013472  ADD     0(R0),@#TBL1    ;TBL1=000377
4561  013610  146037  177777  013472  BICB    -1(R0),@#TBL1   ;TBL1=000252
4562  013616  156037  177776  013473  BISB    -2(R0),@#TBL1+1 ;TBL1=125252
4563  013624  026037  177774  013472  CMP     -4(R0),@#TBL1   ;CHECK RESULT
4564  013632  001404                  BEQ     TST160
4565                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4566                                                          ;           CONDITIONAL BRANCH INST. AND    <====
4567                                                          ;           REPLACE THE MOVE INSTRUCTION    <====
4568                                                          ;           WHICH FOLLOWS W/ 756            <====
4569  013634  012742  000352          MOV     #352,-(R2)      ;MOVE TO MAILBOX #  *******  352  *******
4570  013640  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4571  013642  000000                  HALT                    ;RESULT OF MODE 6 INSTS. INCORRECT
4572                                                          ;  OR SEQUENCE ERROR
4573                          ;*****************************************************************************
4574                          ;
4575                          ;       THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
4576                          ;THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
4577                          ;THE MODE 5 TESTS.  THIS TIME THE DATA IS ACCESSED USING MODE 7.
4578                          ;R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
4579                          ;TEST.  THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
4580                          ;IN THE MODE 7 INSTRUCTIONS.  THE DATA RESULTS ARE IDENTICAL TO
4581                          ;THOSE EXPECTED IN THE MODE 5 TESTS.
4582                          ;
4583                          ;*****************************************************************************
4584                          ;TEST 160       TEST MODE 7 W/ DOP INSTS.
4585                          ;*****************************************************************************
4586  013644  005212          TST160: INC     (R2)            ;UPDATE TEST NUMBER
4587  013646  022712  000160          CMP     #160,(R2)       ;SEQUENCE ERROR?
4588  013652  001022                  BNE     TST161-10       ;BR TO ERROR HALT ON SEQ ERROR
4589  013654  012700  013552          MOV     #TBL2-4,R0      ;INITIALIZE R0
4590  013660  017037  000004  013472  MOV     @4(R0),@#TBL1   ;TBL1=125252
4591  013666  067037  000002  013472  ADD     @2(R0),@#TBL1   ;TBL1=000377
4592  013674  147037  000000  013472  BICB    @0(R0),@#TBL1   ;TBL1=000252
4593  013702  157037  177776  013473  BISB    @-2(R0),@#TBL1+1        ;TBL1=125252
4594  013710  027037  177774  013472  CMP     @-4(R0),@#TBL1          ;CHECK RESULT
4595  013716  001404                  BEQ     TST161
4596                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4597                                                          ;           CONDITIONAL BRANCH INST. AND    <====
```

```
4598                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4599                                                    ;           WHICH FOLLOWS W/ 756           <====
4600  013720  012742  000353          MOV     #353,-(R2)      ;MOVE TO MAILBOX #  *******  353  *******
4601  013724  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4602  013726  000000                  HALT                    ;RESULT OF MODE 7 INSTS INCORRECT
4603                                                    ;  OR SEQUENCE ERROR
4604                          ;*****************************************************************************
4605                          ;
4606                          ;       THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
4607                          ;R0 IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND
4608                          ;AN ROL INSTRUCTION IS EXECUTED WITH MODE 0.  THE OPERATION IS CHECKED
4609                          ;BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
4610                          ;NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
4611                          ;
4612                          ;*****************************************************************************
4613                          ;TEST 161       TEST ROTATE INSTRUCTIONS OF MODE 0
4614                          ;*****************************************************************************
4615  013730  005212          TST161: INC     (R2)            ;UPDATE TEST NUMBER
4616  013732  022712  000161          CMP     #161,(R2)       ;SEQUENCE ERROR?
4617  013736  001026                  BNE     TST162-10       ;BR TO ERROR HALT ON SEQ ERROR
4618  013740  012700  125252          MOV     #125252,R0      ;INITIALIZE DATA
4619  013744  000261                  SEC                     ;SET C-BIT
4620  013746  006100                  ROL     R0              ;TRY ROL W/ MODE 0
4621  013750  102004                  BVC     ROT0A           ;CC=0011
4622  013752  103003                  BCC     ROT0A
4623  013754  022700  052525          CMP     #052525,R0      ;CHECK DATA
4624  013760  001404                  BEQ     ROT0B
4625                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4626                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4627                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4628                                                    ;           WHICH FOLLOWS W/ 767           <====
4629  013762                  ROT0A:
4630  013762  012742  000354          .  MOV     #354,-(R2)      ;MOVE TO MAILBOX #  *******  354  *******
4631  013766  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4632  013770  000000                  HALT                    ;ROL MODE 0 FAILED
4633  013772  012700  125252  ROT0B: MOV     #125252,R0      ;INITIALIZE DATA
4634  013776  000261                  SEC                     ;SET C-BIT
4635  014000  106100                  ROLB    R0              ;TRY ROL W/ MODE 0 EVEN BYTE
4636  014002  102004                  BVC     ROT0C           ;CC=0011
4637  014004  103003                  BCC     ROT0C
4638  014006  022700  125125          CMP     #125125,R0      ;CHECK DATA
4639  014012  001404                  BEQ     TST162
4640                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4641                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4642                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4643                                                    ;           WHICH FOLLOWS W/ 752           <====
4644  014014                  ROT0C:
4645  014014  012742  000355          MOV     #355,-(R2)      ;MOVE TO MAILBOX #  *******  355  *******
4646  014020  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4647  014022  000000                  HALT                    ;ROLB MODE 0 FAILED
4648                                                    ;  OR SEQUENCE ERROR
```



```
4649
4650                          ;*****************************************************************************
4651                          ;
4652                          ;       THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
4653                          ;THE DATA TO BE ROTATED IS IN LOC 0.  R0 IS USED AS THE
4654                          ;ADDRESSING REGISTER.  THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
4655                          ;THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
4656                          ;THE C AND V BITS.  THIS PROCEDURE IS THEN REPEATED TWICE MORE
4657                          ;TO TEST THE BYTE ROTATES.  FIRST ON BYTE 0, THEN ON BYTE 1.
4658                          ;
4659                          ;*****************************************************************************
4660                          ;TEST 162       TEST ROTATE INSTRUCTIONS W/ MODE 1
4661                          ;*****************************************************************************
4662  014024  005212          TST162: INC     (R2)            ;UPDATE TEST NUMBER
4663  014026  022712  000162          CMP     #162,(R2)       ;SEQUENCE ERROR?
4664  014032  001051                  BNE     TST163-10       ;BR TO ERROR HALT ON SEQ ERROR
4665  014034  005000                  CLR     R0              ;POINT TO LOC. 0
4666  014036  012710  052525          MOV     #52525,(R0)     ;INITIALIZE DATA
4667  014042  000241                  CLC                     ;CLEAR C-BIT
4668  014044  006110                  ROL     (R0)            ;TRY ROL W/ MODE 1
4669  014046  102005                  BVC     ROT1A           ;CC=1010
4670  014050  103404                  BCS     ROT1A
4671  014052  023727  000000  125252  CMP     @#0,#125252     ;CHECK RESULT
4672  014060  001404                  BEQ     ROT1B
4673                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4674                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4675                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4676                                                    ;           WHICH FOLLOWS W/ 765           <====
4677  014062                  ROT1A:
4678  014062  012742  000356          MOV     #356,-(R2)      ;MOVE TO MAILBOX #  *******  356  *******
4679  014066  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4680  014070  000000                  HALT                    ;ROL MODE 1 FAILED
4681  014072  000261          ROT1B:  SEC
4682  014074  012710  125252          MOV     #125252,(R0)    ;INITIALIZE DATA
4683  014100  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 EVEN BYTE
4684  014102  102005                  BVC     ROT1C           ;CC=1011
4685  014104  103004                  BCC     ROT1C
4686  014106  022737  125125  000000  CMP     #125125,@#0     ;TEST RESULT
4687  014114  001404                  BEQ     ROT1D
4688                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
4689                                                    ;           CONDITIONAL BRANCH INST. AND   <====
4690                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
4691                                                    ;           WHICH FOLLOWS W/ 747           <====
4692  014116                  ROT1C:
4693  014116  012742  000357          MOV     #357,-(R2)      ;MOVE TO MAILBOX #  *******  357  *******
4694  014122  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
4695  014124  000000                  HALT                    ;ROLB W/ MODE 1 EVEN BYTE FAILED
4696  014126  012710  125252  ROT1D: MOV     #125252,(R0)
4697  014132  005000                  CLR     R0              ;POINT TO ODD BYTE
4698  014134  005200                  INC     R0
4699  014136  000261                  SEC                     ;SET C-BIT
4700  014140  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 ODD BYTE
4701  014142  102005                  BVC     ROT1E           ;CC=0011
4702  014144  103004                  BCC     ROT1E
4703  014146  022737  052652  000000  CMP     #052652,@#0     ;CHECK DATA
4704  014154  001404                  BEQ     TST163
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 727           <====
014156                          ROT1E:
014156  012742  000360                MOV     #360,-(R2)      ;MOVE TO MAILBOX #  *******  360  *******
014162  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014164  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
                                ;*****************************************************************************
014166  005212                TST163: INC     (R2)            ;UPDATE TEST NUMBER
014170  022712  000163                CMP     #163,(R2)       ;SEQUENCE ERROR?
014174  001057                       BNE     TST164-10       ;BR TO ERROR HALT ON SEQ ERROR
014176  005000                       CLR     R0              ;POINT TO LOC 0
014200  012710  173737                MOV     #173737,(R0)    ;INITIALIZE DATA
014204  000241                       CLC                     ;CLEAR C-BIT
014206  006120                       ROL     (R0)+           ;TRY ROL W/ MODE 2
014210  103007                       BCC     ROT2A           ;CHECK C-BIT
014212  022737  167676  000000        CMP     #167676,@#0     ;CHECK DATA
014220  001003                       BNE     ROT2A           ;BRANCH IF RESULT INCORRECT
014222  005300                       DEC     R0              ;TEST R0
014224  005300                       DEC     R0
014226  001404                       BEQ     ROT2B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 763           <====
014230                          ROT2A:
014230  012742  000361                MOV     #361,-(R2)      ;MOVE TO MAILBOX #  *******  361  *******
014234  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014236  000000                       HALT                    ;ROL W/ MODE 2 FAILED
014240  005000                ROT2B: CLR     R0              ;POINT TO LOC 0
014242  012710  004040                MOV     #4040,(R0)      ;INITIALIZE DATA
014246  000241                       CLC                     ;CLEAR C-BIT
014250  106120                       ROLB    (R0)+           ;TRY ROLB W/ MODE 2 EVEN BYTE
014252  103406                       BCS     ROT2C           ;CHECK C-BIT
014254  022737  004100  000000        CMP     #4100,@#0       ;CHECK DATA
014262  001002                       BNE     ROT2C           ;BRANCH IF DATA INCORRECT
014264  005300                       DEC     R0              ;CHECK R0
014266  001404                       BEQ     ROT2D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 743           <====
014270                          ROT2C:
014270  012742  000362                MOV     #362,-(R2)      ;MOVE TO MAILBOX #  *******  362  *******
```



```
014274  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014276  000000                       HALT                    ;ROLB W/ MODE 2 EVEN BYTE FAILED
014300  005000                ROT2D: CLR     R0              ;POINT TO LOC 0
014302  012710  004040                MOV     #4040,(R0)      ;INITIALIZE DATA
014306  005200                       INC     R0              ;POINT TO ODD BYTE OF DATA
014310  000261                       SEC                     ;SET C-BIT
014312  106120                       ROLB    (R0)+           ;TRY ROL W/ MODE 2 ODD BYTE
014314  103407                       BCS     ROT2E           ;CHECK C-BIT
014316  022737  010440  000000        CMP     #10440,@#0      ;CHECK DATA
014324  001003                       BNE     ROT2E           ;BRANCH IF DATA INCORRECT
014326  005300                       DEC     R0              ;CHECK R0
014330  005300                       DEC     R0
014332  001404                       BEQ     TST164
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 721           <====
014334                          ROT2E:
014334  012742  000363                MOV     #363,-(R2)      ;MOVE TO MAILBOX #  *******  363  *******
014340  005242                       INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014342  000000                       HALT                    ;ROLB W/ MODE 2 ODD BYTE FAILED
                                                            ;  OR SEQUENCE ERROR
```

```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
                        ;THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
                        ;TESTS.  THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
                        ;MODE 37.  BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
                        ;
                        ;*****************************************************************************
                        ;TEST 164       TEST ROTATE INSTRUCTIONS /W MODE 3
                        ;*****************************************************************************
014344  005212         ;TST164: INC     (R2)            ;UPDATE TEST NUMBER
014346  022712  000164          CMP     #164,(R2)       ;SEQUENCE ERROR?
014352  001051                  BNE     TST165-10       ;BR TO ERROR HALT ON SEQ ERROR
014354  012737  052525  000000  MOV     #52525,@#0      ;INITIALIZE DATA IN LOC 0
014362  000261                  SEC                     ;SET C-BIT
014364  006137  000000          ROL     @#0             ;TRY ROL W/ MODE 3
014370  103404                  BCS     ROT3A           ;CHECK C-BIT
014372  022737  125253  000000  CMP     #125253,@#0     ;CHECK DATA
014400  001404                  BEQ     ROT3B
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 765           <====
014402                  ROT3A:
014402  012742  000364          MOV     #364,-(R2)      ;MOVE TO MAILBOX #  *******  364  *******
014406  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014410  000000                  HALT                    ;ROL W/ MODE 3 FAILED
014412  012737  125252  000000  ROT3B:  MOV     #125252,@#0     ;INITIALIZE DATA
014420  000241                  CLC                     ;CLEAR C-BIT
014422  106137  000000          ROLB    @#0             ;TRY ROL W/ MODE 3 EVEN BYTE
014426  103004                  BCC     ROT3C           ;CHECK C-BIT
014430  023727  000000  125124  4$:     CMP     @#0,#125124     ;CHECK DATA
014436  001404                  BEQ     ROT3D
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 746           <====
014440                  ROT3C:
014440  012742  000365          MOV     #365,-(R2)      ;MOVE TO MAILBOX #  *******  365  *******
014444  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014446  000000                  HALT                    ;ROL W/ MODE 3 EVEN BYTE FAILED
014450  012737  125252  000000  ROT3D:  MOV     #125252,@#0     ;INITIALIZE DATA IN LOC. 0
014456  000261                  SEC                     ;SET C-BIT
014460  106137  000001          ROLB    @#1             ;TRY ROL W/ MODE 3 ODD BYTE
014464  103004                  BCC     ROT3E           ;CHECK C-BIT
014466  022737  052652  000000  CMP     #052652,@#0     ;CHECK DATA
014474  001404                  BEQ     TST165
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 727           <====
014476                  ROT3E:
014476  012742  000366          MOV     #366,-(R2)      ;MOVE TO MAILBOX #  *******  366  *******
014502  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014504  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
                        ;*****************************************************************************
014506  005212         ;TST165: INC     (R2)            ;UPDATE TEST NUMBER
014510  022712  000165          CMP     #165,(R2)       ;SEQUENCE ERROR?
014514  001016                  BNE     TST166-10       ;BR TO ERROR HALT ON SEQ ERROR
014516  012737  070707  000000  MOV     #070707,@#0     ;INITIALIZE DATA IN LOC. 0
014524  012700  000002          MOV     #2,R0           ;INITIALIZE R0 AS POINTER
014530  000261                  SEC                     ;SET C-BIT
014532  006140                  ROL     -(R0)           ;TRY ROL W/ MODE 4
014534  103406                  BCS     ROT4            ;CHECK C-BIT
014536  022737  161617  000000  CMP     #161617,@#0     ;CHECK DATA
014544  001002                  BNE     ROT4            ;BRANCH IF DATA INCORRECT
014546  005700                  TST     R0              ;CHECK MODE 4 REGISTER
014550  001404                  BEQ     TST166
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
014552                  ROT4:
014552  012742  000367          MOV     #367,-(R2)      ;MOVE TO MAILBOX #  *******  367  *******
014556  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
014560  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
                        ;*****************************************************************************
014562  005212         ;TST166: INC     (R2)            ;UPDATE TEST NUMBER
014564  022712  000166          CMP     #166,(R2)       ;SEQUENCE ERROR?
014570  001021                  BNE     ROT5            ;BR TO ERROR HALT ON SEQ ERROR
014572  012737  014644  000000  MOV     #ROTX,@#0       ;MOVE POINTER TO LOC. 0
014600  012700  000002          MOV     #2,R0           ;SET MODE 5 REG. TO LOC. 0
014604  012767  107070  000032  MOV     #107070,ROTX    ;INITIALIZE DATA
014612  000241                  CLC                     ;CLEAR C-BIT
014614  006150                  ROL     @-(R0)          ;TRY ROL W/ MODE 5
```

```
4895  014616  103006                  BCC     ROT5                    ;CHECK C-BIT
4896  014620  022737  016160  014644  CMP     #016160,@#ROTX          ;CHECK DATA
4897  014626  001002                  BNE     ROT5                    ;BRANCH IF DATA INCORRECT
4898  014630  005700                  TST     R0                      ;CHECK MODE 5 REGISTER
4899  014632  001405                  BEQ     TST167
4900                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4901                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
4902                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
4903                                                                  ;           WHICH FOLLOWS W/ 757            <====
4904  014634                          ROT5:
4905  014634  012742  000370                  MOV     #370,-(R2)      ;MOVE TO MAILBOX #  *******  370  *******
4906  014640  005242                  INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4907  014642  000000                  HALT                            ;ROL MODE 5 FAILED
4908                                                                  ;  OR SEQUENCE ERROR
4909  014644  000000                  ROTX:   0
4910
4911                                  ;*****************************************************************************************
4912                                  ;
4913                                  ;       THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.
4914                                  ;IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE
4915                                  ;ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.
4916                                  ;THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).
4917                                  ;
4918                                  ;*****************************************************************************************
4919                                  ;TEST 167       TEST MODE 6 W/ ROTATE INSTRUCTIONS
4920                                  ;*****************************************************************************************
4921  014646  005212                  TST167: INC     (R2)                    ;UPDATE TEST NUMBER
4922  014650  022712  000167                  CMP     #167,(R2)               ;SEQUENCE ERROR?
4923  014654  001013                  BNE     TST170-10               ;BR TO ERROR HALT ON SEQ ERROR
4924  014656  012737  125252  014644          MOV     #125252,@#ROTX          ;INITIALIZE DATA
4925  014664  000261                  SEC                             ;SET C-BIT
4926  014666  006167  177752                  ROL     ROTX                    ;TRY ROL W/ MODE 6
4927  014672  103004                  BCC     ROT6                    ;CHECK C-BIT
4928  014674  022737  052525  014644          CMP     #52525,@#ROTX           ;CHECK DATA
4929  014702  001404                  BEQ     TST170
4930                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4931                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
4932                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
4933                                                                  ;           WHICH FOLLOWS W/ 765            <====
4934  014704                          ROT6:
4935  014704  012742  000371                  MOV     #371,-(R2)      ;MOVE TO MAILBOX #  *******  371  *******
4936  014710  005242                  INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4937  014712  000000                  HALT                            ;ROL W/ MODE 6 FAILED
4938                                                                  ;  OR SEQUENCE ERROR
```



```
4939
4940                                  ;*****************************************************************************************
4941                                  ;
4942                                  ;THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
4943                                  ;THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST).  THE ROL INSTRUCTION
4944                                  ;ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
4945                                  ;(ROTXAD) FOLLOWING THE TEST CODE.
4946                                  ;
4947                                  ;*****************************************************************************************
4948                                  ;TEST 170       TEST MODE 7 W/ ROTATE INSTRUCTIONS
4949                                  ;*****************************************************************************************
4950  014714  005212                  TST170: INC     (R2)                    ;UPDATE TEST NUMBER
4951  014716  022712  000170                  CMP     #170,(R2)               ;SEQUENCE ERROR?
4952  014722  001016                  BNE     ROT7                    ;BR TO ERROR HALT ON SEQ ERROR
4953  014724  012737  052525  014644          MOV     #52525,@#ROTX           ;INITIALIZE DATA
4954  014732  012737  014644  014770          MOV     #ROTX,@#ROTXAD          ;INITIALIZE ADDRESS POINTER
4955  014740  000241                  CLC                             ;CLEAR C-BIT
4956  014742  006177  000022                  ROL     @ROTXAD                 ;TRY ROL W/ MODE 7
4957  014746  103404                  BCS     ROT7                    ;CHECK C-BIT
4958  014750  023727  014644  125252          CMP     @#ROTX,#125252          ;CHECK DATA
4959  014756  001405                  BEQ     TST171
4960                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4961                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
4962                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
4963                                                                  ;           WHICH FOLLOWS W/ 762            <====
4964  014760                          ROT7:
4965  014760  012742  000372                  MOV     #372,-(R2)      ;MOVE TO MAILBOX #  *******  372  *******
4966  014764  005242                  INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4967  014766  000000                  HALT                            ;ROL W/ MODE 7 FAILED
4968                                                                  ;  OR SEQUENCE ERROR
4969  014770  000000                  ROTXAD: 0
4970
4971
4972                                  ;*****************************************************************************************
4973                                  ;
4974                                  ;       THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION.  R0 IS SET TO
4975                                  ;177400.  A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH
4976                                  ;IS USED TO CHECK THE SIGN OF THE RESULT.  ALSO, A COMPARISON
4977                                  ;IS MADE TO CHECK THE DATA RESULTS.
4978                                  ;
4979                                  ;*****************************************************************************************
4980                                  ;TEST 171       TEST MODE 0 W/ SWAB INST.
4981                                  ;*****************************************************************************************
4982  014772  005212                  TST171: INC     (R2)                    ;UPDATE TEST NUMBER
4983  014774  022712  000171                  CMP     #171,(R2)               ;SEQUENCE ERROR?
4984  015000  001013                  BNE     TST172-10               ;BR TO ERROR HALT ON SEQ ERROR
4985  015002  012700  177400          MOV     #177400,R0              ;MOVE TEST PATTERN TO R0
4986  015006  000300                  SWAB    R0                      ;TRY SWAB MODE 0
4987  015010  100404                  BMI     SB0
4988                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
4989                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
4990                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
4991                                                                  ;           WHICH FOLLOWS W/ 774            <====
4992  015012  012742  000373                  MOV     #373,-(R2)      ;MOVE TO MAILBOX #  *******  373  *******
4993  015016  005242                  INC     -(R2)                   ;SET MSGTYP TO FATAL ERROR
4994  015020  000000                  HALT                            ;SWAB DID NOT SET CC'S CORRECT
```

```
 4995  015022  022700  000377          SB0:    CMP     #377,R0         ;CHECK RESULT
 4996  015026  001404                          BEQ     TST172
 4997                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 4998                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 4999                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 5000                                                                    ;           WHICH FOLLOWS W/ 765           <====
 5001  015030  012742  000374                  MOV     #374,-(R2)      ;MOVE TO MAILBOX #  *******  374  *******
 5002  015034  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5003  015036  000000                          HALT                    ;RESULT OF SWAB MODE 0 FAILED
 5004                                                                    ;  OR SEQUENCE ERROR
 5005
 5006                                  ;*****************************************************************************
 5007                                  ;
 5008                                  ;       THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION.  THE TEST
 5009                                  ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE ADDRESSING
 5010                                  ;REGISTER IN THE MODE 1 SWAB.  THE DATA RESULTS ARE CHECKED WITH
 5011                                  ;A COMPARE.
 5012                                  ;
 5013                                  ;*****************************************************************************
 5014                                  ;TEST 172       TEST MODE 1 W/ SWAB INST
 5015                                  ;*****************************************************************************
 5016  015040  005212                  TST172: INC     (R2)            ;UPDATE TEST NUMBER
 5017  015042  022712  000172                  CMP     #172,(R2)       ;SEQUENCE ERROR?
 5018  015046  001011                          BNE     TST173-10       ;BR TO ERROR HALT ON SEQ ERROR
 5019  015050  012737  125652  000000          MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
 5020  015056  005000                          CLR     R0              ;R0=0
 5021  015060  000310                          SWAB    (R0)            ;TRY SWAB MODE 1
 5022  015062  022737  125253  000000          CMP     #125253,@#0     ;CHECK RESULT
 5023  015070  001404                          BEQ     TST173
 5024                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5025                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 5026                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 5027                                                                    ;           WHICH FOLLOWS W/ 767           <====
 5028  015072  012742  000375                  MOV     #375,-(R2)      ;MOVE TO MAILBOX #  *******  375  *******
 5029  015076  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5030  015100  000000                          HALT                    ;RESULT OF SWAB MODE 1 FAILED
 5031                                                                    ;  OR SEQUENCE ERROR
```



```
 5032
 5033
 5034                                  ;*****************************************************************************
 5035                                  ;
 5036                                  ;       THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION.  THE TEST
 5037                                  ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE MODE
 5038                                  ;2 ADDRESSING REGISTER.  THE RESULTS ARE CHECKED WITH A COMPARE.
 5039                                  ;R0 IS CHECKED FOR PROPER DECREMENTING.
 5040                                  ;
 5041                                  ;*****************************************************************************
 5042                                  ;TEST 173       TEST MODE 2 W/ SWAB INST
 5043                                  ;*****************************************************************************
 5044  015102  005212                  TST173: INC     (R2)            ;UPDATE TEST NUMBER
 5045  015104  022712  000173                  CMP     #173,(R2)       ;SEQUENCE ERROR?
 5046  015110  001020                          BNE     TST174-10       ;BR TO ERROR HALT ON SEQ ERROR
 5047  015112  012737  125152  000000          MOV     #125152,@#0     ;MOVE TEST PATTERN TO LOC. 0
 5048  015120  005000                          CLR     R0              ;R0=0
 5049  015122  000320                          SWAB    (R0)+           ;TRY SWAB MODE 2
 5050  015124  022737  065252  000000          CMP     #65252,@#0      ;CHECK RESULT
 5051  015132  001404                          BEQ     SB2
 5052                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5053                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 5054                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 5055                                                                    ;           WHICH FOLLOWS W/ 767           <====
 5056  015134  012742  000376                  MOV     #376,-(R2)      ;MOVE TO MAILBOX #  *******  376  *******
 5057  015140  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5058  015142  000000                          HALT                    ;RESULT OF SWAB MODE 0 FAILED
 5059  015144  162700  000002          SB2:    SUB     #2,R0           ;CHECK EFFECT OF REG.
 5060  015150  001404                          BEQ     TST174
 5061                                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5062                                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 5063                                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 5064                                                                    ;           WHICH FOLLOWS W/ 760           <====
 5065  015152  012742  000377                  MOV     #377,-(R2)      ;MOVE TO MAILBOX #  *******  377  *******
 5066  015156  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5067  015160  000000                          HALT                    ;REGISTER VALUE INCORRECT
 5068                                                                    ;  OR SEQUENCE ERROR
 5069
 5070
 5071                                  ;*****************************************************************************
 5072                                  ;
 5073                                  ;       THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION.  THE TEST
 5074                                  ;PATTERN IS MOVED TO LOC 0.  A MODE 3 SWAB INSTRUCTION IS EXECUTED
 5075                                  ;USING R7 AS THE ADDRESSING REGISTER.  A COMPARE VERIFIES THE
 5076                                  ;DATA RESULTS.
 5077                                  ;
 5078                                  ;*****************************************************************************
 5079                                  ;TEST 174       TEST MODE 3 W/SWAB INST.
 5080                                  ;*****************************************************************************
 5081  015162  005212                  TST174: INC     (R2)            ;UPDATE TEST NUMBER
 5082  015164  022712  000174                  CMP     #174,(R2)       ;SEQUENCE ERROR?
 5083  015170  001011                          BNE     TST175-10       ;BR TO ERROR HALT ON SEQ ERROR
 5084  015172  012737  000377  000000          MOV     #377,@#0        ;MOVE TEST PATTERN TO LOC. 0
 5085  015200  000337  000000                  SWAB    @#0             ;TRY SWAB W/ MODE 3
 5086  015204  022737  177400  000000          CMP     #177400,@#0     ;CHECK RESULT
 5087  015212  001404                          BEQ     TST175
```

```
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
015214  012742  000400                MOV     #400,-(R2)      ;MOVE TO MAILBOX #  *******  400  *******
015220  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015222  000000                        HALT                    ;RESULT OF SWAB INCORRECT
                                                        ;  OR SEQUENCE ERROR
```



```
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
                        ;IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
                        ;REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
                        ;FOR PROPER DECREMENTING.
                        ;
                        ;*****************************************************************************
                        ;TEST 175       TEST MODE 4 W/ SWAB INST
                        ;*****************************************************************************
015224  005212          TST175: INC     (R2)            ;UPDATE TEST NUMBER
015226  022712  000175          CMP     #175,(R2)       ;SEQUENCE ERROR?
015232  001020                  BNE     TST176-10       ;BR TO ERROR HALT ON SEQ ERROR
015234  012737  125652  000000  MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
015242  012700  000002          MOV     #2,R0           ;SET UP REGISTER POINTER
015246  000340                  SWAB    -(R0)           ;TRY SWAB MODE 4
015250  022737  125253  000000  CMP     #125253,@#0     ;CHECK RESULT
015256  001404                  BEQ     SB4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 766           <====
015260  012742  000401                MOV     #401,-(R2)      ;MOVE TO MAILBOX #  *******  401  *******
015264  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015266  000000                        HALT                    ;RESULT OF SWAB INCORRECT
015270  005700                SB4:    TST     R0              ;CHECK EFFECT ON REG.
015272  001404                        BEQ     TST176
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 760           <====
015274  012742  000402                MOV     #402,-(R2)      ;MOVE TO MAILBOX #  *******  402  *******
015300  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015302  000000                        HALT                    ;REGISTER VALUE INCORRECT
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION.  THE TEST USES
                                ;TWO LOCATIONS FOLLOWING THE TEST CODE.  SB5X HOLDS THE DATA;
                                ;SB5XAD IS A POINTER TO THE DATA LOCATION.  THE DATA IS MOVED TO
                                ;SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD.  FOLLOWING
                                ;THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA.  R0 IS
                                ;CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
                                ;
                                ;*****************************************************************************
                                ;TEST 176        TEST MODE 5 W/ SWAB INST.
                                ;*****************************************************************************
015304  005212                  TST176: INC     (R2)            ;UPDATE TEST NUMBER
015306  022712  000176                  CMP     #176,(R2)       ;SEQUENCE ERROR?
015312  001021                          BNE     SB5             ;BR TO ERROR HALT ON SEQ ERROR
015314  012700  015372                  MOV     #SB5XAD+2,R0    ;SET UP POINTER TO WORK LOCATION
015320  012767  125125  000040          MOV     #125125,SB5X    ;MOVE PATTERN TO WORK LOCATION
015326  000350                          SWAB    @(R0)           ;TRY SWAB MODE 5
015330  022767  052652  000030          CMP     #52652,SB5X     ;CHECK RESULT
015336  001404                          BEQ     SB5A
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                ;           CONDITIONAL BRANCH INST. AND    <====
                                                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                ;           WHICH FOLLOWS W/ 766            <====
015340  012742  000403                          MOV     #403,-(R2)      ;MOVE TO MAILBOX #  *******  403  *******
015344  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015346  000000                                  HALT                    ;RESULT OF SWAB INCORRECT
015350  020027  015370          SB5A:   CMP     R0,#SB5XAD      ;CHECK RESULT OF REG.
015354  001406                          BEQ     TST177
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                ;           CONDITIONAL BRANCH INST. AND    <====
                                                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                ;           WHICH FOLLOWS W/ 757            <====
015356                          SB5:
015356  012742  000404                  MOV     #404,-(R2)      ;MOVE TO MAILBOX #  *******  404  *******
015362  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015364  000000                          HALT                    ;REGISTER VALUE INCORRECT
                                                                ;  OR SEQUENCE ERROR
015366  000000                  SB5X:   0                       ;WORK LOCATION
015370  015366                  SB5XAD: SB5X
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION.  THIS TEST
                                ;USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE.  TEST DATA
                                ;IS LOADED INTO THE WORK LOCATION.  R0, THE ADDRESSING REGISTER
                                ;IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
                                ;THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET.  THE DATA IS
                                ;VERIFIED WITH A COMPARE.
                                ;
                                ;*****************************************************************************
                                ;TEST 177        TEST MODE 6 W/ SWAB INST.
                                ;*****************************************************************************
015372  005212                  TST177: INC     (R2)            ;UPDATE TEST NUMBER
015374  022712  000177                  CMP     #177,(R2)       ;SEQUENCE ERROR?
015400  001013                          BNE     SB6             ;BR TO ERROR HALT ON SEQ ERROR
015402  012767  125125  000030          MOV     #125125,SB6X    ;MOVE PATTERN TO WORK LOCATION
015410  012700  015432                  MOV     #SB6X-6,R0      ;MOVE OFFSET POINTER TO R0
015414  000360  000006                  SWAB    6(R0)           ;TRY SWAB W/ MODE 6
015420  022760  052652  000006          CMP     #52652,6(R0)    ;CHECK RESULT
015426  001405                          BEQ     TST200
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                                ;           CONDITIONAL BRANCH INST. AND    <====
                                                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                                                ;           WHICH FOLLOWS W/ 765            <====
015430                          SB6:
015430  012742  000405                  MOV     #405,-(R2)      ;MOVE TO MAILBOX #  *******  405  *******
015434  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015436  000000                          HALT                    ;RESULT OF SWAB INCORRECT
                                                                ;  OR SEQUENCE ERROR
015440  000000                  SB6X:   0                       ;WORK LOCATION
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION.  THIS TEST
                                ;USES TWO LOCATIONS FOLLOWING THE TEST CODE:  A WORK LOCATION
                                ;(SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD).  DATA IS MOVED
                                ;TO THE WORK LOCATION.  R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
                                ;OF THE ADDRESS POINTER.  THE DATA IS SWAB'ED USING A MODE 7
                                ;INSTRUCTION WITH AN OFFSET OF +72.  THE DATA IS VERIFIED WITH A
                                ;COMPARE.
                                ;
                                ;*****************************************************************************
                                ;TEST 200       TEST MODE 7 W/ SWAB INST.
                                ;*****************************************************************************
015442  005212                TST200: INC     (R2)            ;UPDATE TEST NUMBER
015444  022712  000200                CMP     #200,(R2)       ;SEQUENCE ERROR?
015450  001013                        BNE     SB7             ;BR TO ERROR HALT ON SEQ ERROR
015452  012767  177400  000030        MOV     #177400,SB7X    ;MOVE PATTERN TO WORK LOCATION
015460  012700  015420                MOV     #SB7XAD-72,R0   ;MOVE OFFSET POINTER TO R0
015464  000370  000072                SWAB    @72(R0)         ;TRY SWAB MODE 7
015470  027027  000072  000377        CMP     @72(R0),#377    ;CHECK RESULTS
015476  001406                        BEQ     TST201
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 765            <====
015500                          SB7:
015500  012742  000406                MOV     #406,-(R2)      ;MOVE TO MAILBOX #  *******  406  *******
015504  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015506  000000                        HALT                    ;RESULT OF SWAB INCORRECT
                                                            ;  OR SEQUENCE ERROR
015510  000000                SB7X:   0                       ;WORK LOCATION
015512  015510                SB7XAD: SB7X                    ;POINTER TO WORK LOCATION
```



```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
                                ;BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
                                ;UTILIZES SEVERAL DIFFERENT TECHNIQUES.  THE CODE IS NOT EXECUTED
                                ;IN A LINEAR FASHION.  THE DIFFERENT MODES ARE EXECUTED IN ORDER
                                ;FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
                                ;FROGS THRU THE TEST CODE.  THE ORDER OF APPEARANCE OF THE CODE
                                ;IS:
                                ;       JMP MODE 1
                                ;       JMP MODE 3
                                ;       JMP MODE 2
                                ;       JMP MODE 4
                                ;       JMP MODE 6
                                ;       JMP MODE 5
                                ;       JMP MODE 7
                                ;AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
                                ;JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
                                ;       THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE.  EACH CODE
                                ;BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
                                ;THAT BLOCK.  A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK.  FOR
                                ;EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
                                ;OF THE PREVIOUS MODE 2 JUMP.  (ANY REGISTER CHANGES ARE VERIFIED
                                ;AND THE SEQUENCE CHECK IS MADE).  THEN THE REGISTERS ARE SETUP
                                ;FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
                                ;CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
                                ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
                                ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
                                ;THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
                                ;REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
                                ;
                                ;*****************************************************************************
                                ;TEST 201       TEST THE JMP INSTRUCTION IN ALL MODES
                                ;*****************************************************************************
015514  005212                TST201: INC     (R2)            ;UPDATE TEST NUMBER
015516  022712  000201                CMP     #201,(R2)       ;SEQUENCE ERROR?
015522  001150                        BNE     JMPCK+6         ;BR TO ERROR HALT ON SEQ ERROR
015524  005067  000326                CLR     JMPSEQ          ;ESTABLISH A SEQUENCE CHECKER
015530  012700  015610                MOV     #JMP2,R0        ;SET R0=JUMP TARGET
015534  000110                        JMP     (R0)            ;TRY JMP MODE 1
015536  022700  015540        JMP3:   CMP     #.+2,R0         ;CHECK RESULT OF MODE 2 JUMP
015542  001404                        BEQ     JMP3A
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 770            <====
015544  012742  000407                MOV     #407,-(R2)      ;MOVE TO MAILBOX #  *******  407  *******
015550  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
015552  000000                        HALT                    ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
015554  026727  000276  000001  JMP3A: CMP     JMPSEQ,#1       ;MAKE SURE JMPS ARE IN SEQUENCE: JMPSEQ=1?
015562  001404                        BEQ     JMP3B
                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                            ;           CONDITIONAL BRANCH INST. AND    <====
                                                            ;           REPLACE THE MOVE INSTRUCTION    <====
                                                            ;           WHICH FOLLOWS W/ 760            <====
```

```
 5298  015564  012742  000410                  MOV     #410,-(R2)      ;MOVE TO MAILBOX #  *******  410  *******
 5299  015570  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5300  015572  000000                          HALT                    ;SHOULD BE HERE FROM JMP MODE 2 ONLY
 5301  015574  012700  015606          JMP3B:  MOV     #IJMP4,R0       ;POINT R0 TO INDIRECT JMP ADDR.
 5302  015600  005267  000252                  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
 5303  015604  000130                          JMP     @(R0)+          ;TRY JMP MODE 3
 5304  015606  015640          IJMP4:  JMP4                            ;ADDRESS INDIRECT JUMP
 5305
 5306  015610  005767  000242          JMP2:   TST     JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
 5307  015614  001404                          BEQ     JMP2A
 5308                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5309                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5310                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5311                                                                  ;           WHICH FOLLOWS W/ 743           <====
 5312  015616  012742  000411                  MOV     #411,-(R2)      ;MOVE TO MAILBOX #  *******  411  *******
 5313  015622  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5314  015624  000000                          HALT                    ;SHOULD BE HERE FROM JMP MODE 1 ONLY
 5315  015626  005267  000224          JMP2A:  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
 5316  015632  012700  015536                  MOV     #JMP3,R0        ;SET R0=JUMP TARGET
 5317  015636  000120                          JMP     (R0)+           ;TRY A JUMP MODE 2 TO "JMP3"
 5318  015640  022700  015610          JMP4:   CMP     #IJMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
 5319  015644  001404                          BEQ     JMP4A
 5320                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5321                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5322                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5323                                                                  ;           WHICH FOLLOWS W/ 727           <====
 5324  015646  012742  000412                  MOV     #412,-(R2)      ;MOVE TO MAILBOX #  *******  412  *******
 5325  015652  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5326  015654  000000                          HALT                    ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
 5327  015656  022767  000002  000172  JMP4A:  CMP     #2,JMPSEQ       ;CHECK JUMP SEQUENCE: JMPSEQ=2?
 5328  015664  001404                          BEQ     JMP4B
 5329                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5330                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5331                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5332                                                                  ;           WHICH FOLLOWS W/ 717           <====
 5333  015666  012742  000413                  MOV     #413,-(R2)      ;MOVE TO MAILBOX #  *******  413  *******
 5334  015672  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5335  015674  000000                          HALT                    ;SHOULD BE ONLY FROM MODE 3 JUMP
 5336  015676  012700  015746          JMP4B:  MOV     #JMP5+2,R0      ;SET UP POINTER TO JUMP TARGET
 5337  015702  005267  000150                  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
 5338  015706  000140                          JMP     -(R0)           ;TRY JUMP MODE 4 TO "JMP4"
 5339
 5340  015710  022767  000004  000140  JMP6:   CMP     #4,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
 5341  015716  001404                          BEQ     JMP6A
 5342                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5343                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5344                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5345                                                                  ;           WHICH FOLLOWS W/ 702           <====
 5346  015720  012742  000414                  MOV     #414,-(R2)      ;MOVE TO MAILBOX #  *******  414  *******
 5347  015724  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5348  015726  000000                          HALT                    ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
 5349  015730  012700  016376          JMP6A:  MOV     #JMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
 5350  015734  005267  000116                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
 5351  015740  000160  177402                  JMP     -376(R0)        ;TRY MODE 6 JUMP
 5352
 5353  015744  022767  000003  000104  JMP5:   CMP     #3,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
```



```
 5354  015752  001404                          BEQ     JMP5A
 5355                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5356                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5357                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5358                                                                  ;           WHICH FOLLOWS W/ 664           <====
 5359  015754  012742  000415                  MOV     #415,-(R2)      ;MOVE TO MAILBOX #  *******  415  *******
 5360  015760  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5361  015762  000000                          HALT                    ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
 5362  015764  012700  016000          JMP5A:  MOV     #IJMP5+2,R0     ;SET UP POINTER TO INDIRECT JUMP ADDR.
 5363  015770  005267  000062                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
 5364  015774  000150                          JMP     @-(R0)          ;TRY JUMP MODE 5 TO "JMP6"
 5365  015776  015710          IJMP5:  JMP6                            ;INDIRECT ADDRESS POINTER
 5366
 5367  016000  022767  000005  000050  JMP7:   CMP     #5,JMPSEQ       ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
 5368  016006  001404                          BEQ     JMP7A
 5369                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5370                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5371                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5372                                                                  ;           WHICH FOLLOWS W/ 646           <====
 5373  016010  012742  000416                  MOV     #416,-(R2)      ;MOVE TO MAILBOX #  *******  416  *******
 5374  016014  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5375  016016  000000                          HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
 5376  016020  012700  016044          JMP7A:  MOV     #IJMP+10,R0     ;SET UP OFFSET POINTER TO INDIRECT ADDR.
 5377  016024  005267  000026                  INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
 5378  016030  000170  177770                  JMP     @-10(R0)        ;TRY MODE 7 JUMP
 5379  016034  016036          IJMP:   JMPCK                           ;INDIRECT ADDRESS
 5380
 5381  016036  026727  000014  000006  JMPCK:  CMP     JMPSEQ,#6       ;CHECK JUMPS IN SEQUENCE: JMPSEQ
 5382  016044  001405                          BEQ     TST202
 5383                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 5384                                                                  ;           CONDITIONAL BRANCH INST. AND   <====
 5385                                                                  ;           REPLACE THE MOVE INSTRUCTION   <====
 5386                                                                  ;           WHICH FOLLOWS W/ 627           <====
 5387  016046  012742  000417                  MOV     #417,-(R2)      ;MOVE TO MAILBOX #  *******  417  *******
 5388  016052  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 5389  016054  000000                          HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
 5390                                                                  ;  OR SEQUENCE ERROR
 5391  016056  000000          JMPSEQ: 0
```

```
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
                                ;THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
                                ;IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST).  EACH
                                ;BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
                                ;CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
                                ;THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
                                ;SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
                                ;REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
                                ;SUCCESSFUL.  R1 IS USED AS THE REGISTER IN ALL JSR  INSTRUCTIONS.
                                ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
                                ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
                                ;THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
                                ;REGISTER SAVED).
                                ;
                                ;*****************************************************************************
                                ;TEST 202        TEST JSR INSTRUCTION W/ ALL MODES
                                ;*****************************************************************************
016060  005212                  TST202: INC     (R2)            ;UPDATE TEST NUMBER
016062  022712  000202                  CMP     #202,(R2)       ;SEQUENCE ERROR?
016066  001001                          BNE     JSR0            ;BR TO ERROR HALT ON SEQ ERROR
016070  000402                          BR      JSR1
016072  000137  016526          JSR0:   JMP     @#JSRCK1

016076  012706  000500          JSR1:   MOV     #STBOT,R6       ;SET STACK POINTER
016102  012700  016210                  MOV     #JSR2,R0        ;SET TARGET ADDRESS
016106  005037  016506                  CLR     @#JSRSEQ        ;INITIALIZE SEQUENCE CHECKER
016112  005001                          CLR     R1              ;INITIALIZE R1
016114  005101                          COM     R1
016116  004110                          JSR     R1,(R0)         ;TRY JSR MODE 1
                                                                ; TO SCOPE:  REPLACE THE MOVE INSTRUCTION   <====
                                                                ;            FOLLOWING W/ 774               <====

016120                          JSR1A:
016120  012742  000420                  MOV     #420,-(R2)      ;MOVE TO MAILBOX #  *******  420  *******
016124  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016126  000000                          HALT                    ;JSR MODE 1 FAILED

016130  022737  000001  016506  JSR3:   CMP     #1,@# JSRSEQ    ;CHECK SEQUENCE: JSRSEQ=1?
016136  001014                          BNE     JSR3A           ;BRANCH IF OUT OF SEQUENCE
016140  020127  016272                  CMP     R1,#JSR4        ;PROPER PC SAVED?
016144  001011                          BNE     JSR3A           ;BRANCH IF PC WRONG
016146  022706  000476                  CMP     #STBOT-2,R6     ;STACK POINTER DECREMENTED?
016152  001006                          BNE     JSR3A           ;BRANCH IF SP WRONG
016154  022716  125252                  CMP     #125252,(R6)    ;REG SAVED ON STACK?
016160  001003                          BNE     JSR3A           ;BRANCH IF REG. NOT SAVED
016162  022700  016132                  CMP     #JSR3+2,R0      ;MODE 2 INCREMENT CORRECT?
016166  001404                          BEQ     JSR3B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 740           <====

016170                          JSR3A:
016170  012742  000421                  MOV     #421,-(R2)      ;MOVE TO MAILBOX #  *******  421  *******
016174  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```



```
016176  000000                          HALT                    ;JSR MODE 3 MALFUNCTIONED
016200  005237  016506          JSR3B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
016204  004137  016272                  JSR     R1,@#JSR4       ;TRY JSR MODE 4

016210  005737  016506          JSR2:   TST     @#JSRSEQ        ;CHECK SEQUENCE: JSRSEQ=0?
016214  001011                          BNE     JSR2A           ;BRANCH IF OUT OF SEQUENCE
016216  020127  016120                  CMP     R1,#JSR1A       ;PROPER PC SAVED?
016222  001006                          BNE     JSR2A           ;BRANCH IF PC WRONG
016224  022706  000476                  CMP     #STBOT-2,R6     ;R6 DECREMENT?
016230  001003                          BNE     JSR2A           ;BRANCH IF R6 IS INCORRECT
016232  021627  177777                  CMP     (R6),#-1        ;REGISTER SAVED?
016236  001404                          BEQ     JSR2B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 714           <====

016240                          JSR2A:
016240  012742  000422                  MOV     #422,-(R2)      ;MOVE TO MAILBOX #  *******  422  *******
016244  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016246  000000                          HALT                    ;JSR MODE 1 MALFUNCTIONED
016250  012706  000500          JSR2B:  MOV     #STBOT,R6       ;INITIALIZE R6
016254  012701  125252                  MOV     #125252,R1      ;INITIALIZE R1
016260  005237  016506                  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
016264  012700  016130                  MOV     #JSR3,R0        ;SET TARGET ADDRESS
016270  004120                          JSR     R1,(R0)+        ;TRY JSR MODE 2

016272  022737  000002  016506  JSR4:   CMP     #2,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=2?
016300  001003                          BNE     JSR4A           ;BRANCH IF OUT OF SEQUENCE
016302  022701  016210                  CMP     #JSR2,R1        ;PROPER PC SAVED?
016306  001404                          BEQ     JSR4B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 670           <====

016310                          JSR4A:
016310  012742  000423                  MOV     #423,-(R2)      ;MOVE TO MAILBOX #  *******  423  *******
016314  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016316  000000                          HALT                    ;JSR MODE 3 MALFUNCTIONED
016320  005237  016506          JSR4B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
016324  012700  016400                  MOV     #JSR5+2,R0      ;SET TARGET ADDRESS
016330  004140                          JSR     R1,-(R0)        ;TRY JSR MODE 4

016332  022767  000004  000146  JSR6:   CMP     #4,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=4?
016340  001006                          BNE     JSR6A           ;BRANCH IF OUT OF SEQUENCE
016342  022701  016444                  CMP     #JSR7,R1        ;PROPER PC SAVED?
016346  001003                          BNE     JSR6A           ;BRANCH IF PC WRONG
016350  022700  016502                  CMP     #JSR6AD,R0      ;MODE 5 REGISTER CORRECT?
016354  001404                          BEQ     JSR6B
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 645           <====

016356                          JSR6A:
016356  012742  000424                  MOV     #424,-(R2)      ;MOVE TO MAILBOX #  *******  424  *******
016362  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016364  000000                          HALT                    ;JSR MODE 5 FAILED
```

```
5504  016366  005237  016506          JSR6B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5505  016372  004167  000046                  JSR     R1,JSR7         ;TRY JSR MODE 6
5506  016376  022767  000003  000102  JSR5:   CMP     #3,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=3?
5507  016404  001006                          BNE     JSR5A           ;BRANCH IF OUT OF SEQUENCE
5508  016406  022701  016332                  CMP     #JSR6,R1        ;PROPER PC SAVED?
5509  016412  001003                          BNE     JSR5A           ;BRANCH IF PC WRONG
5510  016414  022700  016376                  CMP     #JSR5,R0        ;CHECK MODE 4 REGISTER
5511  016420  001404                          BEQ     JSR5B
5512                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
5513                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
5514                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
5515                                                                  ;           WHICH FOLLOWS W/ 623            <====
5516  016422                          JSR5A:
5517  016422  012742  000425                  MOV     #425,-(R2)      ;MOVE TO MAILBOX #  *******  425  *******
5518  016426  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5519  016430  000000                          HALT                    ;JSR MODE 4 MALFUNCTIONED
5520  016432  005237  016506          JSR5B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5521  016436  012700  016504                  MOV     #JSR6AD+2,R0    ;POINT R0 TO TARGET ADDRESS
5522  016442  004150                          JSR     R1,@-(R0)       ;TRY JSR MODE 5
5523
5524  016444  022737  000005  016506  JSR7:   CMP     #5,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=5?
5525  016452  001003                          BNE     JSR7A           ;BRANCH IF OUT OF SEQUENCE
5526  016454  022701  016376                  CMP     #JSR5,R1        ;PROPER PC SAVED?
5527  016460  001404                          BEQ     JSR7B
5528                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
5529                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
5530                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
5531                                                                  ;           WHICH FOLLOWS W/ 603            <====
5532  016462                          JSR7A:
5533  016462  012742  000426                  MOV     #426,-(R2)      ;MOVE TO MAILBOX #  *******  426  *******
5534  016466  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5535  016470  000000                          HALT                    ;JSR MODE 6 FAILED
5536  016472  005237  016506          JSR7B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
5537  016476  004177  000002                  JSR     R1,@JSRCKAD     ;TRY JSR MODE 7
5538
5539  016502  016332                  JSR6AD: JSR6                    ;MODE 5 TARGET ADDRESS
5540  016504  016510                  JSRCKAD:JSRCK                   ;MODE 7 TARGET ADDRESS
5541  016506  000000                  JSRSEQ: 0                       ;SEQUENCE CHECKER
5542
5543  016510  022767  000006  177770  JSRCK:  CMP     #6,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=6?
5544  016516  001003                          BNE     JSRCK1          ;BRANCH IF OUT OF SEQUENCE
5545  016520  022701  016502                  CMP     #JSR6AD,R1      ;PROPER PC SAVED?
5546  016524  001404                          BEQ     TST203
5547                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
5548                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
5549                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
5550                                                                  ;           WHICH FOLLOWS W/ 561            <====
5551  016526                          JSRCK1:
5552  016526  012742  000427                  MOV     #427,-(R2)      ;MOVE TO MAILBOX #  *******  427  *******
5553  016532  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5554  016534  000000                          HALT                    ;JSR MODE 7 MALFUNCTIONED
5555                                                                  ;  OR SEQUENCE ERROR
5556
5557
```



```
5558                                  ;*****************************************************************************
5559                                  ;
5560                                  ;       THIS TEST VERIFIES THE RTS INSTRUCTION.  THE STACK POINTER
5561                                  ;IS INITIALIZED AND A TEST PATTERN STORED ON STACK.  R0 IS LOADED
5562                                  ;WITH RETURN ADDRESS.  AN RTS IS EXECUTED, AND, AT THE TARGET
5563                                  ;ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
5564                                  ;STACK.
5565                                  ;*****************************************************************************
5566                                  ;TEST 203       TEST RTS INSTRUCTION
5567                                  ;*****************************************************************************
5568  016536  005212                  TST203: INC     (R2)            ;UPDATE TEST NUMBER
5569  016540  022712  000203                  CMP     #203,(R2)       ;SEQUENCE ERROR?
5570  016544  001016                          BNE     TST204-10       ;BR TO ERROR HALT ON SEQ ERROR
5571  016546  012706  000500                  MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
5572  016552  012746  052525                  MOV     #52525,-(R6)    ;INITIALIZE TOP OF STACK
5573  016556  012700  016574                  MOV     #RTS1,R0        ;INITIALIZE RETURN REGISTER
5574  016562  000200                          RTS     R0              ;TRY RTS THROUGH R0
5575                                                                  ; TO SCOPE: REPLACE THE MOVE INSTRUCTION    <====
5576                                                                  ;           FOLLOWING W/ 770                <====
5577  016564  012742  000430                  MOV     #430,-(R2)      ;MOVE TO MAILBOX #  *******  430  *******
5578  016570  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5579  016572  000000                          HALT                    ;RTS FAILED
5580  016574  022700  052525          RTS1:   CMP     #52525,R0       ;CHECK THAT R0 RESTORED FROM STACK
5581  016600  001404                          BEQ     TST204
5582                                                                  ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
5583                                                                  ;           CONDITIONAL BRANCH INST. AND    <====
5584                                                                  ;           REPLACE THE MOVE INSTRUCTION    <====
5585                                                                  ;           WHICH FOLLOWS W/ 762            <====
5586  016602  012742  000431                  MOV     #431,-(R2)      ;MOVE TO MAILBOX #  *******  431  *******
5587  016606  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5588  016610  000000                          HALT                    ;RTS MALFUNCTIONED
5589                                                                  ;  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
                        ;*****************************************************************************
016612  005212          TST204: INC     (R2)            ;UPDATE TEST NUMBER
016614  022712  000204          CMP     #204,(R2)       ;SEQUENCE ERROR?
016620  001022                  BNE     TST205-10       ;BR TO ERROR HALT ON SEQ ERROR
016622  000277                  SCC                     ;CC=0110
016624  000251                  +CLN!CLC
016626  012700  100000          MOV     #100000,R0      ;CC=1000
016632  101402                  BLOS    MOV1
016634  102401                  BVS     MOV1
016636  100404                  BMI     MOV2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
016640                  MOV1:
016640  012742  000432          MOV     #432,-(R2)      ;MOVE TO MAILBOX #  *******  432  *******
016644  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016646  000000                  HALT                    ;MOV DID NOT SET CC'S CORRECTLY
                        
016650  000277          MOV2:   SCC                     ;CC=1011
016652  000244                  CLZ
016654  012700  000000          MOV     #0,R0           ;CC=0101
016660  101002                  BHI     MOV3            ;C OR Z = 0?
016662  102401                  BVS     MOV3            ;V=1?
016664  100004                  BPL     TST205
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
016666                  MOV3:
016666  012742  000433          MOV     #433,-(R2)      ;MOVE TO MAILBOX #  *******  433  *******
016672  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
016674  000000                  HALT                    ;MOV DID NOT SET CC'S CORRECTLY
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 205       TEST BIT INSTRUCTION
                        ;*****************************************************************************
016676  005212          TST205: INC     (R2)            ;UPDATE TEST NUMBER
017000  022712  000205          CMP     #205,(R2)       ;SEQUENCE ERROR?
017004  001024                  BNE     TST206-10       ;BR TO ERROR HALT ON SEQ ERROR
```



```
017006  012700  100001          MOV     #100001,R0
017012  000277                  SCC                     ;CC=0110
017014  000251                  +CLN!CLC
017016  032700  100000          BIT     #100000,R0      ;CC=1000
017022  101402                  BLOS    BIT1
017024  102401                  BVS     BIT1
017026  100404                  BMI     BIT2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
017030                  BIT1:
017030  012742  000434          MOV     #434,-(R2)      ;MOVE TO MAILBOX #  *******  434  *******
017034  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017036  000000                  HALT                    ;BIT DID NOT SET CC'S CORRECTLY
                        
017040  000277          BIT2:   SCC                     ;CC=1011
017042  000244                  CLZ
017044  032700  077776          BIT     #77776,R0       ;CC=0101
017050  101002                  BHI     BIT3
017052  102401                  BVS     BIT3
017054  100004                  BPL     TST206
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
017056                  BIT3:
017056  012742  000435          MOV     #435,-(R2)      ;MOVE TO MAILBOX #  *******  435  *******
017062  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017064  000000                  HALT                    ;BIT DID NOT SET CC'S CORRECTLY
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 206       TEST BIC INSTRUCTION
                        ;*****************************************************************************
017066  005212          TST206: INC     (R2)            ;UPDATE TEST NUMBER
017070  022712  000206          CMP     #206,(R2)       ;SEQUENCE ERROR?
017074  001024                  BNE     TST207-10       ;BR TO ERROR HALT ON SEQ ERROR
017076  012700  177777          MOV     #177777,R0
017102  000277                  SCC                     ;CC=0110
017104  000251                  +CLN!CLC
017106  042700  077777          BIC     #77777,R0       ;CC=1000
017112  101402                  BLOS    BIC1
017114  102401                  BVS     BIC1
017116  100404                  BMI     BIC2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
017120                  BIC1:
017120  012742  000436          MOV     #436,-(R2)      ;MOVE TO MAILBOX #  *******  436  *******
017124  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017126  000000                  HALT                    ;BIC DID NOT SET CC'S CORRECTLY
017130  000277          BIC2:   SCC                     ;CC=1011
017132  000244                  CLZ
017134  042700  100000          BIC     #100000,R0      ;CC=0101
017140  101002                  BHI     BIC3
```

```
5702  017042  102401                    BVS     BIC3
5703  017044  100004                    BPL     TST207
5704                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5705                                                      ;           CONDITIONAL BRANCH INST. AND   <====
5706                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
5707                                                      ;           WHICH FOLLOWS W/ 754           <====
5708  017046                        BIC3:
5709  017046  012742  000437              MOV     #437,-(R2)      ;MOVE TO MAILBOX #  *******  437  *******
5710  017052  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5711  017054  000000                    HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5712                                                            ;  OR SEQUENCE ERROR
5713                                    ;*****************************************************************************
5714                                    ;TEST 207       TEST BIS INSTRUCTION
5715                                    ;*****************************************************************************
5716  017056  005212                TST207: INC     (R2)            ;UPDATE TEST NUMBER
5717  017060  022712  000207              CMP     #207,(R2)       ;SEQUENCE ERROR?
5718  017064  001025                    BNE     TST210-10       ;BR TO ERROR HALT ON SEQ ERROR
5719  017066  005000                    CLR     R0              ;R0=0
5720  017070  000277                    SCC                     ;CC=1010
5721  017072  000251                    +CLN!CLC
5722  017074  052700  000000              BIS     #0,R0           ;CC=0100  R0=0
5723  017100  103403                    BCS     BIS1
5724  017102  102402                    BVS     BIS1
5725  017104  100401                    BMI     BIS1
5726  017106  001404                    BEQ     BIS2
5727                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5728                                                      ;           CONDITIONAL BRANCH INST. AND   <====
5729                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
5730                                                      ;           WHICH FOLLOWS W/ 767           <====
5731  017110                        BIS1:
5732  017110  012742  000440              MOV     #440,-(R2)      ;MOVE TO MAILBOX #  *******  440  *******
5733  017114  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5734  017116  000000                    HALT                    ;BIS DID NOT SET CC'S CORRECTLY
5735  017120  000277                BIS2:   SCC                     ;CC=0111
5736  017122  000250                    CLN
5737  017124  052700  177777              BIS     #177777,R0      ;CC=1001
5738  017130  103003                    BCC     BIS3
5739  017132  102402                    BVS     BIS3
5740  017134  001401                    BEQ     BIS3
5741  017136  100404                    BMI     TST210
5742                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5743                                                      ;           CONDITIONAL BRANCH INST. AND   <====
5744                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
5745                                                      ;           WHICH FOLLOWS W/ 753           <====
5746  017140                        BIS3:
5747  017140  012742  000441              MOV     #441,-(R2)      ;MOVE TO MAILBOX #  *******  441  *******
5748  017144  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5749  017146  000000                    HALT                    ;BIS DID NOT SET CC'S CORRECTLY
5750                                                            ;  OR SEQUENCE ERROR
```



```
5751
5752                                    ;*****************************************************************************
5753                                    ;
5754                                    ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
5755                                    ;DEC INSTRUCTIONS.  THESE INSTRUCTIONS BOTH EFFECT THE C AND V
5756                                    ;BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
5757                                    ;UPON THE DATA RESULTS.  THE SAME PROCEDURE IS USED.  THE CONDITION
5758                                    ;CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
5759                                    ;RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
5760                                    ;THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
5761                                    ;DIFFERENT COMBINATIONS OF THE C AND V BITS.
5762                                    ;
5763                                    ;*****************************************************************************
5764                                    ;TEST 210       TEST INC INSTRUCTION
5765                                    ;*****************************************************************************
5766  017150  005212                TST210: INC     (R2)            ;UPDATE TEST NUMBER
5767  017152  022712  000210              CMP     #210,(R2)       ;SEQUENCE ERROR?
5768  017156  001037                    BNE     TST211-10       ;BR TO ERROR HALT ON SEQ ERROR
5769  017160  012700  077777              MOV     #077777,R0      ;R0=077777
5770  017164  000257                    CCC                     ;CC=0100
5771  017166  000264                    SEZ
5772  017170  005200                    INC     R0              ;CC=1010   R0=10000
5773  017172  101402                    BLOS    INC1
5774  017174  100001                    BPL     INC1
5775  017176  102404                    BVS     INC2
5776                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5777                                                      ;           CONDITIONAL BRANCH INST. AND   <====
5778                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
5779                                                      ;           WHICH FOLLOWS W/ 770           <====
5780  017200                        INC1:
5781  017200  012742  000442              MOV     #442,-(R2)      ;MOVE TO MAILBOX #  *******  442  *******
5782  017204  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5783  017206  000000                    HALT                    ;INC DID NOT SET CC'S CORRECTLY
5784  017210  052700  077777          INC2:   BIS     #77777,R0       ;R0=177777
5785  017214  000261                    SEC                     ;CC=1011
5786  017216  000244                    CLZ
5787  017220  005200                    INC     R0              ;CC=0101   R0=0
5788  017222  100403                    BMI     INC3
5789  017224  102402                    BVS     INC3
5790  017226  103001                    BCC     INC3
5791  017230  001404                    BEQ     INC4
5792                                                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5793                                                      ;           CONDITIONAL BRANCH INST. AND   <====
5794                                                      ;           REPLACE THE MOVE INSTRUCTION   <====
5795                                                      ;           WHICH FOLLOWS W/ 753           <====
5796  017232                        INC3:
5797  017232  012742  000443              MOV     #443,-(R2)      ;MOVE TO MAILBOX #  *******  443  *******
5798  017236  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5799  017240  000000                    HALT                    ;INC DID NOT SET CC'S CORRECTLY
5800
5801  017242  000277                INC4:   SCC                     ;CC=1110
5802  017244  000241                    CLC
5803  017246  005200                    INC     R0              ;CC=0000   R0=1
5804  017250  101402                    BLOS    INC5
5805  017252  100401                    BMI     INC5
5806  017254  100004                    BPL     TST211
```

```
5807                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5808                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5809                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5810                                                    ;           WHICH FOLLOWS W/ 741           <====
5811  017256                            INC5:
5812  017256  012742  000444                  MOV     #444,-(R2)      ;MOVE TO MAILBOX #  *******  444  *******
5813  017262  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5814  017264  000000                    HALT                    ;INC DID NOT SET CC'S CORRECTLY
5815                                                            ;  OR SEQUENCE ERROR
5816
5817                                    ;*****************************************************************************
5818                                    ;TEST 211       TEST DEC INSTRUCTION
5819                                    ;*****************************************************************************
5820  017266  005212                    TST211: INC     (R2)            ;UPDATE TEST NUMBER
5821  017270  022712  000211                    CMP     #211,(R2)       ;SEQUENCE ERROR?
5822  017274  001051                    BNE     TST212-10       ;BR TO ERROR HALT ON SEQ ERROR
5823  017276  012700  000002                  MOV     #2,R0           ;R0=2
5824  017302  000277                    SCC                     ;CC=1111
5825  017304  005300                    DEC     R0              ;CC=0001   R0=1
5826  017306  100403                    BMI     DEC1
5827  017310  001402                    BEQ     DEC1
5828  017312  102401                    BVS     DEC1
5829  017314  103404                    BCS     DEC2
5830                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5831                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5832                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5833                                                    ;           WHICH FOLLOWS W/ 770           <====
5834  017316                            DEC1:
5835  017316  012742  000445                  MOV     #445,-(R2)      ;MOVE TO MAILBOX #  *******  445  *******
5836  017322  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5837  017324  000000                    HALT                    ;DEC DID NOT SET CC'S CORRECTLY
5838  017326  000261                    DEC2:   SEC                     ;CC=1011
5839  017330  000244                    CLZ
5840  017332  005300                    DEC     R0              ;CC=0101   R0=0
5841  017334  101002                    BHI     DEC3
5842  017336  100401                    BMI     DEC3
5843  017340  102004                    BVC     DEC4
5844                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5845                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5846                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5847                                                    ;           WHICH FOLLOWS W/ 756           <====
5848  017342                            DEC3:
5849  017342  012742  000446                  MOV     #446,-(R2)      ;MOVE TO MAILBOX #  *******  446  *******
5850  017346  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5851  017350  000000                    HALT                    ;DEC DID NOT SET CC'S CORRECTLY
5852  017352  000277                    DEC4:   SCC                     ;CC=0110
5853  017354  000251                    +CLN!CLC
5854  017356  005300                    DEC     R0              ;CC=1000   R0=177777
5855  017360  101402                    BLOS    DEC5
5856  017362  102401                    BVS     DEC5
5857  017364  100404                    BMI     DEC6
5858                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5859                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5860                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5861                                                    ;           WHICH FOLLOWS W/ 744           <====
5862  017366                            DEC5:
```



```
5863  017366  012742  000447                  MOV     #447,-(R2)      ;MOVE TO MAILBOX #  *******  447  *******
5864  017372  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5865  017374  000000                    HALT                    ;DEC DID NOT SET CC'S CORRECTLY
5866  017376  042700  077777            DEC6:   BIC     #77777,R0       ;R0=100000
5867  017402  000277                    SCC                     ;CC=0101
5868  017404  000252                    +CLN!CLV
5869  017406  005300                    DEC     R0              ;CC=1011   R0=77777
5870  017410  100403                    BMI     DEC7            ;CC=0011
5871  017412  001402                    BEQ     DEC7
5872  017414  102001                    BVC     DEC7
5873  017416  103404                    BCS     TST212
5874                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
5875                                                    ;           CONDITIONAL BRANCH INST. AND   <====
5876                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
5877                                                    ;           WHICH FOLLOWS W/ 727           <====
5878  017420                            DEC7:
5879  017420  012742  000450                  MOV     #450,-(R2)      ;MOVE TO MAILBOX #  *******  450  *******
5880  017424  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5881  017426  000000                    HALT                    ;DEC DID NOT SET CC'S CORRECTLY
5882                                                            ;  OR SEQUENCE ERROR
5883
```

```
                                ;*****************************************************************************
                                ;
                                ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
                                ;TST, AND SWAB INSTRUCTIONS.  THESE THREE INSTRUCTIONS ALL LEAVE
                                ;THE C AND V BITS CLEARED.  AGAIN, THE CONDITION CODES ARE PRESET,
                                ;THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
                                ;BRANCH INSTRUCTIONS.  THE PROCEDURE IS REPEATED TO PRODUCE OTHER
                                ;COMBINATIONS OF CONDITION CODES.
                                ;
                                ;*****************************************************************************
                                ;TEST 212        TEST CLR INSTRUCTION
                                ;*****************************************************************************
017430  005212                  TST212: INC     (R2)            ;UPDATE TEST NUMBER
017432  022712  000212                  CMP     #212,(R2)       ;SEQUENCE ERROR?
017436  001007                          BNE     TST213-10       ;BR TO ERROR HALT ON SEQ ERROR
017440  000277                          SCC                     ;CC=1011
017442  000244                          CLZ
017444  005000                          CLR     R0              ;CC=0100   R0=0
017446  100403                          BMI     CLR1
017450  102402                          BVS     CLR1
017452  103401                          BCS     CLR1
017454  001404                          BEQ     TST213
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
017456                          CLR1:
017456  012742  000451                  MOV     #451,-(R2)      ;MOVE TO MAILBOX #  *******  451  *******
017462  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017464  000000                          HALT                    ;CLR DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 213        TEST TST INSTRUCTION
                                ;*****************************************************************************
017466  005212                  TST213: INC     (R2)            ;UPDATE TEST NUMBER
017470  022712  000213                  CMP     #213,(R2)       ;SEQUENCE ERROR?
017474  001022                          BNE     TST214-10       ;BR TO ERROR HALT ON SEQ ERROR
017476  000277                          SCC                     ;CC=1011
017500  000244                          CLZ
017502  005700                          TST     R0              ;CC=0100
017504  100403                          BMI     TEST1
017506  102402                          BVS     TEST1
017510  103401                          BCS     TEST1
017512  001404                          BEQ     TEST2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 771           <====
017514                          TEST1:
017514  012742  000452                  MOV     #452,-(R2)      ;MOVE TO MAILBOX #  *******  452  *******
017520  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017522  000000                          HALT                    ;TEST DID NOT SET CC'S CORRECTLY
017524  005300                  TEST2:  DEC     R0              ;MAKE R0 NEGATIVE
017526  000277                          SCC                     ;CC=0111
```



```
017530  000250                          CLN
017532  005700                          TST     R0              ;CC=1000
017534  101402                          BLOS    TEST3
017536  102401                          BVS     TEST3
017540  100404                          BMI     TST214
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 756           <====
017542                          TEST3:
017542  012742  000453                  MOV     #453,-(R2)      ;MOVE TO MAILBOX #  *******  453  *******
017546  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017550  000000                          HALT                    ;TEST DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;TEST 214        TEST SWAB INSTRUCTION
                                ;*****************************************************************************
017552  005212                  TST214: INC     (R2)            ;UPDATE TEST NUMBER
017554  022712  000214                  CMP     #214,(R2)       ;SEQUENCE ERROR?
017560  001023                          BNE     TST215-10       ;BR TO ERROR HALT ON SEQ ERROR
017562  012700  170000                  MOV     #170000,R0      ;R0=170000
017566  000277                          SCC                     ;CC=0111
017570  000250                          CLN
017572  000300                          SWAB    R0              ;CC=1000   R0=360
017574  101402                          BLOS    SWB1
017576  102401                          BVS     SWB1
017600  100404                          BMI     SWB2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
017602                          SWB1:
017602  012742  000454                  MOV     #454,-(R2)      ;MOVE TO MAILBOX #  *******  454  *******
017606  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017610  000000                          HALT                    ;SWAB DID NOT SET CC'S CORRECTLY
017612  000277                  SWB2:   SCC                     ;CC=1011
017614  000244                          CLZ
017616  000300                          SWAB    R0              ;CC=0100   R0=170000
017620  102403                          BVS     SWB3
017622  103402                          BCS     SWB3
017624  100401                          BMI     SWB3
017626  001404                          BEQ     TST215
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 755           <====
017630                          SWB3:
017630  012742  000455                  MOV     #455,-(R2)      ;MOVE TO MAILBOX #  *******  455  *******
017634  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017636  000000                          HALT                    ;
```

```
                        ;*****************************************************************************
                        ;
                        ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE ADD AND
                        ;ADC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE C AND
                        ;V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
                        ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
                        ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
                        ;BRANCHES.  THES PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
                        ;DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.
                        ;
                        ;*****************************************************************************
                        ;TEST 215       TEST ADD INSTRUCTION
                        ;*****************************************************************************
017640  005212          TST215: INC     (R2)            ;UPDATE TEST NUMBER
017642  022712  000215          CMP     #215,(R2)       ;SEQUENCE ERROR?
017646  001062                  BNE     TST216-10       ;BR TO ERROR HALT ON SEQ ERROR
017650  012700  040000          MOV     #40000,R0       ;R0=40000
017654  000277                  SCC                     ;CC=1111
017656  062700  030000          ADD     #30000,R0       ;CC=0000   R0=70000
017662  101402                  BLOS    ADD1
017664  102401                  BVS     ADD1
017666  100004                  BPL     ADD2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
017670                  ADD1:
017670  012742  000456          MOV     #456,-(R2)      ;MOVE TO MAILBOX #  *******  456  *******
017674  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017676  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
017700  000264          ADD2:   SEZ                     ;CC=0100
017702  062700  010000          ADD     #10000,R0       ;CC=1010   40=100000
017706  101402                  BLOS    ADD3
017710  102001                  BVC     ADD3
017712  100404                  BMI     ADD4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 756           <====
017714                  ADD3:
017714  012742  000457          MOV     #457,-(R2)      ;MOVE TO MAILBOX #  *******  457  *******
017720  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017722  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
017724  000257          ADD4:   CCC                     ;CC=1000
017726  000270                  SEN
017730  062700  100000          ADD     #100000,R0      ;CC=0111   R0=0
017734  101002                  BHI     ADD5
017736  102001                  BVC     ADD5
017740  100004                  BPL     ADD6
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 743           <====
017742                  ADD5:
```



```
017742  012742  000460          MOV     #460,-(R2)      ;MOVE TO MAILBOX #  *******  460  *******
017746  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017750  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
017752  062700  177777  ADD6:   ADD     #177777,R0      ;CC=1000   R0=177777
017756  101402                  BLOS    ADD7
017760  102401                  BVS     ADD7
017762  100404                  BMI     ADD8
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 732           <====
017764                  ADD7:
017764  012742  000461          MOV     #461,-(R2)      ;MOVE TO MAILBOX #  *******  461  *******
017770  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
017772  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
017774  000277          ADD8:   SCC                     ;CC=1010
017776  000245                  +CLC!CLZ
020000  062700  000001          ADD     #1,R0           ;CC=0101   R=0
020004  102403                  BVS     ADD9
020006  103002                  BCC     ADD9
020010  100401                  BMI     ADD9
020012  001404                  BEQ     TST216
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 716           <====
020014                  ADD9:
020014  012742  000462          MOV     #462,-(R2)      ;MOVE TO MAILBOX #  *******  462  *******
020020  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020022  000000                  HALT                    ;ADD DID NOT SET CC'S CORRECTLY
                                                        ;  OR SEQUENCE ERROR
                        ;*****************************************************************************
                        ;TEST 216       TEST ADC INSTRUCTION
                        ;*****************************************************************************
020024  005212          TST216: INC     (R2)            ;UPDATE TEST NUMBER
020026  022712  000216          CMP     #216,(R2)       ;SEQUENCE ERROR?
020032  001037                  BNE     TST217-10       ;BR TO ERROR HALT ON SEQ ERROR
020034  012700  077777          MOV     #077777,R0
020040  000277                  SCC                     ;CC=0101
020042  000252                  +CLN!CLV
020044  005500                  ADC     R0              ;CC=1010
020046  101402                  BLOS    ADC1
020050  102001                  BVC     ADC1
020052  100404                  BMI     ADC2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
020054                  ADC1:
020054  012742  000463          MOV     #463,-(R2)      ;MOVE TO MAILBOX #  *******  463  *******
020060  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020062  000000                  HALT                    ;ADC DID NOT SET CC'S CORRECTLY
020064  052700  077777  ADC2:   BIS     #77777,R0
020070  000277                  SCC                     ;CC=1011
020072  000244                  CLZ
```

```
6102  020074  005500                    ADC     R0              ;CC=0101   R0=0
6103  020076  101002                    BHI     ADC3
6104  020100  102401                    BVS     ADC3
6105  020102  100004                    BPL     ADC4
6106                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6107                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6108                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6109                                                            ;           WHICH FOLLOWS W/ 754           <====
6110  020104                        ADC3:
6111  020104  012742  000464              MOV     #464,-(R2)      ;MOVE TO MAILBOX #  *******  464  *******
6112  020110  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6113  020112  000000                      HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6114  020114  000277                ADC4: SCC
6115  020116  000245                      +CLZ!CLC
6116  020120  005500                      ADC     R0                      ;CC=1010
6117  020122  102403                      BVS     ADC5            ;CC=0100
6118  020124  103402                      BCS     ADC5
6119  020126  100401                      BMI     ADC5
6120  020130  001404                      BEQ     TST217
6121                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6122                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6123                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6124                                                            ;           WHICH FOLLOWS W/ 741           <====
6125  020132                        ADC5:
6126  020132  012742  000465              MOV     #465,-(R2)      ;MOVE TO MAILBOX #  *******  465  *******
6127  020136  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6128  020140  000000                      HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6129                                                            ;  OR SEQUENCE ERROR
```



```
6130
6131                                ;*******************************************************************************
6132                                ;
6133                                ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
6134                                ;CMP, AND COM INSTRUCTIONS.  EACH OF THESE INSTRUCTIONS GENERATE
6135                                ;THE C AND V BITS IDENTICALLY.  THE CONDITION CODES ARE PRESET,
6136                                ;THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
6137                                ;OF CONDITIONAL BRANCH INSTRUCTIONS.  THIS PROCEDURE IS REPEATED
6138                                ;SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
6139                                ;COMBINATIONS OF THE C AND V BITS.
6140                                ;
6141                                ;*******************************************************************************
6142                                ;TEST 217       TEST NEG INSTRUCTION
6143                                ;*******************************************************************************
6144  020142  005212                TST217: INC     (R2)            ;UPDATE TEST NUMBER
6145  020144  022712  000217               CMP     #217,(R2)       ;SEQUENCE ERROR?
6146  020150  001042                      BNE     .+TST250-10     ;BR TO ERROR HALT ON SEQ ERROR
6147  020152  012700  000001               MOV     #1,R0
6148  020156  000277                      SCC                     ;CC=0110
6149  020160  000251                      +CLN!CLC
6150  020162  005400                      NEG     R0              ;CC=1001   R0=177777
6151  020164  103003                      BCC     NEG1
6152  020166  102402                      BVS     NEG1
6153  020170  001401                      BEQ     NEG1
6154  020172  100404                      BMI     NEG2
6155
6156                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6157                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6158                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6159  020174                                                    ;           WHICH FOLLOWS W/ 767           <====
6160  020174  012742  000466       NEG1:  MOV     #466,-(R2)      ;MOVE TO MAILBOX #  *******  466  *******
6161  020200  005242                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6162  020202  000000                      HALT
6163  020204  042700  077777       NEG2:  BIC     #77777,R0       ;NEG DID NOT SET CC'S CORRECTLY
6164  020210  000257                      CCC                     ;CC=0100
6165  020212  000264                      SEZ
6166  020214  005400                      NEG     R0              ;CC=1011   R0=100000
6167  020216  102003                      BVC     NEG3
6168  020220  103002                      BCC     NEG3
6169  020222  001401                      BEQ     NEG3
6170  020224  100404                      BMI     NEG4
6171
6172                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
6173                                                            ;           CONDITIONAL BRANCH INST. AND   <====
6174                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
6175  020226                                                    ;           WHICH FOLLOWS W/ 752           <====
6176  020226  012742  000467       NEG3:
6177  020232  005242                      MOV     #467,-(R2)      ;MOVE TO MAILBOX #  *******  467  *******
6178  020234  000000                      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
6179  020236  005000                      HALT                    ;NEG DID NOT SET CC'S CORRECTLY
6180  020240  000277                NEG4: CLR     R0
6181  020242  000244                      SCC                     ;CC=1011
6182  020244  005400                      CLZ
6183  020246  102403                      NEG     R0              ;CC=0100   R0=0
6184  020250  103402                      BVS     NEG5
6185  020252  001001                      BCS     NEG5
                                        BNE     NEG5
```

```
                                        BPL     TST220
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 736           <====

                                NEG5:
020256  012742  000470                 MOV     #470,-(R2)      ;MOVE TO MAILBOX #  *******  470  *******
020262  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020264  000000                          HALT                    ;NEG DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 220       TEST CMP INSTRUCTION
                                ;*****************************************************************************
020266  005212                  TST220: INC     (R2)            ;UPDATE TEST NUMBER
020270  022712  000220                  CMP     #220,(R2)       ;SEQUENCE ERROR?
020274  001060                          BNE     TST221-10       ;BR TO ERROR HALT ON SEQ ERROR
020276  012700  000005                  MOV     #5,R0
020302  000257                          CCC                     ;CC=1010
020304  000271                          +SENISEC
020306  022700  000005                  CMP     #5,R0           ;CC=0101
020312  101002                          BHI     CMP1
020314  102401                          BVS     CMP1
020316  100004                          BPL     CMP2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
020320                          CMP1:
020320  012742  000471                 MOV     #471,-(R2)      ;MOVE TO MAILBOX #  *******  471  *******
020324  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020326  000000                          HALT                    ;CMP DID NOT SET CC'S CORRECTLY
020330  012700  100000          CMP2:   MOV     #100000,R0
020334  000277                          SCC                     ;CC=1101
020336  000242                          CLV
020340  020027  077777                  CMP     R0,#77777       ;CC=0010
020344  101402                          BLOS    CMP3
020346  102001                          BVC     CMP3
020350  100004                          BPL     CMP4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 752           <====
020352                          CMP3:
020352  012742  000472                 MOV     #472,-(R2)      ;MOVE TO MAILBOX #  *******  472  *******
020356  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020360  000000                          HALT                    ;CMP DID NOT SET CC'S CORRECTLY
020362  052700  040000          CMP4:   BIS     #40000,R0       ;R0=140000
020366  000257                          CCC                     ;CC=0100
020370  000264                          SEZ
020372  022700  040000                  CMP     #40000,R0       ;CC=1011
020376  102003                          BVC     CMP5
020400  103002                          BCC     CMP5
020402  001401                          BEQ     CMP5
020404  100404                          BMI     CMP6
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```



```
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 734           <====
020406                          CMP5:
020406  012742  000473                 MOV     #473,-(R2)      ;MOVE TO MAILBOX #  *******  473  *******
020412  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020414  000000                          HALT                    ;CMP DID NOT SET CC'S CORRECTLY
020416  042700  040000          CMP6:   BIC     #40000,R0
020422  000277                          SCC                     ;CC=1111
020424  022700  177777                  CMP     #-1,R0          ;CC=0000
020430  101402                          BLOS    CMP7
020432  102401                          BVS     CMP7
020434  100004                          BPL     TST221
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 720           <====
020436                          CMP7:
020436  012742  000474                 MOV     #474,-(R2)      ;MOVE TO MAILBOX #  *******  474  *******
020442  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020444  000000                          HALT                    ;CMP DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 221       TEST COM INSTRUCTION
                                ;*****************************************************************************
020446  005212                  TST221: INC     (R2)            ;UPDATE TEST NUMBER
020450  022712  000221                  CMP     #221,(R2)       ;SEQUENCE ERROR?
020454  001010                          BNE     TST222-10       ;BR TO ERROR HALT ON SEQ ERROR
020456  012700  177777                  MOV     #-1,R0
020462  000257                          CCC                     ;CC=1010
020464  000265                          +SECISEZ
020466  005100                          COM     R0              ;CC=0101
020470  101002                          BHI     COM1
020472  102401                          BVS     COM1
020474  100004                          BPL     TST222
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 770           <====
020476                          COM1:
020476  012742  000475                 MOV     #475,-(R2)      ;MOVE TO MAILBOX #  *******  475  *******
020502  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020504  000000                          HALT                    ;COM DID NOT SET CC'S CORRECTLY
                                                                ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************
                                ;
                                ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
                                ;AND SBC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE
                                ;C AND V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
                                ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
                                ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
                                ;BRANCHES.  THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
                                ;DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
                                ;
                                ;*****************************************************************************
                                ;TEST 222        TEST SUB INSTRUCTION
                                ;*****************************************************************************
020506  005212                  TST222: INC     (R2)            ;UPDATE TEST NUMBER
020510  022712  000222                  CMP     #222,(R2)       ;SEQUENCE ERROR?
020514  001055                        BNE     TST223-10       ;BR TO ERROR HALT ON SEQ ERROR
020516  012700  125252                MOV     #125252,R0
020522  000257                        CCC                     ;CC=1010
020524  000271                        +SEN!SEC
020526  162700  125252                SUB     #125252,R0      ;CC=0101   R0=0
020532  101002                        BHI     SUB1
020534  102401                        BVS     SUB1
020536  100004                        BPL     SUB2
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 767           <====
020540                          SUB1:
020540  012742  000476                MOV     #476,-(R2)      ;MOVE TO MAILBOX #  *******  476  *******
020544  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020546  000000                        HALT                    ;SUB DID NOT SET CC'S CORRECTLY
020550  052700  100000          SUB2: BIS     #100000,R0
020554  000277                        SCC                     ;CC=1101
020556  000242                        CLV
020560  162700  077777                SUB     #77777,R0       ;CC=0010   R0=1
020564  101402                        BLOS    SUB3
020566  102001                        BVC     SUB3
020570  100004                        BPL     SUB4
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 752           <====
020572                          SUB3:
020572  012742  000477                MOV     #477,-(R2)      ;MOVE TO MAILBOX #  *******  477  *******
020576  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020600  000000                        HALT                    ;
020602  005100                  SUB4: COM     R0              ;R0=177777
020604  000277                        SCC                     ;CC=11111
020606  162700  100000                SUB     #100000,R0      ;CC=0000   R0=77777
020612  101402                        BLOS    SUB5
020614  102401                        BVS     SUB5
020616  100004                        BPL     SUB6
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
```



```
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 737           <====
020620                          SUB5:
020620  012742  000500                MOV     #500,-(R2)      ;MOVE TO MAILBOX #  *******  500  *******
020624  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020626  000000                        HALT                    ;SUB DID NOT SET CC'S CORRECTLY
020630  000257                  SUB6: CCC                     ;CC=0100
020632  000264                        SEZ
020634  162700  140000                SUB     #140000,R0      ;CC=1011
020640  102003                        BVC     SUB7
020642  103002                        BCC     SUB7
020644  001401                        BEQ     SUB7
020646  100404                        BMI     TST223
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 723           <====
020650                          SUB7:
020650  012742  000501                MOV     #501,-(R2)      ;MOVE TO MAILBOX #  *******  501  *******
020654  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020656  000000                        HALT                    ;
                                ;*****************************************************************************
                                ;TEST 223        TEST SBC INSTRUCTION
                                ;*****************************************************************************
020660  005212                  TST223: INC     (R2)            ;UPDATE TEST NUMBER
020662  022712  000223                  CMP     #223,(R2)       ;SEQUENCE ERROR?
020666  001053                        BNE     TST224-10       ;BR TO ERROR HALT ON SEQ ERROR
020670  012700  000001                MOV     #1,R0
020674  000277                        SCC                     ;CC=1011
020676  000244                        CLZ
020700  005600                        SBC     R0              ;CC=0100   R=0
020702  103403                        BCS     SBC1
020704  102402                        BVS     SBC1
020706  100401                        BMI     SBC1
020710  001404                        BEQ     SBC2
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 767           <====
020712                          SBC1:
020712  012742  000502                MOV     #502,-(R2)      ;MOVE TO MAILBOX #  *******  502  *******
020716  005242                        INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
020720  000000                        HALT                    ;SBC DID NOT SET CC'S CORRECTLY
020722  000277                  SBC2: SCC                     ;CC=1010
020724  000245                        +CLZ!CLC
020726  005600                        SBC     R0              ;CC=0100   R=0
020730  103403                        BCS     SBC3
020732  102402                        BVS     SBC3
020734  100401                        BMI     SBC3
020736  001404                        BEQ     SBC4
                                                              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                              ;           CONDITIONAL BRANCH INST. AND   <====
                                                              ;           REPLACE THE MOVE INSTRUCTION   <====
                                                              ;           WHICH FOLLOWS W/ 754           <====
020740                          SBC3:
```

```
 6400  020740  012742  000503          MOV     #503,-(R2)      ;MOVE TO MAILBOX #  *******  503  *******
 6401  020744  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6402  020746  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
 6403  020750  000277          SBC4:   SCC                     ;CC=0111
 6404  020752  000250                  CLN
 6405  020754  005600                  SBC     R0              ;CC=1001   R0=177777
 6406  020756  103003                  BCC     SBC5
 6407  020760  102402                  BVS     SBC5
 6408  020762  001401                  BEQ     SBC5
 6409  020764  100404                  BMI     SBC6
 6410                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6411                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 6412                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 6413                                                          ;           WHICH FOLLOWS W/ 741           <====
 6414  020766          SBC5:
 6415  020766  012742  000504          MOV     #504,-(R2)      ;MOVE TO MAILBOX #  *******  504  *******
 6416  020772  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6417  020774  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
 6418  020776  042700  077777  SBC6:   BIC     #77777,R0       ;R0=100000
 6419  021002  000277                  SCC                     ;CC=1101
 6420  021004  000242                  CLV
 6421  021006  005600                  SBC     R0              ;CC=0010
 6422  021010  101402                  BLOS    SBC7
 6423  021012  102001                  BVC     SBC7
 6424  021014  100004                  BPL     TST224
 6425                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6426                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 6427                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 6428                                                          ;           WHICH FOLLOWS W/ 725           <====
 6429  021016          SBC7:
 6430  021016  012742  000505          MOV     #505,-(R2)      ;MOVE TO MAILBOX #  *******  505  *******
 6431  021022  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6432  021024  000000                  HALT                    ;SBC DID NOT SET CC'S CORRECTLY
 6433                                                          ;  OR SEQUENCE ERROR
 6434
```



```
 6435                          ;*****************************************************************************
 6436                          ;
 6437                          ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
 6438                          ;ROR, ASL AND ASR INSTRUCTIONS.  SPECIAL DATA PATTERNS ARE LOADED
 6439                          ;AND ROTATED SEVERAL TIMES FOR EACH TEST.  THE CONDITION CODES
 6440                          ;ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
 6441                          ;CHECKED AFTER EACH ROTATION.  THE FINAL CHECK IN EACH TEST IS
 6442                          ;TO VERIFY THE COMMULATIVE DATA RESULT.  THE DATA PATTERNS HAVE
 6443                          ;BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.
 6444                          ;
 6445                          ;*****************************************************************************
 6446                          ;TEST 224       TEST ROL INSTRUCTION
 6447                          ;*****************************************************************************
 6448  021026  005212  TST224: INC     (R2)            ;UPDATE TEST NUMBER
 6449  021030  027712  000224          CMP     #224,(R2)       ;SEQUENCE ERROR?
 6450  021034  001053          BNE     TST225-10       ;BR TO ERROR HALT ON SEQ ERROR
 6451  021036  012700  144000          MOV     #144000,R0      ;R0=144000
 6452  021042  000257                  CCC                     ;CC=0110
 6453  021044  000266                  +SEZ!SEV
 6454  021046  006100                  ROL     R0              ;CC=1001   R0=110000
 6455  021050  103003                  BCC     ROL1
 6456  021052  102402                  BVS     ROL1
 6457  021054  001401                  BEQ     ROL1
 6458  021056  100404                  BMI     ROL2
 6459                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6460                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 6461                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 6462                                                          ;           WHICH FOLLOWS W/ 767           <====
 6463  021060          ROL1:
 6464  021060  012742  000506          MOV     #506,-(R2)      ;MOVE TO MAILBOX #  *******  506  *******
 6465  021064  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6466  021066  000000                  HALT                    ;
 6467  021070  000277          ROL2:   SCC                     ;CC=1100
 6468  021072  000243                  +CLV!CLC
 6469  021074  006100                  ROL     R0              ;CC=0011   R0=020000
 6470  021076  103003                  BCC     ROL3
 6471  021100  102002                  BVC     ROL3
 6472  021102  001401                  BEQ     ROL3
 6473  021104  100004                  BPL     ROL4
 6474                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6475                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 6476                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
 6477                                                          ;           WHICH FOLLOWS W/ 754           <====
 6478  021106          ROL3:
 6479  021106  012742  000507          MOV     #507,-(R2)      ;MOVE TO MAILBOX #  *******  507  *******
 6480  021112  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 6481  021114  000000                  HALT                    ;ROL DID NOT SET CC'S CORRECTLY
 6482  021116  000277          ROL4:   SCC                     ;CC=0111
 6483  021120  000250                  CLN
 6484  021122  006100                  ROL     R0              ;CC=0000   R0=040001
 6485  021124  101402                  BLOS    ROL5
 6486  021126  102401                  BVS     ROL5
 6487  021130  100004                  BPL     ROL6
 6488                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 6489                                                          ;           CONDITIONAL BRANCH INST. AND   <====
 6490                                                          ;           REPLACE THE MOVE INSTRUCTION   <====
```

```
                                                        ;           WHICH FOLLOWS W/ 742          <====
021132                                  ROL5:
021132  012742  000510                          MOV     #510,-(R2)      ;MOVE TO MAILBOX #  *******  510  *******
021136  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021140  000000                                  HALT                    ;ROL DID NOT SET CC'S CORRECTLY
021142  000257                          ROL6:   CCC                     ;CC=0101
021144  000265                                  +SEZ!SEC
021146  006100                                  ROL     R0              ;CC=1010   R0=100003
021150  101405                                  BLOS    ROL7
021152  102004                                  BVC     ROL7
021154  100003                                  BPL     ROL7
021156  022700  100003                          CMP     #100003,R0
021162  001404                                  BEQ     TST225
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 725           <====
021164                                  ROL7:
021164  012742  000511                          MOV     #511,-(R2)      ;MOVE TO MAILBOX #  *******  511  *******
021170  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021172  000000                                  HALT                    ;ROL MALFUNCTIONED
                                                                        ; OR SEQUENCE ERROR
                                        ;*****************************************************************************
                                        ;TEST 225       TEST ROR INSTRUCTION
                                        ;*****************************************************************************
021174  005212                          TST225: INC     (R2)            ;UPDATE TEST NUMBER
021176  022712  000225                          CMP     #225,(R2)       ;SEQUENCE ERROR?
021202  001051                                  BNE     TST226-10       ;BR TO ERROR HALT ON SEQ ERROR
021204  012700  000023                          MOV     #23,R0          ;R0=23
021210  000277                                  SCC                     ;CC=0111
021212  000250                                  CLN
021214  006000                                  ROR     R0              ;CC=1001   R0=100011
021216  102403                                  BVS     ROR1
021220  103002                                  BCC     ROR1
021222  001401                                  BEQ     ROR1
021224  100404                                  BMI     ROR2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
021226                                  ROR1:
021226  012742  000512                          MOV     #512,-(R2)      ;MOVE TO MAILBOX #  *******  512  *******
021232  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021234  000000                                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
021236  000257                          ROR2:   CCC                     ;CC=1100
021240  000274                                  +SEN!SEZ
021242  006000                                  ROR     R0              ;CC=0011   R0=040004
021244  102003                                  BVC     ROR3
021246  103002                                  BCC     ROR3
021250  001401                                  BEQ     ROR3
021252  100004                                  BPL     ROR4
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 754           <====
021254                                  ROR3:
```



```
021254  012742  000513                          MOV     #513,-(R2)      ;MOVE TO MAILBOX #  *******  513  *******
021260  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021262  000000                                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
021264  000277                          ROR4:   SCC                     ;CC=1110
021266  000241                                  CLC
021270  006000                                  ROR     R0              ;CC=0000   R0=020002
021272  101403                                  BLOS    ROR5
021274  102402                                  BVS     ROR5
021276  001401                                  BEQ     ROR5
021300  100004                                  BPL     ROR6
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 741           <====
021302                                  ROR5:
021302  012742  000514                          MOV     #514,-(R2)      ;MOVE TO MAILBOX #  *******  514  *******
021306  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021310  000000                                  HALT                    ;ROR DID NOT SET CC'S CORRECTLY
021312  000257                          ROR6:   CCC                     ;CC=0101
021314  000265                                  +SEC!SEZ
021316  006000                                  ROR     R0              ;CC=1010   R0=110001
021320  101402                                  BLOS    ROR7
021322  102001                                  BVC     ROR7
021324  100404                                  BMI     TST226
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 727           <====
021326                                  ROR7:
021326  012742  000515                          MOV     #515,-(R2)      ;MOVE TO MAILBOX #  *******  515  *******
021332  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021334  000000                                  HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
                                                                        ; OR SEQUENCE ERROR
                                        ;*****************************************************************************
                                        ;TEST 226       TEST ASL INSTRUCTION
                                        ;*****************************************************************************
021336  005212                          TST226: INC     (R2)            ;UPDATE TEST NUMBER
021340  022712  000226                          CMP     #226,(R2)       ;SEQUENCE ERROR?
021344  001054                                  BNE     TST227-10       ;BR TO ERROR HALT ON SEQ ERROR
021346  012700  144000                          MOV     #144000,R0      ;R0=14000
021352  000257                                  CCC                     ;CC=0110
021354  000271                                  +SEN!SEC
021356  006300                                  ASL     R0              ;CC=1001   R0=110000
021360  103003                                  BCC     ASL1
021362  102402                                  BVS     ASL1
021364  001401                                  BEQ     ASL1
021366  100404                                  BMI     ASL2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 767           <====
021370                                  ASL1:
021370  012742  000516                          MOV     #516,-(R2)      ;MOVE TO MAILBOX #  *******  516  *******
021374  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021376  000000                                  HALT                    ;
021400  000277                          ASL2:   SCC                     ;CC=1100
```

```
                                        +CLVICLC
021402  000243                          ASL     R0              ;CC=0011   R0=020000
021404  006300                          BCC     ASL3
021406  103003                          BVC     ASL3
021410  102002                          BEQ     ASL3
021412  001401                          BPL     ASL4
021414  100004
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 754           <====
021416                          ASL3:
021416  012742  000517                  MOV     #517,-(R2)      ;MOVE TO MAILBOX #  *******  517  *******
021422  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021424  000000                          HALT                    ;ASL DID NOT SET CC'S CORRECTLY
021426  000277                  ASL4:   SCC                     ;CC=0111
021430  000250                          CLN
021432  006300                          ASL     R0              ;CC=0000   R0=040000
021434  101402                          BLOS    ASL5
021436  102401                          BVS     ASL5
021440  100004                          BPL     ASL6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 742           <====
021442                          ASL5:
021442  012742  000520                  MOV     #520,-(R2)      ;MOVE TO MAILBOX #  *******  520  *******
021446  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021450  000000                          HALT                    ;ASL DID NOT SET CC'S CORRECTLY
021452  000257                  ASL6:   CCC                     ;CC=0101
021454  000265                          +SEZISEC
021456  006300                          ASL     R0              ;CC=1010   R0=100000
021460  103406                          BCS     ASL7
021462  001405                          BEQ     ASL7
021464  102004                          BVC     ASL7
021466  100003                          BPL     ASL7
021470  022700  100000                  CMP     #100000,R0
021474  001404                          BEQ     TST227
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 724           <====
021476                          ASL7:
021476  012742  000521                  MOV     #521,-(R2)      ;MOVE TO MAILBOX #  *******  521  *******
021502  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021504  000000                          HALT                    ;ASL MALFUNCTIONED
                                                                ;  OR SEQUENCE ERROR
```



```
                                        ;*******************************************************************************
                                        ;TEST 227       TEST ASR INSTRUCTION
                                        ;*******************************************************************************
021506  005212                  TST227: INC     (R2)            ;UPDATE TEST NUMBER
021510  022712  000227                  CMP     #227,(R2)       ;SEQUENCE ERROR?
021514  001060                          BNE     TST230-10       ;BR TO ERROR HALT ON SEQ ERROR
021516  012700  100023                  MOV     #100023,R0      ;R0=100023
021522  000277                          SCC                     ;CC=0110
021524  000250                          CLN
021526  006200                          ASR     R0              ;CC=1001   RP=140011
021530  102403                          BVS     ASR1
021532  103002                          BCC     ASR1
021534  001401                          BEQ     ASR1
021536  100404                          BMI     ASR2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 767           <====
021540                          ASR1:
021540  012742  000522                  MOV     #522,-(R2)      ;MOVE TO MAILBOX #  *******  522  *******
021544  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021546  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
021550  042700  100000          ASR2:   BIC     #100000,R0      ;R0=40011
021554  000277                          SCC                     ;CC=1100
021556  000243                          +CLVICLC
021560  006200                          ASR     R0              ;CC=0011   R0=020004
021562  102003                          BVC     ASR3
021564  103002                          BCC     ASR3
021566  001401                          BEQ     ASR3
021570  100004                          BPL     ASR4
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 752           <====
021572                          ASR3:
021572  012742  000523                  MOV     #523,-(R2)      ;MOVE TO MAILBOX #  *******  523  *******
021576  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021600  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
021602  000277                  ASR4:   SCC                     ;CC=1111
021604  006200                          ASR     R0              ;CC=0000   R0=010002
021606  101403                          BLOS    ASR5
021610  102402                          BVS     ASR5
021612  001401                          BEQ     ASR5
021614  100004                          BPL     ASR6
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 740           <====
021616                          ASR5:
021616  012742  000524                  MOV     #524,-(R2)      ;MOVE TO MAILBOX #  *******  524  *******
021622  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021624  000000                          HALT                    ;ASR DID NOT SET CC'S CORRECTLY
021626  052700  100000          ASR6:   BIS     #100000,R0      ;R0=110002
021632  000257                          CCC                     ;CC=0101
021634  000265                          +SEZISEC
```

```
021636  006200                          ASR     R0              ;C=1010   R0=144001
021640  101406                          BLOS    ASR7
021642  102005                          BVC     ASR7
021644  100004                          BPL     ASR7
021646  001403                          BEQ     ASR7
021650  022700  144001                  CMP     #144001,R0      ;CHECK RESULT OF ASR'S
021654  001404                          BEQ     TST230
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 720           <====
021656                          ASR7:
021656  012742  000525                  MOV     #525,-(R2)      ;MOVE TO MAILBOX #  *******  525  *******
021662  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021664  000000                          HALT                    ;ASR DID NOT FUNCTION CORRECTLY
                                                                ;  OR SEQUENCE ERROR
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION CODES
                                ;ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
                                ;THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
                                ;CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
                                ;IS VERIFIED BY CONDITIONAL BRANCHES.
                                ;
                                ;*****************************************************************************
                                ;TEST 230        TEST THE SXT INSTRUCTION
                                ;*****************************************************************************
021666  005212                  TST230: INC     (R2)            ;UPDATE TEST NUMBER
021670  022712  000230                  CMP     #230,(R2)       ;SEQUENCE ERROR?
021674  001033                          BNE     TST231-10       ;BR TO ERROR HALT ON SEQ ERROR
021676  005000                          CLR     R0
021700  000277                          SCC                     ;SET CC=1011
021702  000244                          CLZ
021704  006700                          SXT     R0              ;TRY SXT
021706  100006                          BPL     SXT0            ;TEST CC=1001
021710  001405                          BEQ     SXT0
021712  102404                          BVS     SXT0
021714  103003                          BCC     SXT0
021716  022700  177777                  CMP     #-1,R0          ;CHECK DATA RESULT
021722  001404                          BEQ     SXT1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 765           <====
021724                          SXT0:
021724  012742  000526                  MOV     #526,-(R2)      ;MOVE TO MAILBOX #  *******  526  *******
021730  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021732  000000                          HALT                    ;RESULTS OF SXT INCORRECT
021734  005000                  SXT1:   CLR     R0              ;R0=0
021736  005010                          CLR     (R0)            ;LOC. 0=0
021740  005110                          COM     (R0)            ;LOC. 0=177777
021742  000257                          CCC                     ;SET CC=0110
021744  000266                          +SEZ!SEV
```



```
021746  006710                          SXT     (R0)
021750  001005                          BNE     SXT2            ;TEST CC=0100
021752  103404                          BCS     SXT2
021754  102403                          BVS     SXT2
021756  100402                          BMI     SXT2
021760  005710                          TST     (R0)
021762  001404                          BEQ     TST231
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                        ;           WHICH FOLLOWS W/ 745           <====
021764                          SXT2:
021764  012742  000527                  MOV     #527,-(R2)      ;MOVE TO MAILBOX #  *******  527  *******
021770  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
021772  000000                          HALT                    ;RESULTS OF SXT INCORRECT
                                                                ;  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
                        ;*****************************************************************************
022774  005212          TST231: INC     (R2)            ;UPDATE TEST NUMBER
022776  022712  000231          CMP     #231,(R2)       ;SEQUENCE ERROR?
023002  001035                  BNE     TST232-10       ;BR TO ERROR HALT ON SEQ ERROR
023004  012700  007463          MOV     #7463,R0        ;SET UP R0
023010  012701  031525          MOV     #31525,R1       ;SET UP R1
023014  000277                  SCC                     ;SET CC=1110
023016  000241                  CLC
023020  074100                  XOR     R1,R0           ;TRY XOR
023022  101406                  BLOS    XOR1            ;CC=0000?
023024  102405                  BVS     XOR1
023026  001404                  BEQ     XOR1
023030  100403                  BMI     XOR1
023032  022700  036146          CMP     #36146,R0       ;DATA RESULT CORRECT?
023036  001404                  BEQ     XOR2
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 762           <====
023040                  XOR1:
023040  012742  000530          MOV     #530,-(R2)      ;MOVE TO MAILBOX #  *******  530  *******
023044  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023046  000000                  HALT                    ;
023050  010104          XOR2:   MOV     R1,R4
023052  000261                  SEC                     ;CC=1110
023054  000241                  CLC
023056  074400                  XOR     R4,R0           ;TRY XOR MODE 0,0
023060  101406                  BLOS    XOR3            ;CC=0000?
023062  102405                  BVS     XOR3
023064  001404                  BEQ     XOR3
023066  100403                  BMI     XOR3
023070  022700  007463          CMP     #7463,R0
023074  001404                  BEQ     TST232
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 743           <====
023076                  XOR3:
023076  012742  000531          MOV     #531,-(R2)      ;MOVE TO MAILBOX #  *******  531  *******
023102  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023104  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
                        ;*****************************************************************************
023106  005212          TST232: INC     (R2)            ;UPDATE TEST NUMBER
023110  022712  000232          CMP     #232,(R2)       ;SEQUENCE ERROR?
023114  001023                  BNE     TST233-10       ;BR TO ERROR HALT ON SEQ ERROR
023116  012700  000525          MOV     #525,R0
023122  010004                  MOV     R0,R4
023124  000277                  SCC                     ;SET CC=1111
023126  101002          SOB1:   BHI     SOB2            ;CC=1111?
023130  100001                  BPL     SOB2
023132  102404                  BVS     SOB3
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 771           <====
023134                  SOB2:
023134  012742  000532          MOV     #532,-(R2)      ;MOVE TO MAILBOX #  *******  532  *******
023140  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023142  000000                  HALT                    ;
023144  005304          SOB3:   DEC     R4              ;COUNT ITERATIONS
023146  000277                  SCC                     ;CC=1111
023150  077012                  SOB     R0,SOB1         ;DO SOB W/ R0
023152  101004                  BHI     SOB4            ;CHECK CC=1111
023154  100003                  BPL     SOB4
023156  102002                  BVC     SOB4
023160  005704                  TST     R4              ;ITERATION COUNT OK?
023162  001404                  BEQ     TST233
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 755           <====
023164                  SOB4:
023164  012742  000533          MOV     #533,-(R2)      ;MOVE TO MAILBOX #  *******  533  *******
023170  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023172  000000                  HALT                    ;INCORRECT # OF BRANCHES OR CC'S CHANGED
                                                        ;  OR SEQUENCE ERROR
```

```
                                ;*****************************************************************************************
                                ;
                                ;       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
                                ;*****************************************************************************************
022174  005212                  TST233: INC     (R2)            ;UPDATE TEST NUMBER
022176  022712  000233                  CMP     #233,(R2)       ;SEQUENCE ERROR?
022202  001062                          BNE     TST234-10       ;BR TO ERROR HALT ON SEQ ERROR
022204  012706  000500                  MOV     #STBOT,SP
022210  012746  125252                  MOV     #125252,-(SP)   ;PUT R5 VALUE ON STACK
022214  162706  000074                  SUB     #74,SP          ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
022220  012705  022246                  MOV     #MRK1,R5        ;SET NEW PC IN R5
022224  012746  006436                  MOV     #6436,-(SP)     ;PUT MARK 36 INST. ON STACK
022230  000277                          SCC                     ;SET CC=1111
022232  000137  000400                  JMP     @#400           ;XFER CONTL TO MARK 36 INST. ON STACK
022236  012742  000534                  MOV     #534,-(R2)      ;MOVE TO MAILBOX #  *******  534  *******
022242  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022244  000000                          HALT                    ;MARK INST. SHOULD HAVE JUMPED TO MRK1
022246  101010                  MRK1:   BHI     MRK2            ;TEST CC UNAFFECTED
022250  100007                          BPL     MRK2            ;IE. CC=1111
022252  102006                          BVC     MRK2
022254  020527  125252                  CMP     R5,#125252      ;CHECK R5 RESTORED FROM STACK
022260  001003                          BNE     MRK2
022262  022706  000500                  CMP     #STBOT,R6       ;CHECK STACK POINTER READJUSTED CORRECTLY.
022266  001404                          BEQ     MRK3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                        ;           WHICH FOLLOWS W/ 746            <====
022270                          MRK2:
022270  012742  000535                  MOV     #535,-(R2)      ;MOVE TO MAILBOX #  *******  535  *******
022274  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022276  000000                          HALT                    ;RESULTS OF MARK INCORRECT
022300  012746  052525          MRK3:   MOV     #52525,-(SP)
022304  012746  006400                  MOV     #6400,-(SP)     ;PUT MARK 0 INST. ON STACK
022310  010605                          MOV     SP,R5           ;SET ADDR. OF MARK INST. IN R5
022312  004737  022322                  JSR     PC,@#MRK4       ;DO JSR
022316  000137  022334                  JMP     @#MRK5          ;
022322  000205          MRK4:   RTS     R5                      ;DO RTS WITH R5 TO MARK INST ON STACK
022324  012742  000536                  MOV     #536,-(R2)      ;MOVE TO MAILBOX #  *******  536  *******
022330  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022332  000000                          HALT                    ;RTS/MARK SEQUENCE FAILED
022334  022706  000500          MRK5:   CMP     #STBOT,R6       ;STACK ADJUSTED CORRECTLY
022340  001003                          BNE     MRK6            ;IF NOT:  BR
022342  022705  052525                  CMP     #52525,R5       ;CHECK IF R5 RESTORED FROM STACK
022346  001404                          BEQ     TST234
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                        ;           WHICH FOLLOWS W/ 716            <====
022350                          MRK6:
```



```
022350  012742  000537                  MOV     #537,-(R2)      ;MOVE TO MAILBOX #  *******  537  *******
022354  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022356  000000                          HALT                    ;RESULTS OF MARK INCORRECT
                                                                ;  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
                                ;*****************************************************************************
022360  005212                  TST234: INC     (R2)            ;UPDATE TEST NUMBER
022362  022712  000234                  CMP     #234,(R2)       ;SEQUENCE ERROR?
022366  001024                          BNE     TST235-10       ;BR TO ERROR HALT ON SEQ ERROR
022370  012700  000377                  MOV     #377,R0
022374  000257                          CCC
022376  106400                          MTPS    R0
022400  022767  000357  155370          CMP     #357,PS
022406  001404                          BEQ     MTPS1
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 770           <====
022410  012742  000540                  MOV     #540,-(R2)      ;MOVE TO MAILBOX #  *******  540  *******
022414  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022416  000000                          HALT                    ;MTPS FAILED
022420  005000                  MTPS1:  CLR     R0
022422  005010                          CLR     (R0)
022424  000277                          SCC                     ;CC=1111
022426  106410                          MTPS    (R0)            ;TRY MTPS MODE 1
022430  100403                          BMI     MTPS1A          ;CHECK PS
022432  102402                          BVS     MTPS1A
022434  103401                          BCS     MTPS1A
022436  001004                          BNE     TST235
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 754           <====
022440                  MTPS1A:
022440  012742  000541                  MOV     #541,-(R2)      ;MOVE TO MAILBOX #  *******  541  *******
022444  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022446  000000                          HALT                    ;MTPS FAILED
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 235        TEST MTPS MODE 2
                                ;*****************************************************************************
022450  005212                  TST235: INC     (R2)            ;UPDATE TEST NUMBER
022452  022712  000235                  CMP     #235,(R2)       ;SEQUENCE ERROR?
022456  001021                          BNE     TST236-10       ;BR TO ERROR HALT ON SEQ ERROR
022460  005000                          CLR     R0              ;R0=0
022462  012710  177777                  MOV     #-1,(R0)        ;LOC. 0=-1
022466  005037  177776                  CLR     @#PS            ;PS=0
022472  106420                          MTPS    (R0)+           ;TRY MTPS W/MODE 2
```



```
022474  022737  000357  177776          CMP     #357,@#PS       ;CHECK DATA
022502  001404                          BEQ     MTPS2
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 766           <====
022504  012742  000542                  MOV     #542,-(R2)      ;MOVE TO MAILBOX #  *******  542  *******
022510  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022512  000000                          HALT                    ;DEST. DATA INCORRECT
022514  022700  000001          MTPS2:  CMP     #1,R0           ;CHECK DEST. REGISTER.
022520  001404                          BEQ     TST236
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 757           <====
022522  012742  000543                  MOV     #543,-(R2)      ;MOVE TO MAILBOX #  *******  543  *******
022526  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022530  000000                          HALT                    ;DEST REGISTER NOT INCREMENTED BY 1
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 236        TEST MTPS MODE 3
                                ;*****************************************************************************
022532  005212                  TST236: INC     (R2)            ;UPDATE TEST NUMBER
022534  022712  000236                  CMP     #236,(R2)       ;SEQUENCE ERROR?
022540  001024                          BNE     TST237-10       ;BR TO ERROR HALT ON SEQ ERROR
022542  012700  000402                  MOV     #402,R0         ;R0=402
022546  005010                          CLR     (R0)            ;LOC. 402=0
022550  012737  052652  000000          MOV     #52652,@#0      ;LOC. 0=52652
022556  005037  177776                  CLR     @#PS            ;PS=0
022562  106430                          MTPS    @(R0)+          ;TRY MTPS W/MODE 3
022564  022737  000252  177776          CMP     #252,@#PS       ;CHECK DEST. DATA
022572  001404                          BEQ     MTPS3
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 763           <====
022574  012742  000544                  MOV     #544,-(R2)      ;MOVE TO MAILBOX #  *******  544  *******
022600  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022602  000000                          HALT                    ;DEST. DATA INCORRECT
022604  022700  000404          MTPS3:  CMP     #404,R0         ;CHECK MODE 3 REGISTER.
022610  001404                          BEQ     TST237
                                                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                ;           CONDITIONAL BRANCH INST. AND   <====
                                                                ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                ;           WHICH FOLLOWS W/ 754           <====
022612  012742  000545                  MOV     #545,-(R2)      ;MOVE TO MAILBOX #  *******  545  *******
022616  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
022620  000000                          HALT                    ;MODE 3 REGISTER INCORRECT
                                                                ;  OR SEQUENCE ERROR

                                ;*****************************************************************************
                                ;TEST 237        TEST MTPS MODE 4
                                ;*****************************************************************************
022622  005212                  TST237: INC     (R2)            ;UPDATE TEST NUMBER
022624  022712  000237                  CMP     #237,(R2)       ;SEQUENCE ERROR?
```

```
7045  022630  001022                    BNE     TST240-10       ;BR TO ERROR HALT ON SEQ ERROR
7046  022632  012700  000001            MOV     #1,R0           ;R0=1
7047  022636  012737  125125  000000    MOV     #125125,@#0     ;LOC. 0 = 125125
7048  022644  005037  177776            CLR     @#PS            ;PS=0
7049  022650  106440                    MTPS    -(R0)           ;TRY MTPS W/MODE 4
7050  022652  022737  000105  177776    CMP     #105,@#PS       ;CHECK DEST. DATA
7051  022660  001404                    BEQ     MTPS4
7052                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7053                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7054                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7055                                                            ;           WHICH FOLLOWS W/ 764           <====
7056  022662  012742  000546            MOV     #546,-(R2)      ;MOVE TO MAILBOX #  *******  546  *******
7057  022666  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7058  022670  000000                    HALT                    ;DEST. DATA INCORRECT
7059  022672  005700          MTPS4:    TST     R0              ;CHECK MODE 4 REGISTER
7060  022674  001404                    BEQ     TST240
7061                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7062                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7063                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7064                                                            ;           WHICH FOLLOWS W/ 756           <====
7065  022676  012742  000547            MOV     #547,-(R2)      ;MOVE TO MAILBOX #  *******  547  *******
7066  022702  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7067  022704  000000                    HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
7068                                                            ;  OR SEQUENCE ERROR
7069
7070                              ;*****************************************************************************************
7071                              ;TEST 240       TEST MTPS MODE 5
7072                              ;*****************************************************************************************
7073  022706  005212          TST240:   INC     (R2)            ;UPDATE TEST NUMBER
7074  022710  022712  000240            CMP     #240,(R2)       ;SEQUENCE ERROR?
7075  022714  001021                    BNE     TST241-10       ;BR TO ERROR HALT ON SEQ ERROR
7076  022716  012700  000404            MOV     #404,R0         ;R0=404
7077  022722  012737  177400  000000    MOV     #177400,@#0     ;LOC. 0=177400
7078  022730  000277                    SCC                     ;SET ALL COND. CODES
7079  022732  106450                    MTPS    @-(R0)          ;TRY MTPS W/MODE 5
7080  022734  005737  177776            TST     @#PS            ;CHECK DEST. DATA.
7081  022740  001404                    BEQ     MTPS5
7082                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7083                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7084                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7085                                                            ;           WHICH FOLLOWS W/ 766           <====
7086  022742  012742  000550            MOV     #550,-(R2)      ;MOVE TO MAILBOX #  *******  550  *******
7087  022746  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7088  022750  000000                    HALT                    ;DESTINATION DATA INCORRECT
7089  022752  022700  000402          MTPS5:    CMP     #402,R0         ;CHECK MODE 5 REGISTER
7090  022756  001404                    BEQ     TST241
7091                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7092                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7093                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7094                                                            ;           WHICH FOLLOWS W/ 757           <====
7095  022760  012742  000551            MOV     #551,-(R2)      ;MOVE TO MAILBOX #  *******  551  *******
7096  022764  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7097  022766  000000                    HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
7098                                                            ;  OR SEQUENCE ERROR
7099
7100                              ;*****************************************************************************************
```



```
7101                              ;TEST 241       TEST MTPS MODE 6
7102                              ;*****************************************************************************************
7103  022770  005212          TST241:   INC     (R2)            ;UPDATE TEST NUMBER
7104  022772  022712  000241            CMP     #241,(R2)       ;SEQUENCE ERROR?
7105  022776  001024                    BNE     TST242-10       ;BR TO ERROR HALT ON SEQ ERROR
7106  023000  012737  052652  000000    MOV     #52652,@#0      ;LOC. 0=52652
7107  023006  012700  000406            MOV     #406,R0         ;R0=406
7108  023012  005037  177776            CLR     @#PS            ;PS=0
7109  023016  106460  177372            MTPS    -106(R0)        ;TRY MTPS W/MODE 6
7110  023022  022737  000252  177776    CMP     #252,@#PS       ;CHECK DEST. DATA
7111  023030  001404                    BEQ     MTPS6
7112                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7113                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7114                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7115                                                            ;           WHICH FOLLOWS W/ 763           <====
7116  023032  012742  000552            MOV     #552,-(R2)      ;MOVE TO MAILBOX #  *******  552  *******
7117  023036  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7118  023040  000000                    HALT                    ;DEST. DATA INCORRECT
7119  023042  022700  000406          MTPS6:    CMP     #406,R0         ;CHECK MODE 6 REGISTER
7120  023046  001404                    BEQ     TST242
7121                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7122                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7123                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7124                                                            ;           WHICH FOLLOWS W/ 754           <====
7125  023050  012742  000553            MOV     #553,-(R2)      ;MOVE TO MAILBOX #  *******  553  *******
7126  023054  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7127  023056  000000                    HALT                    ;MODE 6 REGISTER MODIFIED
7128                                                            ;  OR SEQUENCE ERROR
7129
7130                              ;*****************************************************************************************
7131                              ;TEST 242       TEST MTPS MODE 7
7132                              ;*****************************************************************************************
7133  023060  005212          TST242:   INC     (R2)            ;UPDATE TEST NUMBER
7134  023062  022712  000242            CMP     #242,(R2)       ;SEQUENCE ERROR?
7135  023066  001024                    BNE     TST243-10       ;BR TO ERROR HALT ON SEQ ERROR
7136  023070  012737  052652  000000    MOV     #52652,@#0      ;LOC. 0=52652
7137  023076  012700  000410            MOV     #410,R0         ;R0=410
7138  023102  005037  177776            CLR     @#PS            ;PS=0
7139  023106  106470  177776            MTPS    @-2(R0)         ;TRY MTPS W/MODE 7
7140  023112  022737  000105  177776    CMP     #105,@#PS       ;CHECK DEST. DATA
7141  023120  001404                    BEQ     MTPS7
7142                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7143                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7144                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7145                                                            ;           WHICH FOLLOWS W/ 763           <====
7146  023122  012742  000554            MOV     #554,-(R2)      ;MOVE TO MAILBOX #  *******  554  *******
7147  023126  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7148  023130  000000                    HALT                    ;DESTINATION DATA INCORRECT
7149  023132  022700  000410          MTPS7:    CMP     #410,R0         ;CHECK MODE 7 REGISTER
7150  023136  001404                    BEQ     TST243
7151                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7152                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7153                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7154                                                            ;           WHICH FOLLOWS W/ 754           <====
7155  023140  012742  000555            MOV     #555,-(R2)      ;MOVE TO MAILBOX #  *******  555  *******
7156  023144  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
```

```
 7157  023146  000000                    HALT                    ;MODE 7 REGISTER MODIFIED
 7158                                                            ;  OR SEQUENCE ERROR
 7159
```



```
 7160                    ;*****************************************************************************
 7161                    ;
 7162                    ;       THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
 7163                    ;MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES IS MOVED TO THE
 7164                    ;PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
 7165                    ;BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
 7166                    ;USED TO CHECK PROPER ADDRESSING AND DATA.
 7167                    ;
 7168                    ;*****************************************************************************
 7169                    ;TEST 243       TEST MFPS INSTRUCTION
 7170                    ;*****************************************************************************
 7171  023150  005212    TST243: INC     (R2)            ;UPDATE TEST NUMBER
 7172  023152  022712  000243    CMP     #243,(R2)       ;SEQUENCE ERROR?
 7173  023156  001025            BNE     TST244-10       ;BR TO ERROR HALT ON SEQ ERROR
 7174  023160  012737  000377  177776    MOV     #377,@#PS
 7175  023166  106700            MFPS    R0
 7176  023170  022700  177757          CMP     #177757,R0
 7177  023174  001404            BEQ     MFPS1
 7178
 7179                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7180                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 7181                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 7182  023176  012742  000556                          ;           WHICH FOLLOWS W/ 771           <====
 7183  023202  005242            MOV     #556,-(R2)      ;MOVE TO MAILBOX #  *******  556  *******
 7184  023204  000000            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7185                            HALT                    ;MFPS FAILED
 7186  023206  005000    MFPS1:  CLR     R0
 7187  023210  012737  177777  000000    MOV     #-1,@#0
 7188  023216  005037  177776          CLR     @#PS
 7189  023222  106710            MFPS    (R0)
 7190  023224  105737  000000          TSTB    @#0
 7191  023230  001404            BEQ     TST244
 7192
 7193                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7194                                                    ;           CONDITIONAL BRANCH INST. AND   <====
 7195                                                    ;           REPLACE THE MOVE INSTRUCTION   <====
 7196  023232  012742  000557                          ;           WHICH FOLLOWS W/ 753           <====
 7197  023236  005242            MOV     #557,-(R2)      ;MOVE TO MAILBOX #  *******  557  *******
 7198  023240  000000            INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7199                            HALT                    ;MFPS FAILED
 7200                                                    ;  OR SEQUENCE ERROR
 7201                    ;*****************************************************************************
 7202                    ;TEST 244       TEST MFPS MODE 2
 7203                    ;*****************************************************************************
 7204  023242  005212    TST244: INC     (R2)            ;UPDATE TEST NUMBER
 7205  023244  022712  000244          CMP     #244,(R2)       ;SEQUENCE ERROR?
 7206  023250  001031            BNE     TST245-10       ;BR TO ERROR HALT ON SEQ ERROR
 7207  023252  005000            CLR     R0              ;R0=0
 7208  023254  005010            CLR     (R0)            ;LOC. 0=0
 7209  023256  012737  000377  177776    MOV     #377,@#PS       ;SET PS=357
 7210  023264  106720            MFPS    (R0)+           ;TRY MFPS W/MODE 2
 7211  023266  103003            BCC     MFPS2A          ;BR TO ERROR IF C BIT CLEAR
 7212  023270  102402            BVS     MFPS2A          ;BR TO ERROR IF V BIT SET
 7213  023272  001401            BEQ     MFPS2A          ;BR TO ERROR IF Z BIT SET
 7214  023274  100404            BMI     MFPS2B
 7215                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```

```
 7216
 7217                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7218                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7219  023276                                            ;          WHICH FOLLOWS W/ 766           <====
 7220  023276  012742  000560          MFPS2A:  MOV     #560,-(R2)      ;MOVE TO MAILBOX #  *******  560  *******
 7221  023302  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7222  023304  000000                   HALT                    ;COND. CODES INCORRECT
 7223  023306  022737  000357  000000  MFPS2B:  CMP     #357,@#0        ;CHECK DEST. DATA
 7224  023314  001404                   BEQ     MFPS2C
 7225                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7226                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7227                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7228                                                    ;          WHICH FOLLOWS W/ 756           <====
 7229  023316  012742  000561                   MOV     #561,-(R2)      ;MOVE TO MAILBOX #  *******  561  *******
 7230  023322  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7231  023324  000000                   HALT                    ;DEST. DATA INCORRECT
 7232  023326  022700  000001  MFPS2C:  CMP     #1,R0           ;CHECK MODE Z REGISTER
 7233  023332  001404                   BEQ     TST245
 7234                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7235                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7236                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7237                                                    ;          WHICH FOLLOWS W/ 747           <====
 7238  023334  012742  000562                   MOV     #562,-(R2)      ;MOVE TO MAILBOX #  *******  562  *******
 7239  023340  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7240  023342  000000                   HALT                    ;MODE 2 REGISTER NOT INCREMENTED 1
 7241                                                    ;  OR SEQUENCE ERROR
 7242
 7243                                    ;*****************************************************************************************
 7244                                    ;TEST 245       TEST MFPS MODE 3
 7245                                    ;*****************************************************************************************
 7246  023344  005212          TST245:  INC     (R2)            ;UPDATE TEST NUMBER
 7247  023346  022712  000245           CMP     #245,(R2)       ;SEQUENCE ERROR?
 7248  023352  001033                   BNE     TST246-10       ;BR TO ERROR HALT ON SEQ ERROR
 7249  023354  012700  000406           MOV     #406,R0         ;R0=406
 7250  023360  005037  000000           CLR     @#0             ;LOC. 0=0
 7251  023364  012737  000252  177776   MOV     #252,@#PS       ;PS=252
 7252  023372  106730                   MFPS    @(R0)+          ;TRY MFPS WITH MODE 3
 7253  023374  103403                   BCS     MFPS3A          ;BR TO ERROR IF C-BIT SET
 7254  023376  102402                   BVS     MFPS3A          ;BR TO ERROR IF V-BIT SET
 7255  023400  001401                   BEQ     MFPS3A          ;BR TO ERROR IF Z-BIT SET
 7256  023402  100404                   BMI     MFPS3B
 7257                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7258                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7259                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7260                                                    ;          WHICH FOLLOWS W/ 764           <====
 7261  023404                   MFPS3A:
 7262  023404  012742  000563                   MOV     #563,-(R2)      ;MOVE TO MAILBOX #  *******  563  *******
 7263  023410  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7264  023412  000000                   HALT                    ;CONDITION CODES INCORRECT
 7265  023414  022737  125000  000000  MFPS3B:  CMP     #125000,@#0     ;CHECK DEST. DATA
 7266  023422  001404                   BEQ     MFPS3C
 7267                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7268                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7269                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7270                                                    ;          WHICH FOLLOWS W/ 754           <====
 7271  023424  012742  000564                   MOV     #564,-(R2)      ;MOVE TO MAILBOX #  *******  564  *******
```



```
 7272  023430  005242                   INC     -(R2)
 7273  023432  000000                   HALT                    ;SET MSGTYP TO FATAL ERROR
 7274  023434  020027  000410  MFPS3C:  CMP     R0,#410         ;DEST DATA INCORRECT
 7275  023440  001404                   BEQ     TST246          ;CHECK MODE 3 REGISTER.
 7276                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7277                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7278                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7279                                                    ;          WHICH FOLLOWS W/ 745           <====
 7280  023442  012742  000565                   MOV     #565,-(R2)      ;MOVE TO MAILBOX #  *******  565  *******
 7281  023446  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7282  023450  000000                   HALT                    ;MODE 3 REGISTER NOT INCREMENTED BY 2
 7283                                                    ;  OR SEQUENCE ERROR
 7284
 7285                                    ;*****************************************************************************************
 7286                                    ;TEST 246       TEST MFPS MODE 4
 7287                                    ;*****************************************************************************************
 7288  023452  005212          TST246:  INC     (R2)            ;UPDATE TEST NUMBER
 7289  023454  022712  000246           CMP     #246,(R2)       ;SEQUENCE ERROR?
 7290  023460  001033                   BNE     TST247-10       ;BR TO ERROR HALT ON SEQ ERROR
 7291  023462  012700  000002           MOV     #2,R0           ;R0=2
 7292  023466  005037  000000           CLR     @#0             ;LOC. 0=0
 7293  023472  012737  000125  177776   MOV     #125,@#PS       ;PS=125
 7294  023500  106740                   MFPS    -(R0)           ;TRY MFPS W/MODE 4
 7295  023502  103003                   BCC     MFPS4A          ;BR TO ERROR IF C-BIT CLEAR
 7296  023504  102402                   BVS     MFPS4A          ;BR TO ERROR IF V-BIT SET
 7297  023506  001401                   BEQ     MFPS4A          ;BR TO ERROR IF Z-BIT SET
 7298  023510  100004                   BPL     MFPS4B
 7299                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7300                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7301                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7302                                                    ;          WHICH FOLLOWS W/ 764           <====
 7303  023512                   MFPS4A:
 7304  023512  012742  000566                   MOV     #566,-(R2)      ;MOVE TO MAILBOX #  *******  566  *******
 7305  023516  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7306  023520  000000                   HALT                    ;COND. CODES INCORRECT
 7307  023522  022737  042400  000000  MFPS4B:  CMP     #42400,@#0      ;CHECK DEST. DATA
 7308  023530  001404                   BEQ     MFPS4C
 7309                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7310                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7311                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7312                                                    ;          WHICH FOLLOWS W/ 754           <====
 7313  023532  012742  000567                   MOV     #567,-(R2)      ;MOVE TO MAILBOX #  *******  567  *******
 7314  023536  005242                   INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7315  023540  000000                   HALT                    ;DEST. DATA INCORRECT
 7316  023542  020027  000001  MFPS4C:  CMP     R0,#1           ;CHECK MODE 4 REGISTER
 7317  023546  001404                   BEQ     TST247
 7318
 7319                                                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
 7320                                                    ;          CONDITIONAL BRANCH INST. AND   <====
 7321                                                    ;          REPLACE THE MOVE INSTRUCTION   <====
 7322  023550  012742  000570                   MOV     #570,-(R2)      ;          WHICH FOLLOWS W/ 745           <====
 7323  023554  005242                   INC     -(R2)           ;MOVE TO MAILBOX #  *******  570  *******
 7324  023556  000000                   HALT                    ;SET MSGTYP TO FATAL ERROR
 7325                                                    ;MODE 4 REGISTER NOT DECREMENTED BY 1
 7326                                                    ;  OR SEQUENCE ERROR
 7327                                    ;*****************************************************************************************
```

```
                                        ;TEST 247       TEST MFPS MODE 5
                                        ;*****************************************************************************
                                        TST247: INC     (R2)            ;UPDATE TEST NUMBER
023560  005212                                  CMP     #247,(R2)       ;SEQUENCE ERROR?
023562  022712  000247                          BNE     TST250-10       ;BR TO ERROR HALT ON SEQ ERROR
023566  001033                                  MOV     #410,R0         ;R0=410
023570  012700  000410                          MOV     #-1,@#0         ;LOC. 0=-1
023574  012737  177777  000000                  CLR     @#PS            ;PS=0
023602  005037  177776                          MFPS    @-(R0)          ;TRY MFPS W/MODE 5
023606  106750                                  BCS     MFPS5A          ;BR TO ERROR IF C-BIT SET
023610  103403                                  BVS     MFPS5A          ;BR TO ERROR IF V-BIT SET
023612  102402                                  BMI     MFPS5A          ;BR TO ERROR IF N-BIT SET
023614  100401                                  BEQ     MFPS5B
023616  001404                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 764           <====
023620                                  MFPS5A:
023620  012742  000571                          MOV     #571,-(R2)      ;MOVE TO MAILBOX #  *******  571  *******
023624  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023626  000000                                  HALT                    ;COND. CODES INCORRECT
023630  022737  000377  000000  MFPS5B: CMP     #377,@#0        ;CHECK DEST. DATA
023636  001404                                  BEQ     MFPS5C
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 754           <====
023640  012742  000572                          MOV     #572,-(R2)      ;MOVE TO MAILBOX #  *******  572  *******
023644  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023646  000000                                  HALT                    ;DEST DATA INCORRECT
023650  020027  000406          MFPS5C: CMP     R0,#406         ;CHECK MODE 5 REGISTER
023654  001404                                  BEQ     TST250
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 745           <====
023656  012742  000573                          MOV     #573,-(R2)      ;MOVE TO MAILBOX #  *******  573  *******
023662  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023664  000000                                  HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
                                                                        ;  OR SEQUENCE ERROR

                                        ;*****************************************************************************
                                        ;TEST 250       TEST MFPS MODE 6
                                        ;*****************************************************************************
                                        TST250: INC     (R2)            ;UPDATE TEST NUMBER
023666  005212                                  CMP     #250,(R2)       ;SEQUENCE ERROR?
023670  022712  000250                          BNE     TST251-10       ;BR TO ERROR HALT ON SEQ ERROR
023674  001034                                  MOV     #401,R0         ;R0=410
023676  012700  000401                          CLR     @#0             ;LOC. 0=0
023702  005037  000000                          MOV     #252,@#PS       ;PS=252
023706  012737  000252  177776                  MFPS    -401(R0)        ;TRY MFPS W/MODE 6
023714  106760  177377                          BVS     MFPS6A          ;BR TO ERROR IF V-BIT SET
023720  102403                                  BCS     MFPS6A          ;BR TO ERROR IF C-BIT SET
023722  103402                                  BEQ     MFPS6A          ;BR TO ERROR IF Z-BIT SET
023724  001401                                  BMI     MFPS6B
023726  100404                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
```



```
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 763           <====
023730                                  MFPS6A:
023730  012742  000574                          MOV     #574,-(R2)      ;MOVE TO MAILBOX #  *******  574  *******
023734  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023736  000000                                  HALT                    ;COND. CODES INCORRECT
023740  022737  000252  000000  MFPS6B: CMP     #252,@#0        ;CHECK DEST. DATA
023746  001404                                  BEQ     MFPS6C
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 753           <====
023750  012742  000575                          MOV     #575,-(R2)      ;MOVE TO MAILBOX #  *******  575  *******
023754  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023756  000000                                  HALT                    ;DEST. DATA INCORRECT
023760  022700  000401          MFPS6C: CMP     #401,R0         ;CHECK DEST. REGISTER
023764  001404                                  BEQ     TST251
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 744           <====
023766  012742  000576                          MOV     #576,-(R2)      ;MOVE TO MAILBOX #  *******  576  *******
023772  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
023774  000000                                  HALT                    ;DEST. DATA INCORRECT
                                                                        ;  OR SEQUENCE ERROR

                                        ;*****************************************************************************
                                        ;TEST 251       TEST MFPS MODE 7
                                        ;*****************************************************************************
                                        TST251: INC     (R2)            ;UPDATE TEST NUMBER
023776  005212                                  CMP     #251,(R2)       ;SEQUENCE ERROR?
024000  022712  000251                          BNE     TST252-10       ;BR TO ERROR HALT ON SEQ ERROR
024004  001034                                  MOV     #777,R0         ;R0=777
024006  012700  000777                          CLR     @#0             ;LOC. 0=0
024012  005037  000000                          MOV     #125,@#PS       ;PS=125
024016  012737  000125  177776                  MFPS    @-371(R0)       ;TRY MFPS W/MODE 7
024024  106770  177407                          BVS     MFPS7A          ;BR TO ERROR IF V-BIT SET
024030  102403                                  BCC     MFPS7A          ;BR TO ERROR IF C-BIT SET
024032  103002                                  BEQ     MFPS7A          ;BR TO ERROR IF Z-BIT SET
024034  001401                                  BPL     MFPS7B
024036  100004                                                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 763           <====
024040                                  MFPS7A:
024040  012742  000577                          MOV     #577,-(R2)      ;MOVE TO MAILBOX #  *******  577  *******
024044  005242                                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024046  000000                                  HALT                    ;CONDITION CODE INCORRECT
024050  022737  042400  000000  MFPS7B: CMP     #42400,@#0      ;CHECK DESTINATION DATA
024056  001404                                  BEQ     MFPS7C
                                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                                        ;           WHICH FOLLOWS W/ 753           <====
024060  012742  000600                          MOV     #600,-(R2)      ;MOVE TO MAILBOX #  *******  600  *******
```

```
7440  024064  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7441  024066  000000                    HALT                    ;DEST. DATA INCORRECT
7442  024070  022700  000777  MFPS7C:   CMP     #777,R0         ;CHECK MODE 7 REGISTER
7443  024074  001404                    BEQ     TST252
7444                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7445                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7446                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7447                                                            ;           WHICH FOLLOWS W/ 744           <====
7448  024076  012742  000601                    MOV     #601,-(R2)      ;MOVE TO MAILBOX #  *******  601  *******
7449  024102  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7450  024104  000000                    HALT                    ;MODE 7 REGISTER MODIFIED
7451                                                            ;  OR SEQUENCE ERROR
7452
7453                                ;*************************************************************************************
7454                                ;
7455                                ;       THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
7456                                ;THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
7457                                ;CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
7458                                ;CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 256 (DECIMAL)
7459                                ;PASSES.
7460                                ;
7461                                ;*************************************************************************************
7462                                ;TEST 252       TEST THAT RESET DOES NOT CLEAR PSW
7463                                ;*************************************************************************************
7464  024106  005212  TST252: INC     (R2)            ;UPDATE TEST NUMBER
7465  024110  022712  000252          CMP     #252,(R2)       ;SEQUENCE ERROR?
7466  024114  001014                  BNE     TST253-10       ;BR TO ERROR HALT ON SEQ ERROR
7467  024116  123727  026060  000377        CMPB    @#PASSPT,#377           ;ONLY DUE RESET EVERY 256. PASSES
7468  024124  001014                  BNE     REST            ;BR IF TO SKIP TEST
7469  024126  012737  000357  177776        MOV     #357,@#PS               ;MOV ONES TO PSW
7470  024134  000005                  RESET
7471  024136  022737  000357  177776        CMP     #357,@#PS               ;PSW CORRECT?
7472  024144  001404                  BEQ     TST253
7473                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7474                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7475                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7476                                                            ;           WHICH FOLLOWS W/ 764           <====
7477  024146  012742  000602                    MOV     #602,-(R2)      ;MOVE TO MAILBOX #  *******  602  *******
7478  024152  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7479  024154  000000                    HALT                    ;RESET ALTERED PSW
7480                                                            ;  OR SEQUENCE ERROR
7481  024156                    REST:
7482
7483                                ;*************************************************************************************
7484                                ;
7485                                ;       THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
7486                                ;DATA PATH COMPONENTS WITH USER MODE SET.
7487                                ;
7488                                ;*************************************************************************************
7489                                ;TEST 253       TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
7490                                ;*************************************************************************************
7491  024156  005212  TST253: INC     (R2)            ;UPDATE TEST NUMBER
7492  024160  022712  000253          CMP     #253,(R2)       ;SEQUENCE ERROR?
7493  024164  001014                  BNE     TST254-10       ;BR TO ERROR HALT ON SEQ ERROR
7494  024166  052767  140000  153602        BIS     #USRM,PS                ;SET USER MODE
7495  024174  012706  000001                MOV     #1,R6                   ;SET BIT0
```



```
7496  024200  000241                    CLC                     ;CLEAR C-BIT
7497  024202  006106          USP1:     ROL     R6              ;ROTATE 1 POSITION
7498  024204  103376                    BCC     USP1            ;BR IF NOT ALL DONE
7499  024206  001407                    BEQ     USP1A           ;BR IF NO BITS PICKED
7500  024210  042767  140000  153560        BIC     #USRM,PS                ;CLEAR USER MODE
7501  024216  012742  000603                    MOV     #603,-(R2)      ;MOVE TO MAILBOX #  *******  603  *******
7502  024222  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7503  024224  000000                    HALT                    ;USER MODE R6 PICKED A BIT
7504  024226                    USP1A:
7505
7506                                ;*************************************************************************************
7507                                ;
7508                                ;       THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
7509                                ;AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
7510                                ;OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
7511                                ;OF EACH OTHER.
7512                                ;
7513                                ;*************************************************************************************
7514                                ;TEST 254       TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
7515                                ;*************************************************************************************
7516  024226  005212  TST254: INC     (R2)            ;UPDATE TEST NUMBER
7517  024230  022712  000254          CMP     #254,(R2)       ;SEQUENCE ERROR?
7518  024234  001036                  BNE     USP4-14         ;BR TO ERROR HALT ON SEQ ERROR
7519  024236  052767  140000  153532        BIS     #USRM,PS                ;SET USER MODE
7520  024244  012706  177777                MOV     #-1,R6                  ;SET USER R6 TO ALL ONES
7521  024250  022706  177777                CMP     #-1,R6                  ;READ AND CHECK USER R6
7522  024254  001407                  BEQ     USP2            ;BR IF NO ERROR
7523  024256  042767  140000  153512        BIC     #USRM,PS                ;CLEAR USER MODE
7524  024264  012742  000604                    MOV     #604,-(R2)      ;MOVE TO MAILBOX #  *******  604  *******
7525  024270  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7526  024272  000000                    HALT                    ;USER R6 WILL NOT HOLD ALL ONES
7527  024274  042767  140000  153474  USP2:     BIC     #USRM,PS                ;SET KERNEL MODE
7528  024302  022706  177777                CMP     #-1,R6                  ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
7529  024306  001004                  BNE     USP3
7530                                                            ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
7531                                                            ;           CONDITIONAL BRANCH INST. AND   <====
7532                                                            ;           REPLACE THE MOVE INSTRUCTION   <====
7533                                                            ;           WHICH FOLLOWS W/ 753           <====
7534  024310  012742  000605                    MOV     #605,-(R2)      ;MOVE TO MAILBOX #  *******  605  *******
7535  024314  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7536  024316  000000                    HALT                    ;DUAL ADDRESSING ERROR USER/KERNEL R6
7537  024320  005006          USP3:     CLR     R6                      ;CLEAR KERNEL MODE SP
7538  024322  052767  140000  153446        BIS     #USRM,PS                ;SET USER MODE
7539  024330  022706  177777                CMP     #-1,R6                  ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7540  024334  001404                  BEQ     USP4            ;BR IF NO ERROR
7541  024336  012742  000606                    MOV     #606,-(R2)      ;MOVE TO MAILBOX #  *******  606  *******
7542  024342  005242                    INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
7543  024344  000000                    HALT                    ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7544  024346  012706  000500          USP4:     MOV     #STBOT,R6               ;RESTORE SP USER
7545  024352  042767  140000  153416        BIC     #USRM,PS                ;SET KERNEL MODE
7546  024360  012706  000500                MOV     #STBOT,R6               ;RESTORE SP KERNEL
7547
7548                                ;*************************************************************************************
7549                                ;
7550                                ;       THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
7551                                ;WITH R6 IN MODE 0.
```

```
                                ;
                                ;*****************************************************************************
                                ;TEST 255        TEST MFPI WITH R6 IN MODE 0
                                ;*****************************************************************************
024364  005212                  TST255: INC     (R2)            ;UPDATE TEST NUMBER
024366  022712  000255                  CMP     #255,(R2)       ;SEQUENCE ERROR?
024372  001032                          BNE     TST256-10       ;BR TO ERROR HALT ON SEQ ERROR
024374  012706  000500                  MOV     #STBOT,R6               ;INITIALIZE KERNEL STACK POINTER
024400  012767  140000  153370          MOV     #USRM,PS                ;SET USER MODE,PREVIOUS KERNEL
024406  012706  026424                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
024412  006506                          MFPI    R6                      ;TRY MFPI WITH MODE 0
024414  022767  140000  153354          CMP     #140000,PS              ;CHECK PSW
024422  001407                          BEQ     MFPI0                   ;BR IF NO ERROR
024424  042767  140000  153344          BIC     #USRM,PS                ;CLEAR USER MODE
024432  012742  000607                  MOV     #607,-(R2)      ;MOVE TO MAILBOX #  *******  607  *******
024436  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024440  000000                          HALT                    ;INCORRECT PSW FROM MFPI
024442  022767  000500  001752  MFPI0:  CMP     #STBOT,USTBOT-2         ;CHECK DATA ON STACK
024450  001407                          BEQ     MFPI0A                  ;BR IF NO ERROR
024452  042767  140000  153316          BIC     #USRM,PS                ;CLEAR USER MODE
024460  012742  000610                  MOV     #610,-(R2)      ;MOVE TO MAILBOX #  *******  610  *******
024464  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024466  000000                          HALT                    ;INCORRECT DATA FROM MFPI
024470                          MFPI0A:
                                ;
                                ;*****************************************************************************
                                ;TEST 256        TEST MTPI WITH R6 IN MODE 0
                                ;*****************************************************************************
024470  005212                  TST256: INC     (R2)            ;UPDATE TEST NUMBER
024472  022712  000256                  CMP     #256,(R2)       ;SEQUENCE ERROR?
024476  001033                          BNE     TST257-10       ;BR TO ERROR HALT ON SEQ ERROR
024500  005067  153272                  CLR     PS                      ;SET KERNEL MODE
024504  005006                          CLR     R6                      ;INITIALIZE KERNEL R6
024506  012767  140000  153262          MOV     #USRM,PS                ;SET USER MODE/PREVIOUS KERNEL
024514  012706  026424                  MOV     #USTBOT,R6              ;INITIALIZE USER STACK POINTER
024520  012746  000500                  MOV     #STBOT,-(R6)            ;SET UP TARGET DATA
024524  006606                          MTPI    R6                      ;TRY MODE 0 MTPI
024526  022767  140000  153242          CMP     #USRM,PS                ;CHECK PSW
024534  001407                          BEQ     MTPI0                   ;BR IF NO ERROR
024536  042767  140000  153232          BIC     #USRM,PS                ;CLEAR USER MODE
024544  012742  000611                  MOV     #611,-(R2)      ;MOVE TO MAILBOX #  *******  611  *******
024550  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024552  000000                          HALT                    ;PS INCORRECT FOLLOWING MTPI
024554  005067  153216          MTPI0:  CLR     PS                      ;SET KERNEL MODE
024560  020627  000500                  CMP     R6,#STBOT               ;CHECK TARGET DATA
024564  001404                          BEQ     TST257
                                ;
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                ;           CONDITIONAL BRANCH INST. AND    <====
                                ;           REPLACE THE MOVE INSTRUCTION    <====
                                ;           WHICH FOLLOWS W/ 745            <====
024566  012742  000612                  MOV     #612,-(R2)      ;MOVE TO MAILBOX #  *******  612  *******
024572  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024574  000000                          HALT                    ;DATA INCORRECT FOLLOWING MTPI
                                ;                               ;  OR SEQUENCE ERROR
```



```
                                ;
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE CONTENTS OF THE BRANCH ROM.  THE TEST
                                ;EXECUTES EVERY POSSIBLE BRANCH WITH EVERY POSSIBLE CONDITION
                                ;CODE COMBINATION.
                                ;       THE ROUTINE USES TWO TABLES.  THE BRANCH TABLE HOLDS ALL THE
                                ;POSSIBLE BRANCH INSTRUCTIONS, THE OTHER TABLE (YNTAB) HOLDS BIT MAPS FOR
                                ;EACH BRANCH.  A ONE IN THE BIT MAP INDICATES THAT THE CORRESPONDING
                                ;BRANCH INSTRUCTION SHOULD BRANCH FOR THE CONDITION CODE SETTING WHICH
                                ;CORRESPONDS TO THE BIT POSITION WITHIN THE MAP.  FOR EXAMPLE IF THE LEFT
                                ;MOST BIT IS A ONE THEN THE CORRESPONDING BRANCH INSTRUCTION SHOULD BRANCH
                                ;WHEN THE CONDITION CODES ARE 0.
                                ;       THE ROUTINE CONSISTS OF NESTED LOOPS; THE OUTER LOOP SETS UP
                                ;ALL THE POSSIBLE BRANCH INSTRUCTIONS.  THE INNER LOOP SETS UP EVERY POSSIBLE
                                ;CONDITION CODE FOR EACH BRANCH.
                                ;       THE BIT MAP IS USED TO SET THE ADDRESS LOCATION IN TWO
                                ;JUMP MODE 3 INSTRUCTIONS.  THE ADDRESSES ARE CHANGED TO ALLOW THE
                                ;PROGRAM TO CONTINUE OR JUMP TO AN ERROR ROUTINE DEPENDING UPON
                                ;WHETHER IT HANDLED THE BRANCH INSTRUCTION CORRECTLY.
                                ;       AT ANY ERROR HALT, LOCATION, BRH, HOLDS THE BRANCH INSTRUCTION
                                ;UNDER TEST AND LOCATION, CC, HOLDS THE VALUE OF THE CONDITION CODES
                                ;AT THE TIME THE BRANCH WAS EXECUTED.
                                ;
                                ;*****************************************************************************
                                ;TEST 257        TEST THE BRANCH ROM
                                ;*****************************************************************************
024576  005212                  TST257: INC     (R2)            ;UPDATE TEST NUMBER
024600  022712  000257                  CMP     #257,(R2)       ;SEQUENCE ERROR?
024604  001062                          BNE     ER              ;BR TO ERROR HALT ON SEQ ERROR
024606  012700  026214          SETUP:  MOV     #BRTAB,R0       ;INITIALIZE BRANCH TABLE POINTER
024612  012704  026252                  MOV     #YNTAB,R4       ;INITIALIZE YES/NO BRANCH MAP POINTER
024616  012767  000017  000142          MOV     #15.,BRCT       ;INITIALIZE BRANCH TABLE COUNT
024624  012067  000110          SETBR:  MOV     (R0)+,BRH       ;GET NEXT BRANCH INST.
024630  012401                          MOV     (R4)+,R1        ;GET NEXT BRANCH MAP
024632  012767  177777  000074          MOV     #-1,CC          ;INITIALIZE CONDITION CODE VALUE
024640  012703  000020                  MOV     #16.,R3         ;INITIALIZE CONDITION CODE COUNT
024644  005267  000064          SETCC:  INC     CC              ;SET FOR NEXT CC VALUE
024650  032701  100000                  BIT     #100000,R1      ;SEE IF SHOULD BR W/ THESE CC'S
024654  013705  177776                  MOV     @#177776,R5     ;SIMULATE A JNE
024660  042705  177773                  BIC     #177773,R5      ;       (JUMP NOT EQUAL)
024664  000165  024670                  JMP     .+4(R5)         ;       TO SET2BR
024670  000167  000020                  JMP     SET2BR
024674  012767  024770  000042          MOV     #CONT,NBR       ;SET TO CONTINUE IF NO BRANCH
024702  012767  024752  000040          MOV     #ER,YBR         ;SET TO REPORT ERROR IF BRANCH
024710  000167  000014                  JMP     AROUND          ;GO AROUND OPPOSITE CONDITION
024714  012767  024752  000022  SET2BR: MOV     #ER,NBR         ;SET TO REPORT ERROR IF NO BRANCH
024722  012767  024770  000020          MOV     #CONT,YBR       ;SET TO CONTINUE IF BRANCH
024730  006101                  AROUND: ROL     R1              ;UPDATE BIT MAP
                                ;
024732  012737                          MOV     (PC)+,@(PC)+    ;SET CONDITION CODE
024734  000000                  CC:     0                       ;NEW CC VALUE GOES HERE
024736  177776                          177776
024740  000000                  BRH:    0                       ;BRANCH INST. GOES HERE
024742  000137                          JMP     @(PC)+          ;THIS JUMP IF NO BRANCH
024744  000000                  NBR:    0                       ;WHERE TO GO IF NO BRANCH OCCURS
```

```
                                        JMP     @(PC)+          ;THIS JUMP IF BRANCH OCCURS
024746  000137
024750  000000                  YBR:    0                       ;WHERE TO GO IF BRANCH OCCURS
024752  012702  000304          ER:     MOV     #$TESTN,R2      ;RESTORE POINTER
024756  012742  000613                  MOV     #613,-(R2)      ;MOVE TO MAILBOX #  *******  613  *******
024762  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
024764  000000                          HALT                    ;
024766  000000                  BRCT:   0
024770  005303                  CONT:   DEC     R3              ;CC'S DONE?
024772  013705  177776                  MOV     @#177776,R5     ;SIMULATE A JNE
024776  042705  177773                  BIC     #177773,R5      ;       (JUMP NOT EQUAL)
025002  000165  025006                  JMP     .+4(R5)         ;       TO SETCC
025006  000167  177632                  JMP     SETCC
025012  005367  177750                  DEC     BRCT            ;BR'S DONE?
025016  013705  177776                  MOV     @#177776,R5     ;SIMULATE A JNE
025022  042705  177773                  BIC     #177773,R5      ;       (JUMP NOT EQUAL)
025026  000165  025032                  JMP     .+4(R5)         ;       TO SETBR
025032  000167  177566                  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
                        ;*****************************************************************************************
025036  005212          TST260: INC     (R2)            ;UPDATE TEST NUMBER
025040  022712  000260          CMP     #260,(R2)       ;SEQUENCE ERROR?
025044  001052                  BNE     DAERR           ;BR TO ERROR HALT ON SEQ ERROR
025046  005000          BITCLR: CLR     R0              ;INITIALIZE ALL REGISTERS
025050  005001                  CLR     R1
025052  005002                  CLR     R2
025054  005003                  CLR     R3
025056  005004                  CLR     R4
025060  005005                  CLR     R5
025062  005006                  CLR     R6
025064  052700  000001  BITSET: BIS     #1,R0           ;SET R0=1
025070  052701  000002          BIS     #2,R1           ;R1=2
025074  052702  000004          BIS     #4,R2           ;R2=4
025100  052703  000010          BIS     #10,R3          ;R3=10
025104  052704  000020          BIS     #20,R4          ;R4=20
025110  052705  000040          BIS     #40,R5          ;R5=40
025114  052706  000100          BIS     #100,R6         ;R6=100
025120  022706  000100  BITCHK: CMP     #100,R6         ;TEST THAT NO DUAL ADDRESSING OCCURRED
025124  001022                  BNE     DAERR           ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
025126  022705  000040          CMP     #40,R5
025132  001017                  BNE     DAERR
025134  022704  000020          CMP     #20,R4
025140  001014                  BNE     DAERR
025142  022703  000010          CMP     #10,R3
025146  001011                  BNE     DAERR
025150  022702  000004          CMP     #4,R2
025154  001006                  BNE     DAERR
025156  022701  000002          CMP     #2,R1
025162  001003                  BNE     DAERR
025164  022700  000001          CMP     #1,R0
025170  001404                  BEQ     BITCON
                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS   <====
                                                        ;           CONDITIONAL BRANCH INST. AND   <====
                                                        ;           REPLACE THE MOVE INSTRUCTION   <====
                                                        ;           WHICH FOLLOWS W/ 726           <====
025172                  DAERR:
025172  012742  000614          MOV     #614,-(R2)      ;MOVE TO MAILBOX #  *******  614  *******
025176  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025200  000000                  HALT                    ;DUAL ADDRESSING ERROR
025202  012702  000304  BITCON: MOV     #$TESTN,R2      ;RESTORE POINTER
025206  012706  000500          MOV     #STBOT,R6       ;RESET STACK
```

```
                        ;*****************************************************************************
                        ;
                        ;       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
                        ;*****************************************************************************
025212  005212          TST261: INC     (R2)            ;UPDATE TEST NUMBER
025214  022712  000261          CMP     #261,(R2)       ;SEQUENCE ERROR?
025220  001012                  BNE     BTERR           ;BR TO ERROR HALT ON SEQ ERROR
025222  052737  170357  177776          BIS     #170357,@#PS    ;SET ALL POSSIBLE BITS IN PSW
025230  105037  177776          CLRB    @#PS            ;CLR PR LEVEL AND CC'S
025234  013700  177776          MOV     @#PS,R0         ;COPY CONTENTS OF PSW
025240  032700  170000          BIT     #170000,R0      ;TEST THAT UPPER BYTE IS UNAFFECTED
025244  001006                  BNE     BTCON           ;CONTINUE IF OK
025246  005037  177776  BTERR:  CLR     @#PS            ;RETURN TO KERNEL MODE
025252  012742  000615          MOV     #615,-(R2)      ;MOVE TO MAILBOX #  *******  615  *******
025256  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025260  000000                  HALT                    ;BYTE INSTRUCTION ALTERED PSW
025262  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
                        ;*****************************************************************************
025266  005212          TST262: INC     (R2)            ;UPDATE TEST NUMBER
025270  022712  000262          CMP     #262,(R2)       ;SEQUENCE ERROR?
025274  001010                  BNE     TST263-10       ;BR TO ERROR HALT ON SEQ ERROR
025276  000277                  SCC
025300  000252                  +CLN!CLV                ;CC=0101
025302  000167  000000          JMP     JMPT            ;JUMP TO TEST PSW
025306  100403          JMPT:   BMI     JMPERR          ;BR TO ERROR HALT IF N-BIT IS SET
025310  001002                  BNE     JMPERR          ;BR TO ERROR HALT IF Z-BIT IS CLEAR
025312  102401                  BVS     JMPERR          ;BR TO ERROR HALT IF V-BIT IF SET
025314  103404                  BCS     TST263

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 770            <====
025316                  JMPERR:
025316  012742  000616          MOV     #616,-(R2)      ;MOVE TO MAILBOX #  *******  616  *******
025322  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025324  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
                        ;*****************************************************************************
025326  005212          TST263: INC     (R2)            ;UPDATE TEST NUMBER
025330  022712  000263          CMP     #263,(R2)       ;SEQUENCE ERROR?
025334  001062                  BNE     CCERR           ;BR TO ERROR HALT ON SEQ ERROR
025336  012767  000240  000024          MOV     #240,CC1        ;INITIALIZE CLR CC INSTRUCTION CODES
025344  012767  000017  000032          MOV     #17,CC2         ;INITIALIZE OCTAL MAP
025352  012767  000261  000102          MOV     #261,SC3        ;INITIALIZE SET CC INSTRUCTION CODES
025360  012767  000001  000110          MOV     #1,SC4          ;INITIALIZE OCTAL MAP
025366  000277          CLRCD:  SCC                     ;SET ALL CONDITION CODES
025370  000000          CC1:    0                       ;CONDITION CODE INSTRUCTION
025372  013704  177776          MOV     @#PS,R4         ;COPY THE PSW
025376  042704  177760          BIC     #177760,R4      ;ISOLATE CONDITION CODES
025402  022704                  CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE CLEARED
025404  000000          CC2:    0                       ;OCTAL REPRESENTATION OF CC'S
025406  001404                  BEQ     CON1

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
                                                        ;           WHICH FOLLOWS W/ 753            <====
025410  012742  000617          MOV     #617,-(R2)      ;MOVE TO MAILBOX #  *******  617  *******
025414  005242                  INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025416  000000                  HALT                    ;CLEAR CC INSTRUCTION FAILED
025420  005367  177760  CON1:   DEC     CC2             ;SET NEXT OCTAL MAP OF CC'S
025424  005267  177740          INC     CC1             ;GET NEXT CLEAR CC INSTRUCTION
025430  026727  177734  000257          CMP     CC1,#257        ;TEST FOR CCC INSTRUCTION
025436  003753                  BLE     CLRCD           ;GO TEST NEXT INSTRUCTION IF NOT FOUND
025440  026727  177724  000260          CMP     CC1,#260        ;CHECK FOR NOP=260
025446  001004                  BNE     SETCD           ;GO TEST SET CC INSTRUCTIONS
025450  012767  000017  177726          MOV     #17,CC2         ;SET OCTAL MAP TO TEST NOP
025456  000743                  BR      CLRCD           ;GO TEST NOP
025460  000257          SETCD:  CCC                     ;CLEAR ALL CONDITION CODES
025462  000000          SC3:    0                       ;CONDITION CODE INSTRUCTION
025464  013704  177776          MOV     @#PS,R4         ;COV PSW
025470  042704  177760          BIC     #177760,R4      ;CLEAR AWAY UNWANTED BITS
025474  022704                  CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE SET
025476  000000          SC4:    0                       ;OCTAL REPRESENTATION OF CC'S
025500  001404                  BEQ     CON2

                                                        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS    <====
                                                        ;           CONDITIONAL BRANCH INST. AND    <====
                                                        ;           REPLACE THE MOVE INSTRUCTION    <====
```

```
                                                ;          WHICH FOLLOWS W/ 716          <====
025502                          CCERR:
025502  012742  000620                  MOV     #620,-(R2)      ;MOVE TO MAILBOX #  *******  620  *******
025506  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
025510  000000                          HALT                    ;SET CC FAILED OR SEQUENCE ERROR
025512  005267  177760          CON2:   INC     SC4             ;SET NEXT OCTAL MAP
025516  005267  177740                  INC     SC3             ;PREPARE NEXT SET CC INSTRUCTION
025522  026727  177734  000277          CMP     SC3,#277        ;FINISHED?
025530  003753                          BLE     SETCD           ;BR IF NO
```



```
                                ;*****************************************************************************
                                ;TEST 264       END OF PASS SEQUENCE
                                ;*****************************************************************************
025532  005212                  TST264: INC     (R2)            ;UPDATE TEST NUMBER
025534  022712  000264                  CMP     #264,(R2)       ;SEQUENCE ERROR?
025540  001143                          BNE     EOP1            ;BR TO ERROR HALT ON SEQ ERROR
025542  105267  000312                  INCB    PASSPT          ;SHOULD PRINT THIS PASS?
025546  001136                          BNE     GOAGIN          ;NO
025550  005237  000306                  INC     @#$PASS
025554  132767  000040  152537          BITB    #40,$ENVM       ;WILL APT ALLOW PRINTING?
025562  001120                          BNE     ACT             ;NO
025564  023727  000042  026034          CMP     @#42,#$ENDAD    ;UNDER ACT AUTO ACCEPT?
025572  001514                          BEQ     ACT             ;IF SO SKIP PRINTOUT
025574  023727  000306  000001          CMP     @#$PASS,#1      ;IS THIS 1ST PASS?
025602  001004                          BNE     1$
025604  012700  026066                  MOV     #TITLE,R0       ;THEN PRINT TITLE
025610  004737  025676                  JSR     PC,@#WAIT
025614  012700  026140          1$:     MOV     #MSG,R0         ;NOW PRINT END PASS
025620  004737  025676                  JSR     PC,@#WAIT
025624  012700  026212                  MOV     #BUFF,R0        ;SET UP TO BUILD EOP#
025630  112740  000377                  MOVB    #377,-(R0)      ;MOV TERM INTO BOT OF PSNUM
025634  112740  000000                  MOVB    #0,-(R0)        ;MOVE THREE
025640  112740  000000                  MOVB    #0,-(R0)        ;NULL BYTES
025644  112740  000000                  MOVB    #0,-(R0)        ;ON TOP OF TERMINATOR
025650  004737  025722                  JSR     PC,@#BUILD      ;GO BUILD ASCII NUMBER
025654  112740  000000                  MOVB    #0,-(R0)        ;MOVE THREE
025660  112740  000000                  MOVB    #0,-(R0)        ;NULL BYTES
025664  112740  000000                  MOVB    #0,-(R0)        ;ON TOP OF ASCII NUMBER
025670  004737  025676                  JSR PC,@#WAIT           ;GO PRINT PSNUM (PASSNUMBER)
025674  000453                          BR      ACT             ;SERVICE ACT
                                ;
025676  105737  177564          WAIT:   TSTB    @#TPS           ;ROUTINE TO PRINT MSG
025702  100375                          BPL     WAIT            ;WAIT FOR TTY READY
025704  121027  000377                  CMPB    (R0),#377       ;CHECK FOR TERMINATOR
025710  001403                          BEQ     1$
025712  112037  177566                  MOVB    (R0)+,@#TPB     ;NOT TERM, PRINT CHAR
025716  000767                          BR      WAIT            ;GET NEXT CHARACTER
025720  000207                  1$:     RTS     PC              ;CHAR STRING DONE, RETURN
                                ;
025722  013737  000306  026062  BUILD:  MOV     @#$PASS,@#OCTPSS;ROUTINE TO CONV OCTAL TO ASCII
025730  012737  000060  026064  1$:     MOV     #60,@#ASCPSS    ;MOVE ZERO, ASCII FORMAT
025736  006237  026062                  ASR     @#OCTPSS        ;MOVE LOWEST BIT INTO CARRY
025742  103004                          BCC     2$              ;CHECK CARRY
025744  062737  000001  026064          ADD     #1,@#ASCPSS     ;AND ADD VALUE TO ZERO
025752  000241                          CLC                     ;CLEAR CARRY
025754  006237  026062          2$:     ASR     @#OCTPSS        ;REPEAT FOR 2ND BIT
025760  103004                          BCC     3$
025762  062737  000002  026064          ADD     #2,@#ASCPSS
025770  000241                          CLC
025772  006237  026062          3$:     ASR     @#OCTPSS        ;REPEAT FOR 3RD BIT
026000  103004                          BCC     4$
026002  062737  000004  026064          ADD     #4,@#ASCPSS
026006  000241                          CLC
026010  113740  026064          4$:     MOVB    @#ASCPSS,-(R0)  ;STORE ASCII DIGIT
026014  005737  026062                  TST     @#OCTPSS        ;CHECK FOR MORE BITS
026020  001343                          BNE     1$              ;REPEAT UNTIL OCTPSS=0
```

```
 7904  026022  000207                          RTS     PC              ;THEN RETURN
 7905
 7906  026024  013700  000042          ACT:    MOV     @#42,R0         ;CHECK ACT
 7907  026030  001405                          BEQ     GOAGIN          ;KEEP GOING
 7908  026032  000005                          RESET
 7909  026034  004710                  $ENDAD: JSR     PC,(R0)         ;ACT HOOKS
 7910  026036  000240                          NOP
 7911  026040  000240                          NOP
 7912  026042  000240                          NOP
 7913  026044  000167  152452          GOAGIN: JMP     RESTRT          ;DO NEXT PASS
 7914  026050                          EOP1:
 7915  026050  012742  000621                  MOV     #621,-(R2)      ;MOVE TO MAILBOX #  *******  621  *******
 7916  026054  005242                          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 7917  026056  000000                          HALT                    ;SEQUENCE ERROR
 7918  026060  177777          PASSPT: -1
 7919  026062  000000          OCTPSS: .WORD   0               ;PASSCOUNT, OCTAL, STORED HERE
 7920  026064  000000          ASCPSS: .WORD   0               ;PASSCOUNT, ASCII, BUILT  HERE
 7921  026066  005015  000000  000000  TITLE:  .ASCII  <15><12><0><0><0><0><0><0><0>.CFKAAC0 11/34 BSC INST TST.<0><0><0><0><0><0><0>
 7922  026074  000000  043103  040513
 7923  026102  041501  020060  030461
 7924  026110  031457  020064  051502
 7925  026116  020103  047111  052123
 7926  026124  052040  052123  000000
 7927  026132  000000  000000  177400
 7928                                  .EVEN
 7929  026140  005015  000000  000000  MSG:    .ASCII  <15><12><0><0><0><0><0><0><0>.END PASS .<0><0><0><0><0><0><0><377>
 7930  026146  000000  047105  020104
 7931  026154  040520  051523  000040
 7932  026162  000000  000000  177400
 7933                                  .EVEN
 7934                          ;*********************************************
 7935                          ;THESE ARE A UNIT, INSERT NO CODE BETWEEN THEM  *
 7936  026170  000000  000000  000000  PSNUM:  .WORD   0,0,0             ;               *
 7937  026176  000000  000000  000000          .WORD   0,0,0             ;               *
 7938  026204  000000  000000  000000          .WORD   0,0,0             ;               *
 7939  026212  000000                  BUFF:   .WORD   0                 ;               *
 7940                          ;*********************************************
 7941
```



```
 7942  026214  000402                  BRTAB:  BR      .+6
 7943  026216  001002                          BNE     .+6
 7944  026220  001402                          BEQ     .+6
 7945  026222  002002                          BGE     .+6
 7946  026224  002402                          BLT     .+6
 7947  026226  003002                          BGT     .+6
 7948  026230  003402                          BLE     .+6
 7949  026232  100002                          BPL     .+6
 7950  026234  100402                          BMI     .+6
 7951  026236  101002                          BHI     .+6
 7952  026240  101402                          BLOS    .+6
 7953  026242  102002                          BVC     .+6
 7954  026244  102402                          BVS     .+6
 7955  026246  103002                          BCC     .+6
 7956  026250  103402                          BCS     .+6             ;SAME AS BLO
 7957
 7958          000002                  .RADIX  2
 7959  026252  177777                  YNTAB:  1111111111111111        ;BR
 7960  026254  170360                          1111000011110000        ;BNE:  Z=0
 7961  026256  007417                          0000111100001111        ;BEQ:  Z=1
 7962  026260  146063                          1100110000110011        ;BGE:  N XOR V =0
 7963  026262  031714                          0011001111001100        ;BLT:  N XOR V =1
 7964  026264  140060                          1100000000110000        ;BGT:  Z+(N XOR V) =0
 7965  026266  037717                          0011111111001111        ;BLE:  Z+(N XOR V) =1
 7966  026270  177400                          1111111100000000        ;BPL:  N=0
 7967  026272  000377                          0000000011111111        ;BMI:  N=1
 7968  026274  120240                          1010000010100000        ;BHI:  C+Z=0
 7969  026276  057537                          0101111101011111        ;BLOS: C+Z=1
 7970  026300  146314                          1100110011001100        ;BVC:  V=0
 7971  026302  031463                          0011001100110011        ;BVS:  V=1
 7972  026304  125252                          1010101010101010        ;BCC:  C=0
 7973  026306  052525                          0101010101010101        ;BCS:  C=1
 7974          000010                  .RADIX  8
 7975
 7976  026310  012737  026320  000024  PWRDN:  MOV     #PWRUP,@#24             ;SET UP FOR A POWER UP
 7977  026316  000000                          HALT
 7978
 7979  026320  012737  026310  000024  PWRUP:  MOV     #PWRDN,@#24             ;SET UP FOR A POWER FAIL
 7980  026326  012706  000500                  MOV     #STBOT,R6               ;SET UP STACK POINTER
 7981  026332  132767  000040  151761          BITB    #40,$ENVM               ;SHOULD PRINT?
 7982  026340  001010                          BNE     PWR2                    ;IF NOT: BR
 7983  026342  012700  026366                  MOV     #PFMES,R0               ;GET POWER FAIL MESSG.
 7984  026346  105737  177564          WATE:   TSTB    @#TPS                   ;TTY READY?
 7985  026352  100375                          BPL     WATE                    ;IF NOT: BR
 7986  026354  112037  177566                  MOVB    (R0)+,@#TPB             ;PRINT NEXT CHAR.
 7987  026360  001372                          BNE     WATE                    ;IF NOT DONE:  BR
 7988  026362  000137  000500          PWR2:   JMP     @#START                 ;START PROGRAM AGAIN
 7989
 7990  026366  006412  047520  042527  PFMES:  .ASCIZ  <12><15>.POWER FAILURE.<12><15>
 7991  026374  020122  040506  046111
 7992  026402  051125  005105  000015
 7993                                  .EVEN
 7994  026410  000006                  .BLKW   6
 7995  026424                          USTROT:
 7996                          ;*****************************************************************************************
 7997                          ; THE FOLLOWING ARE SPECIAL CPU TRAP
```

```
 7998                                   ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
 7999                                   ;
 8000                                   ;*****************************************************************************
 8001
 8002  026424                           TO4:
 8003  026424  012742  000622                 MOV     #622,-(R2)      ;MOVE TO MAILBOX #  *******  622  *******
 8004  026430  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8005  026432  000000                         HALT                    ;TRAPPED THRU LOC. 4
 8006  026434                           TO10:
 8007  026434  012742  000623                 MOV     #623,-(R2)      ;MOVE TO MAILBOX #  *******  623  *******
 8008  026440  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8009  026442  000000                         HALT                    ;TRAPPED THRU LOC. 10
 8010  026444                           TO14:
 8011  026444  012742  000624                 MOV     #624,-(R2)      ;MOVE TO MAILBOX #  *******  624  *******
 8012  026450  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8013  026452  000000                         HALT                    ;TRAPPED THRU LOC. 14
 8014  026454                           TO30:
 8015  026454  012742  000625                 MOV     #625,-(R2)      ;MOVE TO MAILBOX #  *******  625  *******
 8016  026460  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8017  026462  000000                         HALT                    ;TRAPPED THRU LOC. 30
 8018  026464                           TO34:
 8019  026464  012742  000626                 MOV     #626,-(R2)      ;MOVE TO MAILBOX #  *******  626  *******
 8020  026470  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8021  026472  000000                         HALT                    ;TRAPPED THRU LOC. 34
 8022  026474                           TO114:
 8023  026474  012742  000627                 MOV     #627,-(R2)      ;MOVE TO MAILBOX #  *******  627  *******
 8024  026500  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8025  026502  000000                         HALT                    ;TRAPPED THRU LOC. 114
 8026  026504                           TO244:
 8027  026504  012742  000630                 MOV     #630,-(R2)      ;MOVE TO MAILBOX #  *******  630  *******
 8028  026510  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8029  026512  000000                         HALT                    ;TRAPPED THRU LOC. 244
 8030  026514                           TO250:
 8031  026514  012742  000631                 MOV     #631,-(R2)      ;MOVE TO MAILBOX #  *******  631  *******
 8032  026520  005242                         INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
 8033  026522  000000                         HALT                    ;TRAPPED THRU LOC. 250
 8034          000001                 .END
```



```
ABASE = 000000              28
ACDW1 = 000000              28
ACDW2 = 000000              28
ACPUOP= 000000              28      43
ACT     026024            7858    7860    7877    7906#
ADC1    020054            6088    6089    6095#
ADC2    020064            6090    6099#
ADC3    020104            6103    6104    6110#
ADC4    020114            6105    6114#
ADC5    020132            6117    6118    6119    6125#
ADDW0 = 000000              28
ADDW1 = 000000              28
ADDW10= 000000              28
ADDW11= 000000              28
ADDW12= 000000              28
ADDW13= 000000              28
ADDW14= 000000              28
ADDW15= 000000              28
ADDW2 = 000000              28
ADDW3 = 000000              28
ADDW4 = 000000              28
ADDW5 = 000000              28
ADDW6 = 000000              28
ADDW7 = 000000              28
ADDW8 = 000000              28
ADDW9 = 000000              28
ADD1    017670            6010    6011    6017#
ADD2    017700            6012    6021#
ADD3    017714            6024    6025    6031#
ADD4    017724            6026    6035#
ADD5    017742            6038    6039    6045#
ADD6    017752            6040    6049#
ADD7    017764            6050    6051    6057#
ADD8    017774            6052    6061#
ADD9    020014            6064    6065    6066    6072#
ADEVCT= 000000              28      34
ADEVM = 000000              28
AENV  = 000000              28      39
AENVM = 000000              28      40
AFATAL= 000000              28      31
AMADR1= 000000              28
AMADR2= 000000              28
AMADR3= 000000              28
AMADR4= 000000              28
AMAMS1= 000000              28
AMAMS2= 000000              28
AMAMS3= 000000              28
AMAMS4= 000000              28
AMSCAD= 000000              28      36
AMSGLG= 000000              28      37
AMSGTY= 000000              28      30
AMTYP1= 000000              28
AMTYP2= 000000              28
AMTYP3= 000000              28
AMTYP4= 000000              28
APASS = 000000              28      33
```

```
APRIOR= 000000    28
AROUND  024730  7652    7655#
ASCPSS  026064  7888*   7891*   7895*   7899*   7901    7920#
ASL1    021370  6590    6591    6592    6598#
ASL2    021400  6593    6602#
ASL3    021416  6605    6606    6607    6613#
ASL4    021426  6608    6617#
ASL5    021442  6620    6621    6627#
ASL6    021452  6622    6631#
ASL7    021476  6634    6635    6636    6637    6644#
ASR1    021540  6659    6660    6661    6667#
ASR2    021550  6662    6671#
ASR3    021572  6675    6676    6677    6683#
ASR4    021602  6678    6687#
ASR5    021616  6690    6691    6692    6698#
ASR6    021626  6693    6702#
ASR7    021656  6706    6707    6708    6709    6716#
ASWREG= 000000    28      41
ATESTN= 000000    28      32
AUNIT = 000000    28      35
AUSWR = 000000    28      42
AVECT1= 000000    28
AVECT2= 000000    28
BIC1    017020  5687    5688    5694#
BIC2    017030  5689    5698#
BIC3    017046  5701    5702    5708#
BIS1    017110  5723    5724    5725    5731#
BIS2    017120  5726    5735#
BIS3    017140  5738    5739    5740    5746#
BITCHK  025120  7708#
BITCLR  025046  7694#
BITCON  025202  7721    7730#
BITSET  025064  7701#
BIT1    016730  5650    5651    5657#
BIT2    016740  5652    5662#
BIT3    016756  5665    5666    5672#
BRCT    024766  7639*   7669#   7675*
BRC1    003040  1039    1045#
BRC2    003050  1040    1050#
BRC3    003060  1052    1058#
BRH     024740  7640*   7660#
BRN1    002720  945     951#
BRN2    002730  946     956#
BRN3    002740  958     964#
BRTAB   026214  7637    7942#
BRV1    002770  992     998#
BRV2    003000  993     1003#
BRV3    003010  1005    1011#
BRZ1    002650  898     904#
BRZ2    002660  899     909#
BRZ3    002670  911     917#
BR1     000572  134     140#
BR2     000602  135     144#
BR3     000614  145     153#
BR4     000622  154     160#
BR5     000632  155     164#
```



```
BTCON   025262  7748    7753#
BTERR   025246  7743    7749#
BUFF    026212  7867    7939#
BUILD   025722  7872    7887#
CC      024734  7642*   7644*   7658#
CCERR   025502  7802    7840#
CC1     025370  7803*   7808#   7822*   7823    7825
CC2     025404  7804*   7812#   7821*   7827*
CLRCD   025366  7807#   7824    7828
CLR1    017456  5903    5904    5905    5911#
CMP1    020320  6207    6208    6214#
CMP2    020330  6209    6218#   6229#
CMP3    020352  6222    6223
CMP4    020362  6224    6233#
CMP5    020406  6237    6238    6239    6245#
CMP6    020416  6240    6249#
CMP7    020436  6252    6253    6259#
COM1    020476  6275    6276    6282#
CONT    024770  7650    7654    7670#
CON1    025420  7813    7821#
CON2    025512  7835    7844#
DAERR   025172  7693    7709    7711    7713    7715    7717    7719    7726#
DEC1    017316  5826    5827    5828    5834#
DEC2    017326  5829    5838#   5848#
DEC3    017342  5841    5842
DEC4    017352  5843    5852#   5862#
DEC5    017366  5855    5856
DEC6    017376  5857    5866#
DEC7    017420  5870    5871    5872    5878#
DNMB0A  010520  3459    3460    3461    3467#
DNMB0B  010530  3462    3471#
DNMB2A  010756  3569    3570    3571    3577#
DNMB2B  010766  3572    3581#
DNMB2C  011002  3582    3590#
DNMB2D  011016  3592    3593    3599#
DNMB2E  011026  3594    3603#
DNMB2F  011044  3605    3613#
DNMB3A  011126  3639    3640    3641    3647#
DNMB3B  011136  3642    3651#
DNMB3C  011154  3652    3660#
DNMB3D  011172  3663    3664    3670#
DNMB3E  011202  3665    3674#
DNMB4A  011372  3743    3744    3750#
DNMB4B  011402  3745    3754#
DNMB4C  011420  3755    3763#
DNMB4D  011430  3764    3770#
DNMB4E  011440  3765    3774#
DNMB4F  011454  3775    3783#
DNM03A  007612  3068    3069    3070    3076#
DNM03B  007622  3071    3080#
DNM03C  007632  3081    3088#
DNM1    007464  3002    3010#
DNM1A   010576  3492    3493    3494    3500#
DNM1B   010606  3495    3504#
DNM2    007500  3011    3019#
DNM2A   010654  3525    3526    3532#
```

```
DNM2B   010664          3527    3536#
DNM2C   010672          3543#
DNM2D   010702          3538    3547#
DNM3    007516          3021    3029#
DNM4    007540          3033    3041#
DNM4A   011264          3698    3699    3700    3706#
DNM4B   011274          3701    3710#
DNM4C   011310          3711    3719#
DNM5A   011534          3806    3807    3808    3814#
DNM5B   011544          3809    3818#
DNM5C   011562          3819    3827#
DNM6A   011642          3849    3850    3851    3857#
DNM6B   011652          3852    3861#
DNM6C   011670          3862    3870#
DNM7A   011752          3892    3893    3894    3900#
DNM7B   011762          3895    3904#
DNM7C   012000          3905    3913#
DOPB2A  010224          3312    3320#
DOPB2B  010302          3350    3358#
DOP0A   007226          2886    2894#
DOP0B   007252          2899    2907#
DOP0C   007272          2910    2918#
DOP0D   007322          2925    2933#
DOP03A  007400          2963    2964    2965    2971#
DOP03B  007410          2966    2975#
DOP1    010036          3204    3212#
DOP2    010150          3272    3280#
DOP4    013452          4482    4494#
DOP5    013536          4520    4532#
EOP1    026050          7853    7914#
ER      024752          7636    7651    7653    7665#
GOAGIN  026044          7855    7907    7913#
IJMP    016034          5376    5379#
IJMP4   015606          5301    5304#   5318
IJMP5   015776          5362    5365#
INC1    017200          5773    5774    5780#
INC2    017210          5775    5784#
INC3    017232          5788    5789    5790    5796#
INC4    017242          5791    5801#
INC5    017256          5804    5805    5811#
JMPCK   016036          5279    5379    5381#
JMPERR  025316          7770    7771    7772    7778#
JMPSEQ  016056          5280*   5292    5302*   5306    5315*   5327    5337*   5340    5350*   5353    5363*   5367    5377*
                        5381    5391#
JMPT    025306          7769    7770#
JMP2    015610          5281    5306#
JMP2A   015626          5307    5315#
JMP3    015536          5283#   5316
JMP3A   015554          5284    5292#
JMP3B   015574          5293    5301#
JMP4    015640          5304    5318#
JMP4A   015656          5319    5327#
JMP4B   015676          5328    5336#
JMP5    015744          5336    5353#
JMP5A   015764          5354    5362#
JMP6    015710          5340#   5365
```



```
JMP6A   015730          5341    5349#
JMP7    016000          5349    5367#
JMP7A   016020          5368    5376#
JSRCK   016510          5540    5543#
JSRCKA  016504          5537    5540#
JSRCK1  016526          5416    5544    5551#
JSRSEQ  016506          5420*   5431    5449*   5452    5470*   5474    5486*   5490    5504*   5506    5520*   5524    5536*
                        5541#   5543
JSR0    016072          5414    5416#
JSR1    016076          5415    5418#
JSR1A   016120          5426#   5454
JSR2    016210          5419    5452#   5476
JSR2A   016240          5453    5455    5457    5464#
JSR2B   016250          5459    5468#
JSR3    016130          5431#   5439    5471
JSR3A   016170          5432    5434    5436    5438    5445#
JSR3B   016200          5440    5449#
JSR4    016272          5433    5450    5474#
JSR4A   016310          5475    5482#
JSR4B   016320          5477    5486#
JSR5    016376          5487    5506#   5510    5526
JSR5A   016422          5507    5509    5516#
JSR5B   016432          5511    5520#
JSR6    016332          5490#   5508    5539
JSR6A   016356          5491    5493    5500#
JSR6AD  016502          5494    5521    5539#   5545
JSR6B   016366          5495    5504#
JSR7    016444          5492    5505    5524#
JSR7A   016462          5525    5532#
JSR7B   016472          5527    5536#
MBDM2A  012214          4026    4027    4033#
MBDM2B  012224          4028    4037#
MBDM2C  012242          4038    4046#
MBDM2D  012254          4047    4048    4054#
MBDM2E  012264          4049    4058#
MBDM2F  012302          4059    4067#
MBDM4A  012636          4210    4218#
MBDM4B  012654          4219    4227#
MBDM4C  012666          4228    4229    4235#
MBDM4D  012676          4230    4239#
MBDM4E  012712          4240    4248#
MDM1A   012050          3941    3942    3948#
MDM1B   012060          3943    3952#
MDM2A   012122          3979    3980    3986#
MDM2B   012132          3981    3990#
MDM2C   012140          3997#
MDM2D   012150          3992    4001#
MDM3A   012356          4093    4094    4100#
MDM3B   012366          4095    4104#
MDM3C   012404          4105    4113#
MDM3D   012424          4114    4122#
MDM3E   012452          4124    4132#
MDM4A   012532          4160    4161    4167#
MDM4B   012542          4162    4171#
MDM4C   012556          4172    4180#
MDM5A   012762          4278    4279    4285#
```

```
MDM5B   012772          4280    4289#
MDM5C   013010          4290    4298#
MDM5D   013026          4299    4307#
MDM5E   013054          4310    4318#
MDM6A   013124          4347    4348    4354#
MDM6B   013134          4349    4358#
MDM6C   013152          4359    4367#
MDM6D   013172          4368    4376#
MDM6E   013222          4379    4387#
MDM7A   013276          4415    4416    4422#
MDM7B   013306          4417    4426#
MDM7C   013324          4427    4435#
MDM7D   013344          4436    4444#
MDM7E   013370          4446    4454#
MFPI0   024442          7564    7569#
MFPI0A  024470          7570    7575#
MFPS1   023206          7177    7186#
MFPS2A  023276          7211    7212    7213    7219#
MFPS2B  023306          7214    7223#
MFPS2C  023326          7224    7232#
MFPS3A  023404          7253    7254    7255    7261#
MFPS3B  023414          7256    7265#
MFPS3C  023434          7266    7274#
MFPS4A  023512          7295    7296    7297    7303#
MFPS4B  023522          7298    7307#
MFPS4C  023542          7308    7316#
MFPS5A  023620          7337    7338    7339    7345#
MFPS5B  023630          7340    7349#
MFPS5C  023650          7350    7358#
MFPS6A  023730          7379    7380    7381    7387#
MFPS6B  023740          7382    7391#
MFPS6C  023760          7392    7400#
MFPS7A  024040          7421    7422    7423    7429#
MFPS7B  024050          7424    7433#
MFPS7C  024070          7434    7442#
MOV1    016640          5613    5614    5620#
MOV2    016650          5615    5625#
MOV3    016666          5628    5629    5635#
MRK1    022246          6889    6896#
MRK2    022270          6896    6897    6898    6900    6907#
MRK3    022300          6902    6911#
MRK4    022322          6914    6916#
MRK5    022334          6915    6920#
MRK6    022350          6921    6928#
MSG     026140          7865    7929#
MTPI0   024554          7590    7595#
MTPS1   022420          6953    6961#
MTPS1A  022440          6965    6966    6967    6973#
MTPS2   022514          6990    6998#
MTPS3   022604          7021    7029#
MTPS4   022672          7051    7059#
MTPS5   022752          7081    7089#
MTPS6   023042          7111    7119#
MTPS7   023132          7141    7149#
NBR     024744          7650*   7653*   7662#
NEG00   004010          1501    1502    1503    1509#
```



```
NEG01   004020          1504    1514#
NEG02   004034          1515    1524#
NEG03   004050          1526    1527    1528    1534#
NEG04   004060          1529    1538#
NEG1    020174          6151    6152    6153    6159#
NEG10   004124          1558    1559    1560    1566#
NEG11   004134          1561    1571#
NEG12   004152          1572    1580#
NEG13   004166          1582    1583    1584    1590#
NEG14   004176          1585    1594#
NEG2    020204          6154    6163#
NEG20   004244          1614    1615    1616    1622#
NEG21   004254          1617    1626#
NEG22   004302          1632    1640#
NEG3    020226          6167    6168    6169    6175#
NEG30   004624          1818    1819    1820    1826#
NEG31   004634          1821    1830#
NEG32   004650          1831    1839#
NEG33   004674          1842    1850#
NEG34   004710          1851    1859#
NEG4    020236          6170    6179#
NEG40   005266          2057    2058    2059    2065#
NEG41   005276          2060    2069#
NEG42   005312          2070    2078#
NEG5    020256          6183    6184    6185    6191#
NEG50   005366          2102    2103    2104    2110#
NEG51   005376          2105    2114#
NEG52   005412          2115    2123#
NEG60   005470          2147    2148    2149    2155#
NEG61   005500          2150    2159#
NEG70   005550          2180    2181    2182    2188#
NEG71   005560          2183    2192#
OCTPSS  026062          7887*   7889*   7893*   7897*   7902    7919#
PASSPT  026060           121*   7467    7854*   7918#
PFMES   026366          7983    7990#
PS    = 177776             9#    809*    810     827*    828     845*    846     863*    864    6933#   6952    6987*   6989
                        7018*   7020    7048*   7050    7080    7108*   7110    7138*   7140    7174#   7188*   7209*   7251*
                        7293*   7335*   7377*   7419*   7469*   7471    7494*   7500*   7519*   7523*   7527*   7538*   7545*
                        7560*   7563    7565*   7571*   7583*   7585*   7589    7591*   7595*   7744*   7745*   7746    7749*
                        7753*   7809    7831
PSNUM   026170          7936#
PUSRM = 030000            13#
P#RDN   026310           119     7976#   7979
P#RUP   026320          7976    7979#
P#R2    026362          7982    7988#
REG1    001530           505#    508
REG1A   001574           529#    532
REG1E   001542           506     514#
REG2    001644           553#    556
REG2A   001672           549     554     557     566#
REG2B   001720           577#    580
REG2C   001752           582     588#
REG3    002000           599#    602
REG3A   002044           623#    626
REG3E   002012           600     608#
REG4    002114           648#    651
```

```
REG4A   002160   672#   675
REG4E   002126   649    657#
REG5    002230   698#   701
REG5A   002274   722#   725
REG5E   002242   699    707#
REG6    002344   747#   750
REG6A   002410   771#   774
REG6E   002356   748    756#
REST    024156  7468  7481#
RESTRT  000522   122#  7913
ROL1    021060  6455  6456   6457   6463#
ROL2    021070  6458  6467#
ROL3    021106  6470  6471   6472   6478#
ROL4    021116  6473  6482#
ROL5    021132  6485  6486   6492#
ROL6    021142  6487  6496#
ROL7    021164  6499  6500   6501   6508#
ROR1    021226  6523  6524   6525   6531#
ROR2    021236  6526  6535#
ROR3    021254  6538  6539   6540   6546#
ROR4    021264  6541  6550#
ROR5    021302  6553  6554   6555   6561#
ROR6    021312  6556  6565#
ROR7    021326  6568  6569   6575#
ROTX    014644  4890  4892*  4896   4909#  4924*  4926*  4928   4953*  4954   4958
ROTXAD  014770  4954* 4956*  4969#
ROTAA   013762  4621  4622   4629#
ROT0B   013772  4624  4633#
ROT0C   014014  4636  4637   4644#
ROT1A   014062  4669  4670   4677#
ROT1B   014072  4672  4681#
ROT1C   014116  4684  4685   4692#
ROT1D   014126  4687  4696#
ROT1E   014156  4701  4702   4709#
ROT2A   014230  4732  4734   4742#
ROT2B   014240  4737  4746#
ROT2C   014270  4750  4752   4759#
ROT2D   014300  4754  4763#
ROT2E   014334  4768  4770   4778#
ROT3A   014402  4800  4807#
ROT3B   014412  4802  4811#
ROT3C   014440  4814  4821#
ROT3D   014450  4816  4825#
ROT3E   014476  4828  4835#
ROT4    014552  4859  4861   4868#
ROT5    014634  4889  4895   4897   4904#
ROT6    014704  4927  4934#
ROT7    014760  4952  4957   4964#
RTS1    016574  5573  5580#
R1ERR   001612   530   539#
R2ERR   001742   578   584#
R3ERR   002062   624   633#
R4ERR   002176   673   682#
R5ERR   002312   723   732#
R6ERR   002426   772   781#
SBC1    020712  6376  6377   6378   6384#
```



```
SBC2    020722  6379  6388#
SBC3    020740  6391  6392   6393   6399#
SBC4    020750  6394  6403#
SBC5    020766  6406  6407   6408   6414#
SBC6    020776  6409  6418#
SBC7    021016  6422  6423   6429#
SB0     015022  4987  4995#
SB2     015144  5051  5059#
SB4     015270  5115  5123#
SB5     015356  5149  5168#
SB5A    015350  5154  5162#
SB5X    015366  5151* 5153   5173#  5174
SB5XAD  015370  5150  5162   5174#
SB6     015430  5191  5201#
SB6X    015440  5192* 5193   5206#
SB7     015500  5224  5234#
SB7X    015510  5225* 5239#  5240
SB7XAD  015512  5226  5240#
SCOPE = 000240     6#
SC3     025462  7805* 7830#  7845*  7846
SC4     025476  7806* 7834#  7844*
SETBR   024624  7640#  7679
SETCC   024644  7644#  7674
SETCD   025460  7826  7829#  7847
SETUP   024606  7637#
SET2BR  024714  7649  7653#
SHL     001200   337#   340
SHLE    001214   336   346#
SHR     001314   391#   394
SHRE    001330   387   390#
SNMB0A  005754  2297  2298   2304#
SNMB1A  006060  2362  2363   2369#
SNMB1B  006070  2364  2373#
SNMB1C  006112  2378  2379   2380   2386#
SNMB2A  006234  2454  2455   2461#
SNMB2B  006244  2456  2465#
SNMB2C  006260  2466  2474#
SNMB2D  006300  2478  2479   2480   2486#
SNMB2E  006310  2481  2490#
SNMB3A  006452  2570  2571   2577#
SNMB3B  006462  2572  2581#
SNMB3C  006500  2584  2585   2586   2592#
SNMB3D  006510  2587  2596#
SNM0A   005714  2264  2265   2266   2272#
SNM1A   006016  2329  2330   2331   2337#
SNM2A   006154  2410  2411   2412   2418#
SNM2B   006164  2413  2422#
SNM3A   006364  2522  2523   2524   2530#
SNM3B   006374  2525  2534#
SNM4A   006560  2627  2628   2634#
SNM4B   006570  2629  2638#
SNM5A   006642  2670  2671   2677#
SNM5B   006652  2672  2681#
SNM6A   006726  2713  2714   2720#
SNM6B   006736  2715  2724#
SNM7A   007010  2755  2756   2762#
```

```
SNM7B   007020  2757  2766#
SOB1    022126  6845# 6858
SOB2    022134  6845  6846  6852#
SOB3    022144  6847  6856#
SOB4    022164  6859  6860  6861  6868#
SOPA    005632  2222  2230#
SOPB    005652  2219  2234  2239#
SOPB0A  003256  1195  1203#
SOPB0B  003266  1204  1211#
SOPB1A  003400  1273  1281#
SOPB1B  003416  1282  1285  1292#
SOPB1C  003462  1319  1327#
SOPB1D  003502  1332  1339#
SOPB2A  003636  1415  1423#
SOPB2B  003656  1427  1435#
SOPB2C  003726  1459  1467#
SOPB2D  003752  1473  1481#
SOPB3A  004440  1725  1733#
SOPB3B  004464  1736  1738  1747#
SOPB3C  004532  1778  1786#
SOPB3D  004554  1792  1799#
SOPX    005616  2221# 2230* 2231*  2244
SOPXAD  005662  2233* 2244#
SOPZA   003550  1367  1375#
SOP0A   003114  1106  1114#
SOP0B   003134  1117  1125#
SOP0C   003176  1154  1162#
SOP0D   003222  1164  1167  1176#
SOP1A   003324  1233  1241#
SOP1B   003336  1242  1250#
SOP2B   003570  1378  1387#
SOP3A   004354  1676  1684#
SOP3B   004370  1687  1694#
SOP4A   004766  1890  1898#
SOP4B   005006  1901  1910#
SOP5A   005050  1941  1949#
SOP5B   005064  1952  1959#
SOP6A   005130  1983  1991#
SOP6B   005144  1992  1999#
SOP7A   005212  2025  2033#
SOP7B   005226  2034  2041#
START   000500   109   113#  7988
STBOT = 000500     3#   111   122   808   5418   5435   5456   5468   5571   6886   6901   6920   7544
                7546  7559  7569  7587  7596  7731  7980
SUB0    007154  2847  2848  2849  2855#
SUB0A   007164  2850  2859#
SUB1    020540  6309  6310  6316#
SUB2    020550  6311  6320#
SUB3    020572  6324  6325  6331#
SUB4    020602  6326  6335#
SUB5    020620  6339  6340  6346#
SUB6    020630  6341  6350#
SUB7    020650  6353  6354  6355  6361#
SWB1    017602  5964  5965  5971#
SWB2    017612  5966  5975#
SWB3    017630  5978  5979  5980  5986#
```



```
SXT0    021724  6742  6743  6744  6745  6752#
SXT1    021734  6747  6756#
SXT2    021764  6762  6763  6764  6765  6772#
TBL1    013472  4483  4484* 4485* 4486* 4487* 4488  4504# 4522* 4523* 4524* 4525* 4526  4537
                4538  4539  4540  4541  4558  4559* 4560# 4561* 4562* 4563  4590* 4591* 4592*
                4593* 4594
TBL2    013556  4521  4541# 4589
TEST1   017514  5926  5927  5928  5934#
TEST2   017524  5929  5938#
TEST3   017542  5942  5943  5949#
TITLE   026066  7863  7921#
TO10    026434    80  8006#
TO114   026474    89  8022#
TO14    026444    82  8010#
TO244   026504    92  8026#
TO250   026514    94  8030#
TO30    026454    84  8014#
TO34    026464    86  8018#
TO4     026424    78  8002#
TPB   = 177566   114# 7883* 7986*
TPS   = 177564   110# 7879  7984
TST1    000554   130#
TST10   001154   304   308   318   331#
TST100  006122  2355  2381  2402#
TST101  006202  2404  2424  2445#
TST102  006326  2447  2492  2512#
TST103  006412  2514  2536  2558#
TST104  006526  2560  2598  2618#
TST105  006604  2620  2639  2659#
TST106  006670  2661  2683  2703#
TST107  006752  2705  2725  2745#
TST11   001224   333   341   355#
TST110  007034  2747  2767  2786#
TST111  007070  2788  2794  2813#
TST112  007124  2815  2821  2840#
TST113  007200  2842  2860  2881#
TST114  007342  2883  2936  2954#
TST115  007432  2956  2977  2995#
TST116  007556  2997  3043  3060#
TST117  007642  3062  3083  3103#
TST12   001270   357   362   375#
TST120  007700  3105  3112  3131#
TST121  007736  3133  3140  3160#
TST122  007774  3162  3169  3193#
TST123  010054  3195  3214  3234#
TST124  010114  3236  3244  3264#
TST125  010166  3266  3282  3304#
TST126  010242  3306  3321  3340#
TST127  010320  3342  3360  3380#
TST13   001340   377   385   426#
TST130  010362  3382  3387  3406#
TST131  010424  3408  3413  3432#
TST132  010466  3434  3439  3452#
TST133  010544  3454  3472  3485#
TST134  010622  3487  3505  3518#
TST135  010720  3520  3548  3561#
```

```
TST136  011062  3563  3614  3628#
TST137  011222  3630  3676  3689#
TST14   001370   428   432   445#
TST140  011330  3691  3720  3733#
TST141  011472  3735  3784  3797#
TST142  011602  3799  3828  3841#
TST143  011710  3843  3871  3884#
TST144  012020  3886  3914  3933#
TST145  012074  3935  3953  3972#
TST146  012166  3974  4002  4020#
TST147  012322  4022  4068  4086#
TST15   001422   447   450   463#
TST150  012500  4088  4134  4153#
TST151  012574  4155  4181  4202#
TST152  012730  4204  4249  4271#
TST153  013072  4273  4319  4340#
TST154  013242  4342  4388  4408#
TST155  013410  4410  4455  4480#
TST156  013474  4489  4518#
TST157  013560  4527  4555#
TST16   001454   465   468   481#
TST160  013644  4557  4564  4586#
TST161  013730  4588  4595  4615#
TST162  014024  4617  4639  4662#
TST163  014166  4664  4704  4725#
TST164  014344  4727  4773  4794#
TST165  014506  4796  4830  4852#
TST166  014562  4854  4863  4887#
TST167  014646  4899  4921#
TST17   001506   483   486   499#
TST170  014714  4923  4929  4950#
TST171  014772  4959  4982#
TST172  015040  4984  4996  5016#
TST173  015102  5018  5023  5044#
TST174  015162  5046  5060  5081#
TST175  015224  5083  5087  5108#
TST176  015304  5110  5124  5147#
TST177  015372  5163  5189#
TST2    000644   132   165   192#
TST20   001552   501   509   523#
TST200  015442  5196  5222#
TST201  015514  5229  5277#
TST202  016060  5382  5412#
TST203  016536  5546  5968#
TST204  016612  5570  5981  5607#
TST205  016676  5609  5630  5643#
TST206  016766  5645  5667  5680#
TST207  017056  5682  5703  5716#
TST21   001622   525   534   547#
TST210  017150  5718  5741  5766#
TST211  017266  5768  5806  5820#
TST212  017430  5822  5873  5897#
TST213  017466  5899  5906  5920#
TST214  017552  5922  5944  5957#
TST215  017640  5959  5981  6004#
TST216  020024  6006  6067  6081#
```



```
TST217  020142  6083  6120  6144#
TST22   001676   571#
TST220  020266  6146  6186  6200#
TST221  020446  6202  6254  6268#
TST222  020506  6270  6277  6302#
TST223  020660  6304  6356  6369#
TST224  021026  6371  6424  6440#
TST225  021174  6450  6503  6516#
TST226  021336  6518  6570  6583#
TST227  021506  6585  6639  6652#
TST23   001756   573   593#
TST230  021666  6654  6711  6735#
TST231  021774  6737  6767  6788#
TST232  022106  6790  6819  6839#
TST233  022174  6841  6863  6883#
TST234  022360  6885  6923  6946#
TST235  022450  6948  6968  6982#
TST236  022532  6984  6999  7015#
TST237  022622  7014  7030  7043#
TST24   002022   595   603   617#
TST240  022706  7045  7060  7073#
TST241  022770  7075  7090  7103#
TST242  023060  7105  7120  7133#
TST243  023150  7135  7150  7171#
TST244  023242  7173  7191  7204#
TST245  023344  7206  7233  7246#
TST246  023452  7248  7275  7288#
TST247  023560  7290  7317  7330#
TST25   002072   619   628   642#
TST250  023666  7332  7359  7372#
TST251  023776  7374  7401  7414#
TST252  024106  7416  7443  7464#
TST253  024156  7466  7472  7491#
TST254  024226  7493  7516#
TST255  024364  7556#
TST256  024470  7558  7580#
TST257  024576  7582  7597  7634#
TST26   002136   644   652   666#
TST260  025036  7691#
TST261  025212  7741#
TST262  025266  7764#
TST263  025326  7766  7773  7800#
TST264  025532  7851#
TST27   002206   668   677   692#
TST3    000700   194   198   211#
TST30   002252   694   702   716#
TST31   002322   718   727   741#
TST32   002366   743   751   765#
TST33   002436   767   776   805#
TST34   002476   807   811   824#
TST35   002534   826   829   842#
TST36   002572   844   847   860#
TST37   002630   862   865   892#
TST4    000736   213   217   230#
TST40   002700   894   912   939#
TST41   002750   941   959   986#
```

```
TST42   003020    988    1006    1033#
TST43   003070   1035    1053    1102#
TST44   003150   1104    1126    1149#
TST45   003232   1151    1171    1191#
TST46   003276   1193    1206    1220#
TST47   003346   1230    1245    1266#
TST5    000774    232     236     249#
TST50   003426   1268    1287    1311#
TST51   003512   1313    1334    1358#
TST52   003600   1360    1382    1406#
TST53   003666   1408    1430    1449#
TST54   003762   1451    1476    1495#
TST55   004074   1497    1539    1551#
TST56   004214   1553    1595    1607#
TST57   004320   1609    1641    1668#
TST6    001032    251     254     282#
TST60   004400   1670    1689    1715#
TST61   004474   1717    1742    1769#
TST62   004564   1771    1794    1807#
TST63   004734   1809    1862    1883#
TST64   005016   1885    1905    1934#
TST65   005074   1936    1954    1976#
TST66   005154   1978    1994    2017#
TST67   005236   2019    2036    2050#
TST7    001076    284     289     302#
TST70   005326   2052    2079    2091#
TST71   005430   2093    2125    2137#
TST72   005514   2139    2160    2172#
TST73   005604   2174    2196    2217#
TST74   005664   2234    2257#
TST75   005724   2259    2267    2289#
TST76   005764   2291    2299    2321#
TST77   006026   2323    2332    2353#
USP1    024202   7497#   7498
USP1A   024226   7499    7504#
USP2    024274   7522    7527#
USP3    024320   7529    7537#
USP4    024346   7518    7540    7544#
USRM  = 140000     12#   7494    7500    7519    7523    7527    7538    7545    7560    7565    7571    7585    7589
                 7591
USTBOT  026424   7561    7569    7586    7995#
WAIT    025676   7864    7866    7876    7879#   7880    7884
WATE    026346   7984#   7985    7987
XOR1    022040   6796    6797    6798    6799    6806#
XOR2    022050   6801    6810#
XOR3    022076   6814    6815    6816    6817    6824#
YBR     024750   7651*   7654*   7664#
YNTAB   026252   7638    7959#
$APTHD  000330     61      67#
$CPUOP  000326     43#
$DEVCT  000310     34#
$ENDAD  026034     20    7859    7909#
$ENV    000320     39#
$ENVM   000321     40#   7857    7981
$ERN  = 000632      1#    141     142#    150     151#    161     162#    170     171#    203     204#    222     223#
                  241     242#    259     260#    294     295#    313     314#    323     324#    347     348#    367
```



```
                368#    391     392#    437     438#    455     456#    473     474#    491     492#    515     516#
                540     541#    563     564#    585     586#    609     610#    634     635#    658     659#    683
                684#    708     709#    733     734#    757     758#    782     783#    816     817#    834     835#
                852     853#    870     871#    905     906#    918     919#    952     953#    965     966#    999
               1000#   1012    1013#   1046    1047#   1059    1060#   1111    1112#   1122    1123#   1131    1132#
               1159    1160#   1177    1178#   1200    1201#   1212    1213#   1238    1239#   1251    1252#   1278
               1279#   1293    1294#   1324    1325#   1340    1341#   1372    1373#   1388    1389#   1420    1421#
               1436    1437#   1464    1465#   1482    1483#   1510    1511#   1520    1521#   1535    1536#   1544
               1545#   1567    1568#   1577    1578#   1591    1592#   1600    1601#   1623    1624#   1637    1638#
               1646    1647#   1681    1682#   1695    1696#   1730    1731#   1748    1749#   1783    1784#   1800
               1801#   1827    1828#   1836    1837#   1847    1848#   1856    1857#   1867    1868#   1895    1896#
               1911    1912#   1946    1947#   1960    1961#   1988    1989#   2000    2001#   2030    2031#   2042
               2043#   2066    2067#   2075    2076#   2084    2085#   2111    2112#   2123    2121#   2130    2131#
               2156    2157#   2165    2166#   2189    2190#   2201    2202#   2227    2228#   2240    2241#   2273
               2274#   2305    2306#   2338    2339#   2370    2371#   2387    2388#   2419    2420#   2429    2430#
               2462    2463#   2471    2472#   2487    2488#   2497    2498#   2531    2532#   2541    2542#   2578
               2579#   2593    2594#   2603    2604#   2635    2636#   2644    2645#   2678    2679#   2688    2689#
               2721    2722#   2730    2731#   2763    2764#   2772    2773#   2799    2800#   2826    2827#   2856
               2857#   2865    2866#   2891    2892#   2904    2905#   2915    2916#   2930    2931#   2941    2942#
               2972    2973#   2982    2983#   3007    3008#   3016    3017#   3026    3027#   3038    3039#   3048
               3049#   3077    3078#   3089    3090#   3117    3118#   3145    3146#   3174    3175#   3209    3210#
               3219    3220#   3249    3250#   3277    3278#   3287    3288#   3317    3318#   3326    3327#   3355
               3356#   3365    3366#   3392    3393#   3418    3419#   3444    3445#   3468    3469#   3477    3478#
               3501    3502#   3510    3511#   3533    3534#   3544    3545#   3553    3554#   3578    3579#   3587
               3588#   3600    3601#   3610    3611#   3619    3620#   3648    3649#   3657    3658#   3671    3672#
               3681    3682#   3707    3708#   3716    3717#   3725    3726#   3751    3752#   3760    3761#   3771
               3772#   3780    3781#   3789    3790#   3815    3816#   3824    3825#   3833    3834#   3858    3859#
               3867    3868#   3876    3877#   3901    3902#   3910    3911#   3919    3920#   3949    3950#   3958
               3959#   3987    3988#   3998    3999#   4007    4008#   4034    4035#   4043    4044#   4055    4056#
               4064    4065#   4073    4074#   4101    4102#   4110    4111#   4119    4120#   4129    4130#   4139
               4140#   4168    4169#   4177    4178#   4186    4187#   4215    4216#   4224    4225#   4236    4237#
               4245    4246#   4254    4255#   4286    4287#   4295    4296#   4304    4305#   4315    4316#   4324
               4325#   4355    4356#   4364    4365#   4373    4374#   4384    4385#   4393    4394#   4423    4424#
               4432    4433#   4441    4442#   4451    4452#   4460    4461#   4495    4496#   4533    4534#   4569
               4570#   4600    4601#   4630    4631#   4645    4646#   4678    4679#   4693    4694#   4710    4711#
               4743    4744#   4760    4761#   4779    4780#   4808    4809#   4822    4823#   4836    4837#   4869
               4870#   4905    4906#   4935    4936#   4965    4966#   4992    4993#   5001    5002#   5028    5029#
               5056    5057#   5065    5066#   5092    5093#   5120    5121#   5129    5130#   5159    5160#   5169
               5170#   5202    5203#   5235    5236#   5289    5290#   5298    5299#   5312    5313#   5324    5325#
               5333    5334#   5346    5347#   5359    5360#   5373    5374#   5387    5388#   5427    5428#   5446
               5447#   5465    5466#   5483    5484#   5501    5502#   5517    5518#   5533    5534#   5552    5553#
               5577    5578#   5586    5587#   5621    5622#   5636    5637#   5658    5659#   5673    5674#   5695
               5696#   5709    5710#   5732    5733#   5747    5748#   5781    5782#   5797    5798#   5812    5813#
               5835    5836#   5849    5850#   5863    5864#   5879    5880#   5913    5913#   5935    5936#   5950
               5951#   5972    5973#   5987    5988#   6018    6019#   6032    6033#   6046    6047#   6058    6059#
               6073    6074#   6096    6097#   6111    6112#   6126    6127#   6160    6161#   6176    6177#   6192
               6193#   6215    6216#   6230    6231#   6246    6247#   6260    6261#   6283    6284#   6317    6318#
               6332    6333#   6347    6348#   6362    6363#   6385    6386#   6400    6401#   6415    6416#   6430
               6431#   6464    6465#   6479    6480#   6493    6494#   6509    6510#   6533    6533#   6547    6548#
               6562    6563#   6576    6577#   6599    6600#   6614    6615#   6628    6629#   6645    6646#   6668
               6669#   6684    6685#   6699    6700#   6717    6718#   6753    6754#   6773    6774#   6807    6808#
               6825    6826#   6853    6854#   6869    6870#   6893    6894#   6908    6909#   6917    6918#   6929
               6930#   6958    6959#   6974    6975#   6995    6996#   7004    7005#   7026    7027#   7035    7036#
               7056    7057#   7065    7066#   7086    7087#   7095    7096#   7116    7117#   7125    7126#   7146
               7147#   7155    7156#   7182    7183#   7196    7197#   7220    7221#   7229    7230#   7238    7239#
               7262    7263#   7271    7272#   7280    7281#   7304    7305#   7313    7314#   7322    7323#   7346
```

$ERROR= 000302
$ETABL 000320
$ETEND 000330
$FATAL 000302
$HIBTS 000330
$MAIL   000300
$MBADR 000332
$MSGAD 000314
$MSGLG 000316
$MSGTV 000300
$PASS   000306
$PASTM 000336
$SVPC = 000400
$SWR = 000000
$SWREG 000322
$TESTN 000304
$TN   = 000265

7347# 7355 7356# 7364 7365# 7388 7389# 7397 7398# 7406 7407# 7430 7431#
7439 7440# 7448 7449# 7477 7478# 7501 7502# 7524 7525# 7534 7535# 7541
7562# 7566 7567# 7572 7573# 7592 7593# 7602 7603# 7666 7667# 7727 7728#
7750 7751# 7780# 7788 7818 7819# 7841 7842# 7915 7916# 8003 8004# 8007
8008# 8011 8012# 8015 8016# 8019 8020# 8023 8024# 8027 8028# 8031 8032#
117# 125*
38#
50# 73
31# 117
68#
29# 69 73
69#
36#
37#
30# 126*
33# 120* 7856* 7861 7887
71#
18# 23
1#
44#
32# 112 118 123 562 566 583 588 7665 7730
1# 127 133# 165 189 195# 198 208 214# 217 227 233# 236
246 252# 254 279 285# 289 299 305# 308 318 328 334# 341
352 358# 362 372 378# 385 423 429# 432 442 448# 450 460
466# 468 478 484# 486 496 502# 509 520 526# 534 544 550#
568 574# 590 596# 603 614 620# 628 639 645# 652 663 669#
677 689 695# 702 713 719# 727 738 744# 751 762 768# 776
802 808# 811 821 827# 829 839 845# 847 857 863# 865 889
895# 912 936 942# 959 983 989# 1006 1030 1036# 1053 1099 1105#
1126 1146 1152# 1171 1188 1194# 1206 1225 1231# 1245 1263 1269# 1287
1308 1314# 1334 1355 1361# 1382 1403 1409# 1430 1446 1452# 1476 1492
1498# 1539 1548 1554# 1595 1604 1610# 1641 1665 1671# 1689 1712 1718#
1742 1766 1772# 1794 1804 1810# 1862 1880 1886# 1905 1931 1937# 1954
1973 1979# 1994 2014 2020# 2036 2047 2053# 2079 2088 2094# 2125 2134
2140# 2160 2169 2175# 2196 2214 2220# 2234 2254 2260# 2267 2286 2292#
2299 2318 2324# 2332 2350 2356# 2381 2399 2405# 2424 2442 2448# 2483 2492
2509 2515# 2536 2555 2561# 2598 2615 2621# 2639 2656 2662# 2683 2700
2706# 2715 2742 2748# 2767 2783 2789# 2794 2810 2816# 2821 2837 2843#
2860 2878 2884# 2936 2951 2957# 2977 2992 2998# 3043 3057 3063# 3083
3100 3106# 3112 3128 3134# 3140 3157 3163# 3169 3190 3196# 3214 3231
3237# 3244 3261 3267# 3282 3301 3307# 3321 3337 3343# 3360 3377 3383#
3387 3403 3409# 3413 3429 3435# 3439 3449 3455# 3472 3482 3488# 3505
3515 3521# 3548 3558 3564# 3614 3625 3631# 3676 3686 3692# 3720 3730
3736# 3784 3794 3800# 3828 3838 3844# 3871 3881 3887# 3914 3930 3936#
3953 3969 3975# 4002 4017 4023# 4068 4083 4089# 4134 4150 4156# 4181
4199 4205# 4249 4268 4274# 4319 4337 4343# 4388 4405 4411# 4455 4477
4483# 4489 4515 4521# 4527 4552 4558# 4564 4583 4589# 4595 4612 4618#
4639 4659 4665# 4704 4722 4728# 4773 4791 4797# 4830 4849 4855# 4863
4884 4890# 4899 4918 4924# 4929 4947 4953# 4959 4979 4985# 4996 5013
5019# 5023 5041 5047# 5060 5078 5084# 5087 5105 5111# 5124 5144 5150#
5163 5186 5192# 5196 5219 5225# 5229 5274 5280# 5382 5409 5415# 5546
5565 5571# 5581 5604 5610# 5630 5640 5646# 5667 5677 5683# 5703 5713
5719# 5741 5763 5769# 5806 5817 5823# 5873 5894 5900# 5906 5917 5923#
5944 5954 5960# 5981 6001 6007# 6067 6078 6084# 6120 6141 6147# 6186
6197 6203# 6254 6265 6271# 6277 6299 6305# 6356 6366 6372# 6424 6445
6451# 6503 6513 6519# 6570 6580 6586# 6639 6649 6655# 6711 6732 6738#

$TSTM   000334
$TSTNM= 000304
$UNIT   000312
$UNITM 000340
$USWR   000324
$X   = 025542
$XX   = 177716

6767 6785 6791# 6819 6836 6842# 6863 6880 6886# 6923 6943 6949# 6968
6979 6985# 6999 7009 7015# 7030 7040 7046# 7060 7070 7076# 7090 7100
7106# 7120 7130 7136# 7150 7168 7174# 7191 7201 7207# 7233 7243 7249#
7275 7285 7291# 7317 7327 7333# 7359 7369 7375# 7401 7411 7417# 7443
7461 7467# 7472 7488 7494# 7513 7519# 7553 7559# 7577 7583# 7597 7631
7637# 7688 7694# 7738 7744# 7761 7767# 7773 7797 7803# 7848 7854#
70#
118# 124*
35#
72#
42#
133# 148 168 195# 201 214# 220 233# 239 252# 257 285# 292
305# 311 321 334# 344 358# 365 378# 388 429# 435 448# 453
466# 471 484# 489 502# 512 526# 537 550# 574# 596# 606 620#
631 645# 655 669# 680 695# 705 719# 730 744# 754 768# 779
808# 814 827# 832 845# 850 863# 868 895# 902 915 942# 949
962 989# 996 1009 1036# 1043 1056 1105# 1109 1120 1129 1152# 1157
1174 1194# 1198 1209 1231# 1236 1248 1269# 1275 1290 1314# 1322 1337
1361# 1370 1385 1409# 1418 1433 1452# 1462 1479 1498# 1507 1518 1532
1542 1554# 1564 1575 1588 1598 1610# 1620 1635 1644 1671# 1679 1692
1718# 1728 1745 1772# 1781 1797 1810# 1824 1834 1845 1854 1879 1886#
1893 1908 1937# 1944 1957 1979# 1986 1997 2020# 2028 2039 2065 2053#
2073 2082 2094# 2108 2118 2128 2140# 2153 2163 2175# 2186 2199 2220#
2225 2237 2260# 2270 2292# 2302 2324# 2335 2356# 2367 2384 2405# 2416
2427 2448# 2459 2469 2484 2495 2515# 2528 2539 2561# 2575 2590 2601
2621# 2632 2642 2662# 2675 2686 2706# 2718 2728 2748# 2760 2770 2789#
2797 2816# 2824 2843# 2853 2863 2884# 2889 2902 2913 2928 2939 2957#
2969 2980 2998# 3005 3014 3024 3036 3046 3063# 3074 3086 3106# 3115
3134# 3143 3163# 3172 3196# 3207 3217 3237# 3247 3267# 3275 3285 3307#
3315 3324 3343# 3353 3363 3383# 3390 3409# 3416 3435# 3442 3455# 3465
3475 3488# 3498 3508 3521# 3530 3541 3551 3564# 3575 3585 3597 3608
3617 3631# 3645 3655 3668 3679 3692# 3704 3714 3723 3736# 3748 3758
3768 3778 3787 3800# 3812 3822 3831 3844# 3855 3865 3874 3887# 3898
3908 3917 3936# 3946 3956 3975# 3984 3995 4005 4023# 4031 4041 4052
4062 4071 4089# 4098 4108 4117 4127 4137 4156# 4165 4175 4184 4205#
4213 4222 4233 4243 4252 4274# 4283 4293 4302 4313 4322 4343# 4352
4362 4371 4382 4391 4411# 4420 4430 4439 4449 4458 4483# 4492 4521#
4530 4558# 4567 4589# 4598 4618# 4627 4642 4665# 4675 4690 4707 4728#
4740 4757 4776 4797# 4805 4819 4833 4855# 4866 4890# 4902 4924# 4939
4953# 4962 4985# 4990 4999 5019# 5026 5047# 5054 5063 5084# 5090 5111#
5118 5127 5150# 5157 5166 5192# 5199 5225# 5232 5280# 5287 5296 5310
5322 5331 5344 5357 5371 5385 5415# 5443 5462 5480 5498 5514 5530
5549 5571# 5584 5610# 5618 5633 5646# 5655 5670 5683# 5692 5706 5719#
5729 5744 5769# 5778 5794 5809 5823# 5832 5846 5860 5876 5900# 5909
5923# 5932 5947 5960# 5969 5984 6007# 6015 6029 6043 6055 6070 6084#
6093 6108 6123 6147# 6157 6173 6189 6203# 6212 6227 6243 6257 6271#
6280 6305# 6314 6329 6344 6359 6372# 6382 6397 6412 6427 6451# 6461
6476 6490 6506 6519# 6529 6544 6559 6573 6586# 6596 6611 6625 6642
6655# 6665 6681 6696 6714 6730# 6750 6770 6791# 6804 6822 6842# 6850
6866 6886# 6905 6926 6949# 6956 6971 6985# 6993 7002 7015# 7024 7033
7046# 7054 7063 7076# 7084 7093 7106# 7114 7123 7136# 7144 7153 7174#
7180 7194 7207# 7217 7227 7236 7249# 7259 7269 7278 7291# 7301 7311
7320 7333# 7343 7353 7362 7375# 7385 7395 7404 7417# 7427 7437 7446
7467# 7475 7494# 7519# 7532 7559# 7583# 7600 7637# 7694# 7724 7744# 7767#
7776 7803# 7816 7838 7854#
148# 168# 201# 220# 239# 257# 292# 311# 321# 344# 365# 388# 435#

```
                453#  471#  489#  512#  537#  606#  631#  655#  680#  705#  730#  754#  779#
                814#  832#  850#  868#  902#  915#  949#  962#  996# 1009# 1043# 1056# 1109#
               1120# 1129# 1157# 1174# 1196# 1209# 1236# 1248# 1276# 1290# 1322# 1337# 1370#
               1385# 1418# 1433# 1462# 1479# 1507# 1518# 1532# 1547# 1564# 1575# 1588# 1598#
               1620# 1635# 1644# 1679# 1692# 1728# 1745# 1781# 1797# 1824# 1834# 1845# 1854#
               1865# 1893# 1908# 1944# 1957# 1986# 1997# 2026# 2039# 2063# 2073# 2082# 2108#
               2118# 2128# 2153# 2163# 2186# 2199# 2225# 2237# 2270# 2302# 2335# 2367# 2384#
               2416# 2427# 2459# 2469# 2484# 2495# 2520# 2530# 2575# 2590# 2601# 2632# 2642#
               2675# 2686# 2718# 2728# 2760# 2770# 2797# 2824# 2853# 2863# 2889# 2902# 2913#
               2928# 2939# 2969# 2980# 3005# 3014# 3024# 3036# 3046# 3074# 3086# 3115# 3143#
               3172# 3207# 3217# 3247# 3275# 3285# 3315# 3324# 3353# 3363# 3390# 3416# 3442#
               3465# 3475# 3498# 3508# 3530# 3541# 3551# 3575# 3585# 3597# 3608# 3617# 3645#
               3655# 3668# 3679# 3704# 3714# 3723# 3748# 3758# 3768# 3778# 3787# 3812# 3822#
               3831# 3855# 3865# 3874# 3898# 3908# 3917# 3946# 3956# 3984# 3995# 4005# 4031#
               4041# 4052# 4062# 4071# 4098# 4106# 4117# 4127# 4137# 4165# 4175# 4184# 4213#
               4222# 4233# 4243# 4252# 4283# 4293# 4302# 4313# 4322# 4352# 4362# 4371# 4382#
               4391# 4420# 4430# 4439# 4449# 4458# 4492# 4530# 4567# 4598# 4627# 4642# 4675#
               4690# 4707# 4740# 4757# 4776# 4805# 4819# 4833# 4866# 4902# 4932# 4964# 4990#
               4999# 5026# 5054# 5063# 5090# 5118# 5127# 5157# 5166# 5195# 5232# 5261# 5296#
               5310# 5322# 5331# 5344# 5357# 5371# 5385# 5443# 5462# 5480# 5498# 5514# 5530#
               5549# 5584# 5618# 5633# 5655# 5670# 5692# 5706# 5729# 5744# 5778# 5794# 5809#
               5832# 5846# 5860# 5876# 5909# 5932# 5947# 5969# 5984# 6015# 6029# 6043# 6055#
               6070# 6093# 6108# 6123# 6157# 6173# 6189# 6212# 6227# 6243# 6257# 6280# 6314#
               6329# 6344# 6359# 6387# 6397# 6412# 6427# 6461# 6476# 6490# 6506# 6529# 6544#
               6559# 6573# 6596# 6611# 6625# 6642# 6665# 6681# 6696# 6714# 6750# 6770# 6804#
               6822# 6850# 6866# 6905# 6926# 6956# 6971# 6993# 7002# 7024# 7033# 7054# 7063#
               7084# 7093# 7114# 7123# 7144# 7153# 7180# 7194# 7217# 7227# 7236# 7259# 7269#
               7278# 7301# 7311# 7320# 7343# 7353# 7362# 7385# 7395# 7404# 7427# 7437# 7446#
               7475# 7532# 7600# 7724# 7776# 7816# 7838#
$XXX  = 000716  148#  168#  201#  220#  239#  257#  292#  311#  321#  344#  365#  388#  435#
                453#  471#  489#  512#  537#  606#  631#  655#  680#  705#  730#  754#  779#
                814#  832#  850#  868#  902#  915#  949#  962#  996# 1009# 1043# 1056# 1109#
               1120# 1129# 1157# 1174# 1196# 1209# 1236# 1248# 1276# 1290# 1322# 1337# 1370#
               1385# 1418# 1433# 1462# 1479# 1507# 1518# 1532# 1547# 1564# 1575# 1588# 1598#
               1620# 1635# 1644# 1679# 1692# 1728# 1745# 1781# 1797# 1824# 1834# 1845# 1854#
               1865# 1893# 1908# 1944# 1957# 1986# 1997# 2026# 2039# 2063# 2073# 2082# 2108#
               2118# 2128# 2153# 2163# 2186# 2199# 2225# 2237# 2270# 2302# 2335# 2367# 2384#
               2416# 2427# 2459# 2469# 2484# 2495# 2520# 2530# 2575# 2590# 2601# 2632# 2642#
               2675# 2686# 2718# 2728# 2760# 2770# 2797# 2824# 2853# 2863# 2889# 2902# 2913#
               2928# 2939# 2969# 2980# 3005# 3014# 3024# 3036# 3046# 3074# 3086# 3115# 3143#
               3172# 3207# 3217# 3247# 3275# 3285# 3315# 3324# 3353# 3363# 3390# 3416# 3442#
               3465# 3475# 3498# 3508# 3530# 3541# 3551# 3575# 3585# 3597# 3608# 3617# 3645#
               3655# 3668# 3679# 3704# 3714# 3723# 3748# 3758# 3768# 3778# 3787# 3812# 3822#
               3831# 3855# 3865# 3874# 3898# 3908# 3917# 3946# 3956# 3984# 3995# 4005# 4031#
               4041# 4052# 4062# 4071# 4098# 4106# 4117# 4127# 4137# 4165# 4175# 4184# 4213#
               4222# 4233# 4243# 4252# 4283# 4293# 4302# 4313# 4322# 4352# 4362# 4371# 4382#
               4391# 4420# 4430# 4439# 4449# 4458# 4492# 4530# 4567# 4598# 4627# 4642# 4675#
               4690# 4707# 4740# 4757# 4776# 4805# 4819# 4833# 4866# 4902# 4932# 4964# 4990#
               4999# 5026# 5054# 5063# 5090# 5118# 5127# 5157# 5166# 5195# 5232# 5261# 5296#
               5310# 5322# 5331# 5344# 5357# 5371# 5385# 5443# 5462# 5480# 5498# 5514# 5530#
               5549# 5584# 5618# 5633# 5655# 5670# 5692# 5706# 5729# 5744# 5778# 5794# 5809#
               5832# 5846# 5860# 5876# 5909# 5932# 5947# 5969# 5984# 6015# 6029# 6043# 6055#
               6070# 6093# 6108# 6123# 6157# 6173# 6189# 6212# 6227# 6243# 6257# 6280# 6314#
               6329# 6344# 6359# 6387# 6397# 6412# 6427# 6461# 6476# 6490# 6506# 6529# 6544#
               6559# 6573# 6596# 6611# 6625# 6642# 6665# 6681# 6696# 6714# 6750# 6770# 6804#
               6822# 6850# 6866# 6905# 6926# 6956# 6971# 6993# 7002# 7024# 7033# 7054# 7063#
```



```
               7084# 7093# 7114# 7123# 7144# 7153# 7180# 7194# 7217# 7227# 7236# 7259# 7269#
               7278# 7301# 7311# 7320# 7343# 7353# 7362# 7385# 7395# 7404# 7427# 7437# 7446#
               7475# 7532# 7600# 7724# 7776# 7816# 7838#
      = 026524   14#   18#   19#   21#   23#   24#   57#   58#   60#   62#   77#   83#   88#
                 91#  100#  104#  107#  108#  116#  133#  148#  168#  195#  201#  214#  220#
                233#  239#  252#  257#  285#  292#  305#  311#  321#  334#  344#  358#  365#
                378#  388#  429#  435#  448#  453#  466#  471#  484#  489#  502#  512#  526#
                537#  550#  574#  596#  606#  620#  631#  645#  655#  669#  680#  695#  705#
                719#  730#  744#  754#  768#  779#  808#  814#  827#  832#  845#  850#  863#
                868#  895#  902#  915#  942#  949#  962#  989#  996# 1009# 1036# 1043# 1056#
               1105# 1109# 1120# 1129# 1152# 1157# 1174# 1194# 1198# 1209# 1231# 1236# 1248#
               1269# 1276# 1290# 1314# 1322# 1337# 1361# 1370# 1385# 1409# 1418# 1433# 1452#
               1462# 1479# 1498# 1507# 1518# 1532# 1542# 1554# 1564# 1575# 1588# 1598# 1610#
               1620# 1635# 1644# 1671# 1679# 1692# 1718# 1728# 1745# 1772# 1781# 1797# 1810#
               1824# 1834# 1845# 1854# 1865# 1886# 1893# 1908# 1937# 1944# 1957# 1979# 1986#
               1997# 2020# 2028# 2039# 2053# 2063# 2073# 2082# 2094# 2108# 2118# 2128# 2140#
               2153# 2163# 2175# 2186# 2199# 2220# 2225# 2237# 2260# 2270# 2292# 2302# 2324#
               2335# 2356# 2367# 2384# 2405# 2416# 2427# 2448# 2459# 2469# 2484# 2495# 2515#
               2528# 2539# 2561# 2575# 2590# 2601# 2621# 2632# 2642# 2662# 2675# 2686# 2706#
               2718# 2728# 2748# 2760# 2770# 2789# 2797# 2816# 2824# 2843# 2853# 2863# 2884#
               2889# 2902# 2913# 2928# 2939# 2957# 2969# 2980# 2998# 3005# 3014# 3024# 3036#
               3046# 3063# 3074# 3086# 3106# 3115# 3134# 3143# 3163# 3172# 3196# 3207# 3217#
               3237# 3247# 3267# 3275# 3285# 3307# 3315# 3324# 3343# 3353# 3363# 3383# 3390#
               3409# 3416# 3435# 3442# 3455# 3465# 3475# 3488# 3498# 3508# 3521# 3530# 3541#
               3551# 3564# 3575# 3585# 3597# 3608# 3617# 3631# 3645# 3655# 3668# 3679# 3692#
               3704# 3714# 3723# 3736# 3748# 3758# 3768# 3778# 3787# 3800# 3812# 3822# 3831#
               3844# 3855# 3865# 3874# 3887# 3898# 3908# 3917# 3936# 3946# 3956# 3975# 3984#
               3995# 4005# 4023# 4031# 4041# 4052# 4062# 4071# 4089# 4098# 4108# 4117# 4127#
               4137# 4156# 4165# 4175# 4184# 4205# 4213# 4222# 4235# 4243# 4252# 4274# 4283#
               4293# 4302# 4313# 4322# 4343# 4352# 4362# 4371# 4382# 4391# 4411# 4420# 4430#
               4439# 4449# 4458# 4468# 4483# 4492# 4521# 4530# 4558# 4567# 4589# 4598# 4618# 4627#
               4642# 4655# 4675# 4690# 4707# 4728# 4740# 4757# 4776# 4797# 4805# 4819# 4833#
               4852# 4866# 4890# 4902# 4924# 4932# 4953# 4962# 4985# 4990# 4999# 5019# 5026#
               5047# 5054# 5063# 5084# 5090# 5111# 5118# 5127# 5150# 5157# 5166# 5192# 5199#
               5225# 5232# 5280# 5283# 5287# 5296# 5310# 5322# 5331# 5344# 5357# 5371# 5385#
               5415# 5443# 5462# 5480# 5498# 5514# 5530# 5549# 5571# 5584# 5610# 5618# 5633#
               5646# 5655# 5670# 5683# 5692# 5706# 5719# 5729# 5744# 5769# 5778# 5794# 5809#
               5823# 5832# 5846# 5860# 5876# 5900# 5909# 5923# 5932# 5947# 5960# 5969# 5984#
               6007# 6015# 6029# 6043# 6055# 6070# 6084# 6093# 6108# 6123# 6147# 6157# 6173#
               6189# 6203# 6212# 6227# 6243# 6257# 6271# 6280# 6305# 6314# 6329# 6344# 6359#
               6372# 6382# 6397# 6412# 6427# 6451# 6461# 6476# 6490# 6506# 6519# 6529# 6544#
               6559# 6573# 6586# 6596# 6611# 6625# 6642# 6655# 6665# 6681# 6696# 6714# 6738#
               6750# 6770# 6791# 6804# 6822# 6842# 6850# 6866# 6886# 6905# 6926# 6949# 6956#
               6971# 6985# 6993# 7002# 7015# 7024# 7033# 7046# 7054# 7063# 7076# 7084# 7093#
               7106# 7114# 7123# 7136# 7144# 7153# 7174# 7180# 7194# 7207# 7217# 7227# 7236#
               7249# 7259# 7269# 7278# 7291# 7301# 7311# 7320# 7333# 7343# 7353# 7362# 7375#
               7385# 7395# 7404# 7417# 7427# 7437# 7446# 7467# 7475# 7494# 7519# 7532# 7559#
               7583# 7600# 7637# 7648# 7673# 7678# 7694# 7724# 7747# 7767# 7776# 7803# 7816#
               7838# 7854# 7942# 7943# 7944# 7945# 7946# 7947# 7948# 7949# 7950# 7951# 7952#
               7953# 7954# 7955# 7956# 7994#
.$X   = 000500   57#   62#  107#  116#
```

COMMEN	1#													
ENDCOM	1#													
ERROR	1#	140	144	160	164	198	217	236	254	289	308	318	341	362	385
	432	450	468	486	509	534	563	584	603	628	652	677	702	727	751
	776	811	829	847	865	899	912	946	959	993	1006	1040	1053	1106	1117
	1126	1154	1171	1195	1206	1233	1245	1273	1287	1319	1334	1367	1382	1415	1430
	1459	1476	1504	1515	1529	1539	1561	1572	1585	1595	1617	1632	1641	1676	1689
	1725	1742	1778	1794	1821	1831	1842	1851	1862	1890	1905	1941	1954	1983	1994
	2025	2036	2060	2070	2079	2105	2115	2125	2150	2160	2183	2196	2222	2234	2267
	2299	2332	2364	2381	2413	2424	2456	2466	2481	2492	2525	2536	2572	2587	2598
	2629	2639	2672	2683	2715	2725	2757	2767	2794	2821	2850	2860	2886	2899	2910
	2925	2936	2966	2977	3002	3011	3021	3033	3043	3071	3083	3112	3140	3169	3204
	3214	3244	3272	3282	3312	3321	3350	3360	3387	3413	3439	3462	3472	3495	3505
	3527	3538	3548	3572	3582	3594	3605	3614	3642	3652	3665	3676	3701	3711	3720
	3745	3755	3765	3775	3784	3809	3819	3828	3852	3862	3871	3895	3905	3914	3943
	3953	3981	3992	4002	4028	4038	4049	4059	4068	4095	4105	4114	4124	4134	4152
	4179	4188	4217	4227	4236	4246	4255	4265	4280	4290	4299	4310	4319	4349	4359
	4379	4388	4417	4427	4436	4446	4455	4489	4527	4564	4595	4624	4639	4672	4687
	4704	4737	4754	4773	4802	4816	4830	4863	4899	4929	4959	4987	4996	5023	5051
	5060	5087	5115	5124	5154	5163	5196	5229	5284	5263	5307	5319	5328	5341	5354
	5368	5382	5426	5440	5459	5477	5495	5511	5527	5546	5577	5581	5615	5630	5652
	5667	5689	5703	5726	5741	5775	5791	5806	5829	5843	5857	5873	5906	5929	5944
	5966	5981	6012	6026	6040	6052	6067	6090	6105	6120	6154	6170	6186	6209	6224
	6240	6254	6277	6311	6326	6341	6356	6379	6394	6409	6424	6458	6473	6487	6503
	6526	6541	6556	6570	6593	6608	6622	6639	6662	6678	6693	6711	6747	6767	6801
	6819	6847	6863	6893	6902	6917	6923	6953	6968	6990	6999	7021	7030	7051	7060
	7081	7090	7111	7120	7141	7150	7177	7191	7214	7224	7233	7256	7266	7275	7298
	7308	7317	7340	7350	7359	7382	7392	7401	7424	7434	7443	7472	7501	7524	7529
	7541	7566	7572	7592	7597	7666	7721	7750	7773	7813	7835	7914	8002	8006	8010
	8014	8018	8022	8026	8030										
ESCAPE	1#														
GETPRI	1#														
GETSWR	1#														
JNE	6721#	7646	7671	7676											
LOOP	1#	148	168	201	220	239	257	292	311	321	344	365	388	435	453
	471	489	512	537	606	631	655	680	705	730	754	779	814	832	850
	868	902	915	949	962	996	1009	1043	1056	1109	1120	1129	1157	1174	1198
	1209	1236	1248	1276	1290	1322	1337	1370	1385	1418	1433	1462	1479	1507	1518
	1532	1542	1564	1575	1588	1598	1620	1635	1644	1679	1692	1728	1745	1781	1797
	1824	1834	1845	1854	1865	1893	1908	1944	1957	1986	1997	2028	2039	2063	2073
	2082	2108	2118	2128	2153	2163	2186	2199	2225	2237	2270	2302	2335	2367	2384
	2416	2427	2459	2469	2484	2495	2528	2539	2575	2590	2601	2632	2642	2675	2686
	2718	2728	2760	2770	2797	2824	2853	2863	2889	2902	2913	2928	2939	2969	2980
	3005	3014	3024	3036	3046	3074	3086	3115	3143	3172	3207	3217	3247	3275	3285
	3315	3324	3353	3363	3390	3416	3442	3465	3475	3498	3508	3530	3541	3551	3575
	3585	3597	3608	3617	3645	3655	3668	3679	3704	3714	3723	3748	3758	3768	3778
	3787	3812	3822	3831	3855	3865	3874	3898	3908	3917	3946	3956	3984	3995	4005
	4031	4041	4052	4062	4071	4098	4108	4117	4127	4137	4165	4175	4184	4213	4223
	4233	4243	4252	4283	4293	4302	4313	4322	4352	4362	4371	4407	4417	4427	4440
	4459	4449	4458	4492	4520	4547	4556	4567	4642	4652	4690	4700	4740	4757	4767
	4805	4815	4833	4866	4902	4932	4962	4990	4999	5026	5054	5063	5090	5118	5127
	5157	5166	5199	5232	5287	5296	5310	5322	5331	5344	5357	5371	5385	5443	5462
	5480	5498	5514	5530	5549	5564	5618	5633	5655	5670	5692	5706	5729	5744	5778
	5794	5809	5832	5846	5860	5876	5909	5932	5947	5969	5984	6015	6029	6043	6055
	6070	6093	6108	6123	6157	6173	6189	6212	6227	6243	6257	6280	6314	6329	6344
	6359	6382	6397	6412	6427	6461	6476	6490	6506	6529	6544	6559	6573	6596	6611

	6625	6642	6665	6681	6696	6714	6750	6770	6804	6822	6850	6866	6905	6926	6956
	6971	6993	7002	7024	7033	7054	7063	7084	7093	7114	7123	7144	7153	7180	7194
	7217	7227	7236	7259	7269	7278	7301	7311	7320	7343	7353	7362	7385	7395	7404
	7427	7437	7446	7475	7532	7600	7724	7776	7816	7838					
MULT	1#														
NEWTST	1#	127	189	208	227	246	279	299	328	352	372	423	442	460	478
	496	520	544	568	590	614	639	663	689	713	738	762	802	821	839
	857	889	936	983	1030	1099	1146	1188	1225	1263	1308	1355	1403	1446	1492
	1548	1604	1665	1712	1766	1804	1880	1931	1973	2014	2047	2088	2134	2169	2214
	2254	2286	2318	2350	2399	2442	2509	2555	2615	2656	2700	2742	2783	2810	2837
	2878	2951	2992	3057	3100	3128	3157	3190	3231	3261	3301	3337	3377	3403	3429
	3449	3482	3515	3558	3625	3686	3730	3794	3838	3881	3930	3969	4017	4083	4150
	4199	4268	4337	4405	4477	4515	4552	4583	4612	4659	4722	4791	4849	4884	4918
	4947	4979	5013	5041	5078	5105	5144	5186	5219	5274	5409	5565	5604	5640	5677
	5713	5763	5817	5894	5917	5954	6001	6078	6141	6197	6265	6299	6366	6445	6513
	6580	6649	6732	6785	6836	6880	6943	6979	7009	7040	7070	7100	7130	7168	7201
	7243	7285	7327	7369	7411	7461	7488	7513	7553	7577	7631	7688	7738	7761	7797
	7848														
POP	1#														
PUSH	1#														
REPORT	1#														
SETPRI	1#														
SETUP	1#														
SKIP	1#														
SLASH	1#														
STARS	1#	16	27	54	56	63	74	76	97	99	105	127	129	175	189
	191	208	210	227	229	246	248	264	279	281	299	301	328	330	352
	354	372	374	396	423	425	442	444	460	462	478	480	496	498	520
	522	544	546	568	570	590	592	614	616	639	641	663	665	689	691
	713	715	738	740	762	764	787	802	804	821	823	839	841	857	859
	876	889	891	923	936	938	970	983	985	1017	1030	1032	1064	1082	1086
	1099	1101	1137	1146	1148	1182	1188	1190	1217	1225	1227	1256	1263	1265	1299
	1308	1310	1345	1355	1357	1393	1403	1405	1441	1446	1448	1487	1492	1494	1548
	1550	1604	1606	1651	1665	1667	1700	1712	1714	1753	1766	1768	1804	1806	1872
	1880	1882	1916	1931	1933	1965	1973	1975	2005	2014	2016	2047	2049	2088	2090
	2134	2136	2169	2171	2206	2214	2216	2246	2254	2256	2278	2286	2288	2310	2318
	2320	2343	2350	2352	2392	2399	2401	2434	2442	2444	2502	2509	2511	2546	2555
	2597	2607	2615	2637	2646	2656	2658	2693	2700	2702	2735	2742	2744	2777	2783
	2795	2804	2810	2812	2831	2837	2839	2870	2878	2880	2946	2951	2953	2986	2992
	2994	3052	3057	3059	3093	3100	3102	3122	3128	3130	3150	3157	3159	3179	3190
	3192	3224	3231	3233	3254	3261	3263	3292	3301	3303	3330	3337	3339	3369	3377
	3379	3396	3403	3405	3422	3429	3431	3449	3451	3482	3484	3515	3517	3558	3560
	3625	3627	3686	3688	3730	3732	3794	3796	3838	3840	3881	3883	3924	3930	3932
	3963	3969	3971	4012	4017	4019	4078	4083	4085	4143	4150	4152	4191	4199	4201
	4259	4268	4270	4329	4337	4339	4398	4405	4407	4465	4477	4479	4506	4515	4517
	4542	4552	4554	4573	4583	4585	4604	4612	4614	4650	4659	4661	4715	4722	4724
	4784	4791	4793	4841	4849	4851	4874	4884	4886	4911	4918	4920	4940	4947	4949
	4972	4979	4981	5006	5013	5015	5034	5041	5043	5071	5078	5080	5098	5105	5107
	5135	5144	5146	5177	5186	5188	5209	5219	5221	5243	5274	5276	5293	5409	5411
	5558	5565	5567	5590	5604	5606	5640	5642	5671	5679	5713	5715	5752	5763	5765
	5817	5819	5885	5894	5896	5917	5919	5954	5956	5991	6001	6003	6078	6080	6131
	6141	6143	6197	6199	6265	6267	6289	6299	6301	6366	6368	6435	6445	6447	6513
	6515	6580	6582	6649	6651	6724	6732	6734	6777	6785	6787	6829	6836	6838	6873
	6880	6882	6934	6943	6945	6979	6981	7009	7011	7040	7042	7070	7072	7100	7102
	7130	7132	7160	7168	7170	7201	7203	7243	7245	7285	7287	7327	7329	7369	7371
	7411	7413	7453	7461	7463	7483	7488	7490	7506	7513	7515	7548	7553	7555	7577

	7579	7608	7631	7633	7681	7688	7690	7732	7738	7740	7755	7761	7763	7783	7797
	7799	7848	7850	7996	8000										
SWRSU	1#														
TYPBIN	1#														
TYPDEC	1#														
TYPNAM	1#														
TYPNUM	1#														
TYPOCS	1#														
TYPOCT	1#														
TYPTXT	1#														
$$ERCD	1#	141	150	161	170	203	222	241	259	294	313	323	347	367	391
	437	455	473	491	515	540	563	585	609	634	658	683	708	733	757
	782	816	834	852	870	905	918	952	965	999	1012	1046	1059	1111	1122
	1131	1159	1177	1200	1212	1238	1251	1278	1293	1324	1340	1372	1388	1420	1436
	1464	1482	1510	1520	1535	1544	1567	1577	1591	1600	1623	1637	1646	1681	1695
	1730	1748	1783	1800	1827	1836	1847	1856	1867	1895	1911	1946	1960	1988	2000
	2030	2042	2066	2075	2084	2111	2120	2130	2156	2165	2189	2201	2227	2240	2273
	2305	2338	2370	2387	2419	2429	2462	2471	2487	2497	2531	2541	2578	2593	2603
	2635	2644	2678	2688	2721	2730	2763	2772	2799	2826	2856	2865	2891	2904	2915
	2930	2941	2972	2982	3007	3016	3026	3038	3048	3077	3089	3117	3145	3174	3209
	3219	3249	3277	3287	3317	3326	3355	3365	3392	3418	3444	3468	3477	3501	3510
	3533	3544	3553	3578	3587	3600	3610	3619	3648	3657	3671	3681	3707	3716	3725
	3751	3760	3771	3780	3789	3815	3824	3833	3858	3867	3876	3901	3910	3919	3949
	3958	3987	3998	4007	4034	4043	4055	4064	4073	4101	4110	4119	4129	4139	4168
	4177	4186	4215	4224	4236	4245	4254	4286	4295	4304	4315	4324	4355	4364	4373
	4384	4393	4423	4432	4441	4451	4460	4495	4533	4569	4600	4630	4645	4678	4693
	4710	4743	4760	4779	4808	4822	4836	4869	4905	4935	4965	4992	5001	5028	5056
	5065	5092	5120	5129	5159	5169	5202	5235	5289	5298	5312	5324	5333	5346	5359
	5373	5387	5427	5446	5465	5483	5501	5517	5533	5552	5577	5586	5621	5636	5658
	5673	5695	5709	5732	5747	5781	5797	5812	5835	5849	5863	5879	5912	5935	5950
	5972	5987	6018	6037	6046	6058	6073	6096	6111	6126	6160	6176	6192	6215	6230
	6246	6260	6283	6317	6332	6347	6362	6385	6400	6415	6430	6464	6479	6493	6509
	6532	6547	6562	6576	6599	6614	6628	6645	6668	6684	6699	6717	6753	6773	6807
	6825	6853	6869	6893	6908	6917	6929	6958	6974	6995	7004	7026	7035	7056	7065
	7086	7095	7116	7125	7146	7155	7182	7196	7220	7229	7238	7262	7271	7280	7304
	7313	7322	7346	7355	7364	7388	7397	7406	7430	7439	7448	7477	7501	7524	7534
	7541	7566	7572	7592	7602	7666	7727	7750	7779	7818	7841	7915	8003	8007	8011
	8015	8019	8023	8027	8031										
$$ERNU	1#	141	150	161	170	203	222	241	259	294	313	323	347	367	391
	437	455	473	491	515	540	563	585	609	634	658	683	708	733	757
	782	816	834	852	870	905	918	952	965	999	1012	1046	1059	1111	1122
	1131	1159	1177	1200	1212	1238	1251	1278	1293	1324	1340	1372	1388	1420	1436
	1464	1482	1510	1520	1535	1544	1567	1577	1591	1600	1623	1637	1646	1681	1695
	1730	1748	1783	1800	1827	1836	1847	1856	1867	1895	1911	1946	1960	1988	2000
	2030	2042	2066	2075	2084	2111	2120	2130	2156	2165	2189	2201	2227	2240	2273
	2305	2338	2370	2387	2419	2429	2462	2471	2487	2497	2531	2541	2578	2593	2603
	2635	2644	2678	2688	2721	2730	2763	2772	2799	2826	2856	2865	2891	2904	2915
	2930	2941	2972	2982	3007	3016	3026	3038	3048	3077	3089	3117	3145	3174	3209
	3219	3249	3277	3287	3317	3326	3355	3365	3392	3418	3444	3468	3477	3501	3510
	3533	3544	3553	3578	3587	3600	3610	3619	3648	3657	3671	3681	3707	3716	3725
	3751	3760	3771	3780	3789	3815	3824	3833	3858	3867	3876	3901	3910	3919	3949
	3958	3987	3998	4007	4034	4043	4055	4064	4073	4101	4110	4119	4129	4139	4168
	4177	4186	4215	4224	4236	4245	4254	4286	4295	4304	4315	4324	4355	4364	4373
	4384	4393	4423	4432	4441	4451	4460	4495	4533	4569	4600	4630	4645	4678	4693
	4710	4743	4760	4779	4808	4822	4836	4869	4905	4935	4965	4992	5001	5028	5056
	5065	5092	5120	5129	5159	5169	5202	5235	5289	5298	5312	5324	5333	5346	5359

	5373	5387	5427	5446	5465	5483	5501	5517	5533	5552	5577	5586	5621	5636	5658
	5673	5695	5709	5732	5747	5781	5797	5812	5835	5849	5863	5879	5912	5935	5950
	5972	5987	6018	6037	6046	6058	6073	6096	6111	6126	6160	6176	6192	6215	6230
	6246	6260	6283	6317	6332	6347	6362	6385	6400	6415	6430	6464	6479	6493	6509
	6532	6547	6562	6576	6599	6614	6628	6645	6668	6684	6699	6717	6753	6773	6807
	6825	6853	6869	6893	6908	6917	6929	6958	6974	6995	7004	7026	7035	7056	7065
	7086	7095	7116	7125	7146	7155	7182	7196	7220	7229	7238	7262	7271	7280	7304
	7313	7322	7346	7355	7364	7388	7397	7406	7430	7439	7448	7477	7501	7524	7534
	7541	7566	7572	7592	7602	7666	7727	7750	7779	7818	7841	7915	8003	8007	8011
	8015	8019	8023	8027	8031										
$$ERRO	1#	165	198	217	236	254	289	308	318	341	362	385	432	450	468
	486	509	534	603	628	652	677	702	727	751	776	811	829	847	865
	912	959	1006	1053	1126	1171	1206	1245	1287	1334	1382	1430	1476	1539	1595
	1641	1689	1742	1794	1862	1905	1954	1994	2036	2079	2125	2160	2196	2234	2267
	2299	2332	2381	2424	2492	2536	2598	2639	2683	2725	2767	2794	2821	2860	2936
	2977	3043	3083	3112	3140	3169	3214	3244	3282	3321	3360	3387	3413	3439	3472
	3505	3548	3614	3676	3720	3784	3828	3871	3914	3953	4002	4068	4134	4181	4249
	4319	4388	4455	4489	4527	4564	4595	4639	4704	4773	4830	4863	4899	4929	4959
	4996	5023	5060	5087	5124	5163	5196	5229	5382	5546	5581	5630	5667	5703	5741
	5806	5873	5906	5944	5981	6067	6120	6186	6254	6277	6356	6424	6503	6570	6639
	6711	6767	6819	6863	6923	6960	6999	7030	7060	7090	7120	7150	7191	7233	7275
	7317	7359	7401	7443	7472	7597	7773								
$$ESCA	1#														
$$LOOP	1#	148	168	201	220	239	257	292	311	321	344	365	388	435	453
	471	489	512	537	606	631	655	680	705	730	754	779	814	832	850
	868	902	915	949	962	996	1009	1043	1056	1109	1120	1129	1157	1174	1198
	1209	1236	1248	1276	1290	1322	1337	1370	1385	1418	1433	1462	1479	1507	1518
	1532	1542	1564	1575	1588	1598	1620	1635	1644	1679	1692	1728	1745	1781	1797
	1824	1834	1845	1854	1865	1893	1908	1944	1957	1986	1997	2028	2039	2063	2073
	2082	2108	2118	2128	2153	2163	2186	2199	2225	2237	2270	2302	2335	2367	2384
	2416	2427	2459	2469	2484	2495	2528	2539	2575	2590	2601	2632	2642	2675	2686
	2718	2728	2760	2770	2797	2824	2853	2863	2889	2902	2913	2928	2938	2969	2980
	3005	3014	3024	3036	3046	3074	3086	3115	3143	3172	3207	3217	3247	3275	3285
	3315	3324	3353	3363	3390	3416	3442	3465	3475	3498	3508	3530	3541	3557	3575
	3585	3597	3608	3617	3645	3655	3668	3679	3704	3714	3723	3748	3758	3768	3778
	3787	3812	3822	3831	3855	3865	3874	3898	3908	3917	3946	3956	3984	3995	4005
	4031	4041	4052	4062	4071	4098	4108	4117	4127	4137	4165	4175	4184	4213	4222
	4233	4243	4252	4263	4293	4302	4313	4322	4352	4362	4371	4382	4391	4420	4430
	4439	4449	4458	4482	4530	4567	4598	4627	4642	4675	4690	4707	4740	4757	4776
	4805	4819	4833	4866	4902	4932	4962	4990	4999	5026	5054	5063	5090	5118	5127
	5157	5166	5199	5232	5267	5296	5310	5322	5331	5344	5357	5371	5385	5443	5462
	5480	5498	5514	5530	5549	5584	5618	5633	5655	5670	5692	5706	5729	5744	5778
	5794	5809	5832	5846	5860	5876	5909	5932	5947	5969	5984	6015	6029	6043	6055
	6070	6093	6108	6123	6157	6173	6189	6212	6227	6243	6257	6280	6314	6329	6344
	6359	6382	6397	6412	6427	6461	6476	6490	6506	6529	6544	6559	6573	6596	6611
	6625	6642	6665	6681	6696	6714	6750	6770	6804	6822	6850	6866	6905	6926	6956
	6971	6993	7002	7024	7033	7054	7063	7084	7093	7114	7123	7144	7153	7180	7194
	7217	7227	7236	7259	7269	7278	7301	7311	7320	7343	7353	7362	7385	7395	7404
	7427	7437	7446	7475	7532	7600	7724	7776	7816	7838					
$$NEWT	1#	127	169	208	227	246	279	299	328	352	372	423	442	460	478
	496	520	544	568	590	614	639	663	689	713	736	762	802	821	839
	857	889	936	983	1030	1099	1146	1188	1225	1263	1308	1355	1403	1446	1492
	1548	1604	1665	1712	1766	1804	1880	1931	1973	2014	2047	2088	2134	2169	2214
	2254	2286	2318	2350	2399	2442	2509	2555	2615	2656	2700	2742	2783	2810	2837
	2878	2951	2992	3057	3100	3128	3157	3190	3231	3261	3301	3337	3377	3403	3429
	3449	3482	3515	3558	3625	3686	3730	3794	3838	3881	3930	3969	4017	4083	4150

```
          4199    4268    4337    4405    4477    4515    4552    4583    4612    4659    4722    4791    4849    4884    4918
          4947    4979    5013    5041    5078    5105    5144    5186    5219    5274    5409    5565    5604    5640    5677
          5713    5763    5817    5894    5917    5954    6001    6078    6141    6197    6265    6299    6366    6445    6513
          6580    6649    6732    6785    6836    6880    6943    6979    7009    7040    7070    7100    7130    7168    7201
          7243    7285    7327    7369    7411    7461    7488    7513    7553    7577    7631    7688    7738    7761    7797
          7848
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KTI1       1#
.SETUP      1#
.SWRHI      1#
.SACT1      1#      14#
.SAPTB      1#      14#     25
.SAPTH      1#      14#     52
.SAPTV      1#
.SASTA      1#
.SCATC      1#
.SCMTA      1#
.SDB2D      1#
.SDB2O      1#
.SDIV       1#
.SEOP       1#
.SERRO      1#
.SERRT      1#
.SMULT      1#
.SPOWE      1#
.SRAND      1#
.SRDDE      1#
.SRDOC      1#
.SREAD      1#
.SR2AZ      1#
.SSAVE      1#
.SSB2D      1#
.SSB2O      1#
.SSCOP      1#
.SSIZE      1#
.SSUPR      1#
.STRAP      1#
.STYPB      1#
.STYPD      1#
.STYPE      1#
.STYPO      1#
.$40CA      1#
.1170       1#

. ABS.  026524      000

 ERRORS DETECTED:  0

 CFKAAC.BIN,CFKAAC.LST/CRF/SOL=CFKAAC.SML,CFKAAC.P11
 RUN-TIME:  30  40  3 SECONDS
 RUN-TIME RATIO: 170/74=2.2
 CORE USED:  33K  (65 PAGES)
```