

PDP11/04

CPU TEST
MD-11-DGKAA-A

EP-DGKAA-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN US

This microfiche card contains a grid of frames. The first 10 columns of frames contain data, likely test results or program output, with some frames showing vertical bar patterns. The remaining 10 columns of frames are mostly blank or contain very faint, illegible data. The card is oriented vertically on the page.

11:00:00 -06:00 11:04 CPU TEST

170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190

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 = ADDRESS OF SUBTEST
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, \$ENVN, WHICH IS LOCATED IN THE APT MAILBOX. SETTING \$ENVN 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 ONE PASS IN LESS THAN 1 SEC.
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

1 280
2 281
3 282
4 283
5 284
6 285
7 286
8 287
9 288
10 289
11 290
12 291
13 292
14 293
15 294
16 295
17 296
18 297
19 298
20 299
21 300
22 301
23 302

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 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 MESSAGE END OF DGKAA IS PRINTED ON THE CONSOLE
TELETYPE AFTER THE FIRST PASS AND FOLLOWING EVERY 400TH PASS
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
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 FAULT WAS DETECTED.
\$TESTN IS USED IN CONJUNCTION WITH \$FATAL TO DISTINGUISH
BETWEEN SEQUENCE AND FUNCTIONAL 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 DEVELOPED DURING THE RUNNING OF THE
DIAGNOSTIC.

359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414

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

000500

000240
000007
000006
177776
177564
177566

000400

```

%
.TITLE MAINDEC-11-DGKAA 11/04 CPU TEST
.ENABLE ABS
STBOT=500
.NLIST CND,MC,MD
.LIST ME
SCOPE=NOP
R7=%7
R6=%6
PS=177776
TPS=177564
TPB=177566
.MCALL .SAPTHDR, .SAPTBL, .SACT11
.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
$SVPC=;SAVE PC

```

415
416 000046
417
418 000052
419 000400
420 000300
421
422
423
424
425 000300
426 000300 000000
427 000302 000000
428 000304 000000
429 000306 000000
430 000310 000000
431 000312 000000
432 000314 000000
433 000316 000000
434 000320
435 000320 000
436 000321 000
437 000322 000000
438 000324 000000
439 000326 000000
440
441
442
443
444
445
446 000330
447
448
449
450
451
452
453 000330
454 000024
455 000024 000200
456 000044 000044
457 000044 000330
458 000330
459
460
461
462
463 000330
464 000330 000000
465 000332 000300
466 000334 000002
467 000336 000002
468 000340 000000
469 000342 000014
470 000370

```

.=46
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD 2) SET
.=52
.WORD 0 ;;2)SET LOC.52 TO ZERO
.=5VPC ;; RESTORE PC
.=300
.SBTTL APT MAILBOX-ETABLE

;*****
;EVEN
$MAIL: .WORD AMSGTY ;; APT MAILBOX
$MSGTY: .WORD AFATAL ;; MESSAGE TYPE CODE
$FATAL: .WORD ATESTN ;; FATAL ERROR NUMBER
$TESTN: .WORD APASS ;; TEST NUMBER
$PASS: .WORD ADEVCT ;; PASS COUNT
$DEVCT: .WORD AUNIT ;; DEVICE COUNT
$UNIT: .WORD AMSGAD ;; I/O UNIT NUMBER
$MSGAD: .WORD AMSGLG ;; MESSAGE ADDRESS
$MSGLG: .WORD APTENV ;; MESSAGE LENGTH
$ETABLE: .WORD AENV ;; APT ENVIRONMENT TABLE
$ENV: .BYTE AENVM ;; ENVIRONMENT BYTE
$ENVM: .BYTE ASWREG ;; ENVIRONMENT MODE BITS
$SWREG: .WORD AUSWR ;; APT SWITCH REGISTER
$USWR: .WORD ACPUOP ;; USER SWITCHES
$CPUOP: .WORD CPUOPT ;; CPU TYPE, OPTIONS
;*
;* BIT 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
$ETEND:
.MEXIT
.SBTTL APT PARAMETER BLOCK

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

$APTHD:
$HIBTS: .WORD 0 ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 2 ;; RUN TIME OF LONGEST TEST
$PASTM: .WORD 2 ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 0 ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
.=370 $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)

```

```

471 003370 000000 000000 000000
472 000376 000000 000000 000000
473 000404 000001 000001 177777
474
475
476
477 000500
478 000200
479 000200 000167 000274
480
481 000204 012706 000500
482 000210 012702 000304
483 000214 000137
484 000216 000000
485
486 000500
487 000302
488 000304
489 000500 012737 015426 000024
490 000506 012737 000000 000306
491 000514 012737 177777 015310
492 000522 012706 000500
493 000526 012702 000304
494 000532 012737 000000 000304
495 000540 012737 000000 000302
496 000546 012737 000000 000300

```

```

0.0.0,0.0.0
1,1,-1
.=500
*****
;SET UP STARTING ADDRESS
.SX=
.=200
JMP START
MOV #STBOT,R6 ;SET STACK POINTER
MOV #STSTN,R2 ;SET MAILBOX POINTER
JMP @PC+ ;JUMP TO SUBTEST
0 ;ADDR. OF SUBTEST GOES HERE

.=.SX
$ERROR=$FATAL
$STSTNM=$STSTN
START: MOV #PWRDN,@#24 ;SET UP FOR POWER FAIL
MOV #0,@#SPASS ;CLEAR PASS COUNT
MOV #-1,@#PASSPT ;SET PRINT COUNTER
RESTRT: MOV #STBOT,R6 ;INITIALIZE STACK POINTER
MOV #STSTN,R2 ;SET UP POINTER TO MESSAGE TYPE
MOV #0,@#STSTNM ;CLEAR TEST NUMBER
MOV #0,@#SEERROR ;CLEAR ERROR NUMBER
MOV #0,@#SMSGTY ;CLEAR MESSAGE TYPE(FOR APT)

```

```

-97
438
4-19
500 000554 005212
501 000556 022712 000001
502 000562 001024
503 000564 000257
504 000566 001401
505 000570 000404
506
507
508
509
510 000572
511 000572 012742 000001
512 000576 005242
513 000600 000000
514 000602
515 000602 001004
516
517
518
519
520 000604 012742 000002
521 000610 005242
522 000612 000000
523 000614 000264
524 000616 001001
525 000620 000404
526
527
528
529
530 000622
531 000622 012742 000003
532 000626 005242
533 000630 000000
534 000632
535 000632 001404
536
537
538
539
540 000634 012742 000004
541 000640 005242
542 000642 000000
543

```

```

;*****
;TEST 1 CHECK BRANCHES ON Z BIT
;*****
TST1: INC (R2) ;UPDATE TEST NUMBER
      CMP #1,(R2) ;SEQUENCE ERROR?
      BNE TST2-10 ;BR TO ERROR HALT ON SEQ ERRCR
      CCC ;CLEAR ALL CONDITION CODES
      BEQ BR1 ;SHOULD BRANCH
      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 ;====

BR1: MOV #1, -(R2) ;MOVE TO MAILBOX # ***** 1 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;SHOULD HAVE BRANCHED: Z=0

BR2: BNE BR3
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 770 ;====

BR3: MOV #2, -(R2) ;MOVE TO MAILBOX # ***** 2 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT
      SEZ
      BNE BR4
      BR BR5
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; BRANCH INSTRUCTION AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; FOLLOWING W/ 760 ;====

BR4: MOV #3, -(R2) ;MOVE TO MAILBOX # ***** 3 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;SHOULD NOT HAVE BRANCHED HERE ON Z=1

BR5: BEQ TST2
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 754 ;====

      MOV #4, -(R2) ;MOVE TO MAILBOX # ***** 4 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;SHOULD HAVE BRANCHED ON Z=1
      ; OR SEQUENCE ERROR

```

544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599

: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

```
*****
TST2:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #2,(R2)     ;SEQUENCE ERROR?
        BNE     TST3-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #0,2#0      ;MOVE ZEROES THRU ADDRESS LINES, DATA
                                ;LINES AND INTERNAL PATHS
                                ;SUCCESSFUL?
        TST     2#0
        BEQ     TST3
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
                                ; CONDITIONAL BRANCH INST. AND
                                ; REPLACE THE MOVE INSTRUCTION
                                ; WHICH FOLLOWS W/ 772
                                ;=====
        MOV     #5,-(R2)     ;MOVE TO MAILBOX # ***** 5 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;DATA INCORRECT
                                ; OR SEQUENCE ERROR
*****
```

:TEST 3 TEST OF PATTERN 125252 IN DATA PATH

```
*****
TST3:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #3,(R2)     ;SEQUENCE ERROR?
        BNE     TST4-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #125252,2#0  ;MOVE ALTERNATING ONES AND ZEROES
                                ;THRU DATA PATHS
                                ;SUCCESSFUL
        CMP     #125252,2#0
        BEQ     TST4
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
                                ; CONDITIONAL BRANCH INST. AND
                                ; REPLACE THE MOVE INSTRUCTION
                                ; WHICH FOLLOWS W/ 771
                                ;=====
        MOV     #6,-(R2)     ;MOVE TO MAILBOX # ***** 6 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;DATA INCORRECT
                                ; OR SEQUENCE ERROR
*****
```

:TEST 4 TEST OF PATTERN 052525 IN DATA PATH

633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688

: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
: RO 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.

: 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 RO ARE SUCCESSFUL BUT THE TESTS WITH THE OTHER
: REGISTERS FAIL, SUSPECT THE REGISTER SELECT LINES AND THEN THE SCRATCH
: PAD ITSELF.

:TEST 6 TEST IF RO CAN HOLD ALL ZEROES

```
TST6:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #6,(R2)     ;SEQUENCE ERROR?
        BNE      TST7-10    ;BR TO ERROR HALT ON SEQ ERROR
        MOV      #0,RO       ;MOVE ZEROES TO RO
        TST      RO         ;SUCCESSFUL?
        BEQ      TST7
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
        ;                CONDITIONAL BRANCH INST. AND <===
        ;                REPLACE THE MOVE INSTRUCTION <===
        ;                WHICH FOLLOWS W/ 774 <===
        MOV      #11,-(R2)   ;MOVE TO MAILBOX # ***** 11 *****
        INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                ;RO NOT 0
        ; OR SEQUENCE ERROR
```

:TEST 7 TEST IF RO CAN HOLD ONES AND ZEROES

```
TST7:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #7,(R2)     ;SEQUENCE ERROR?
        BNE      TST10-10   ;BR TO ERROR HALT ON SEQ ERROR
        MOV      #125252,RO  ;MOVE ALTERNATING ONES AND ZEROES TO RO
        CMP      RO,#125252  ;SUCCESSFUL?
        BEQ      TST10
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
        ;                CONDITIONAL BRANCH INST. AND <===
        ;                REPLACE THE MOVE INSTRUCTION <===
```

MO1

```

689
690 001104 012742 000012      MOV      #12,-(R2)      ; WHICH FOLLOWS W/ 773
691 001110 005242              INC      -(R2)        ; MOVE TO MAILBOX * ***** 12 *****
692 001112 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
693                                     ; RO NOT 125252
694                                     ; OR SEQUENCE ERROR
695
696 ;*****
697 ;TEST 10      TEST IF RO CAN HOLD ZEROES AND ONES
698 ;*****
698 001114 005212      TST10:  INC      (R2)      ; UPDATE TEST NUMBER
699 001116 022712 000010      CMP      #10,(R2)      ; SEQUENCE ERROR?
700 001122 001005      BNE      TST11-10      ; BR TO ERROR HALT ON SEQ ERROR
701 001124 012700 052525      MOV      #052525,RO    ; MOVE ALTERNATING ZEROES AND ONES TO RO
702 001130 020027 052525      CMP      RO,#052525    ; SUCCESSFUL?
703 001134 001404      BEQ      TST11
704                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
705                                     ; CONDITIONAL BRANCH INST. AND <====
706                                     ; REPLACE THE MOVE INSTRUCTION <====
707                                     ; WHICH FOLLOWS W/ 773 <====
708 001136 012742 000013      MOV      #13,-(R2)      ; MOVE TO MAILBOX * ***** 13 *****
709 001142 005242              INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
710 001144 000000              HALT                    ; RO NOT 52525
711                                     ; OR SEQUENCE ERROR
712
713 ;*****
714 ;TEST 11      TEST IF RO CAN HOLD ALL ONES
715 ;*****
716 001146 005212      TST11:  INC      (R2)      ; UPDATE TEST NUMBER
717 001150 022712 000011      CMP      #11,(R2)      ; SEQUENCE ERROR?
718 001154 001005      BNE      TST12-10      ; BR TO ERROR HALT ON SEQ ERROR
719 001156 012700 177777      MOV      #177777,RO    ; MOVE ALL ONES TO RO
720 001162 020027 177777      CMP      RO,#177777    ; SUCCESSFUL?
721 001166 001404      BEQ      TST12
722                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
723                                     ; CONDITIONAL BRANCH INST. AND <====
724                                     ; REPLACE THE MOVE INSTRUCTION <====
725                                     ; WHICH FOLLOWS W/ 773 <====
726 001170 012742 000014      MOV      #14,-(R2)      ; MOVE TO MAILBOX * ***** 14 *****
727 001174 005242              INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
728 001176 000000              HALT                    ; RO NOT 177777
729                                     ; OR SEQUENCE ERROR
730
731 ;*****
732 ;TEST 12      TEST IF R1 CAN HOLD A ONE IN ALL BITS
733 ;*****
734 001200 005212      TST12:  INC      (R2)      ; UPDATE TEST NUMBER
735 001202 022712 000012      CMP      #12,(R2)      ; SEQUENCE ERROR?
736 001206 001006      BNE      TST13-10      ; BR TO ERROR HALT ON SEQ ERROR
737 001210 012701 000001      MOV      #1,R1        ; SET BIT 0
738 001214 000241      CLC                    ; CLEAR C-BIT
739 001216 006101      REG1:  ROL      R1        ; ROTATE 1 POSITION
740 001220 103376      BCC      REG1          ; ALL DONE
741 001222 001404      BEQ      TST13
742                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
743                                     ; CONDITIONAL BRANCH INST. AND <====
744                                     ; REPLACE THE MOVE INSTRUCTION <====

```

```

745
746 001224 012742 000015      MOV    #15, -(R2)      ; MOVE TO MAILBOX # ***** 15 *****
747 001230 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
748 001232 000000              HALT                   ; FAILURE WITH R1
749                                     ; OR SEQUENCE ERROR
750
751 ;*****:*****
752 ;TEST 13      TEST IF R2 CAN HOLD A ONE IN ALL BITS
753 ;*****:*****
754 001234 005212              TST13: INC    (R2)          ; UPDATE TEST NUMBER
755 001236 022712 000013      CMP    #13, (R2)      ; SEQUENCE ERROR?
756 001242 001006              BNE   REG2A-14        ; BR TO ERROR HALT ON SEQ ERROR
757 001244 012702 000001      MOV    #1, R2         ; SET BIT 0
758 001250 000241              CLC                   ; CLEAR C-BIT
759 001252 006102              REG2:  ROL    R2         ; ROTATE 1 POSITION
760 001254 103376              BCC   REG2            ; ALL DONE
761 001256 001406              BEQ   REG2A
762                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
763                                     ; BRANCH INSTRUCTION AND <====
764                                     ; REPLACE THE MOVE INSTRUCTION <====
765                                     ; FOLLOWING W/ 771 <====
766 001260 012702 000304      MOV    #STESTN, R2    ; RESTORE POINTER
767 001264 012742 000016      MOV    #16, -(R2)     ; MOVE TO MAILBOX # ***** 16 *****
768 001270 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
769 001272 000000              HALT                   ; FAILURE WITH R2
770 001274 012702 000304      REG2A: MOV    #STESTN, R2 ; RESTORE POINTER
771 ;*****:*****
772 ;TEST 14      TEST IF R3 CAN HOLD A ONE IN ALL BITS
773 ;*****:*****
774 001300 005212              TST14: INC    (R2)          ; UPDATE TEST NUMBER
775 001302 022712 000014      CMP    #14, (R2)      ; SEQUENCE ERROR?
776 001306 001006              BNE   TST15-10        ; BR TO ERROR HALT ON SEQ ERROR
777 001310 012703 000001      MOV    #1, R3         ; SET BIT 0
778 001314 000241              CLC                   ; CLEAR C-BIT
779 001316 006103              REG3:  ROL    R3         ; ROTATE 1 POSITION
780 001320 103376              BCC   REG3            ; ALL DONE
781 001322 001404              BEQ   TST15
782                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
783                                     ; CONDITIONAL BRANCH INST. AND <====
784                                     ; REPLACE THE MOVE INSTRUCTION <====
785                                     ; WHICH FOLLOWS W/ 772 <====
786 001324 012742 000017      MOV    #17, -(R2)     ; MOVE TO MAILBOX # ***** 17 *****
787 001330 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
788 001332 000000              HALT                   ; FAILURE WITH R3
789                                     ; OR SEQUENCE ERROR
790
791 ;*****:*****
792 ;TEST 15      TEST IF R4 CAN HOLD A ONE IN ALL BITS
793 ;*****:*****
794 001334 005212              TST15: INC    (R2)          ; UPDATE TEST NUMBER
795 001336 022712 000015      CMP    #15, (R2)      ; SEQUENCE ERROR?
796 001342 001006              BNE   TST16-10        ; BR TO ERROR HALT ON SEQ ERROR
797 001344 012704 000001      MOV    #1, R4         ; SET BIT 0
798 001350 000241              CLC                   ; CLEAR C-BIT
799 001352 006104              REG4:  ROL    R4         ; ROTATE 1 POSITION
800 001354 103376              BCC   REG4            ; ALL DONE

```

```

001356 001404 BEQ TST16
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 772
001360 012742 000020 MOV #20, -(R2) ; MOVE TO MAILBOX # ***** 20 *****
001364 005242 INC -(R2) ; SET MSGTYP TO FATAL ERROR
001366 000000 HALT ; FAILURE WITH R4
; OR SEQUENCE ERROR

;*****
;TEST 16 TEST IF R5 CAN HOLD A ONE IN ALL BITS
;*****
814 001370 005212 TST16: INC (R2) ; UPDATE TEST NUMBER
815 001372 022712 000016 CMP #16, (R2) ; SEQUENCE ERROR?
816 001376 001006 BNE TST17-10 ; BR TO ERROR HALT ON SEQ ERROR
817 001400 012705 000001 MOV #1, R5 ; SET BIT 0
818 001404 000241 CLC ; CLEAR C-BIT
819 001406 006105 REG5: ROL R5 ; ROTATE 1 POSITION
820 001410 103376 BCC REG5 ; ALL DONE
821 001412 001404 BEQ TST17
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 772
826 001414 012742 000021 MOV #21, -(R2) ; MOVE TO MAILBOX # ***** 21 *****
827 001420 005242 INC -(R2) ; SET MSGTYP TO FATAL ERROR
828 001422 000000 HALT ; FAILURE WITH R5
; OR SEQUENCE ERROR

;*****
;TEST 17 TEST IF R6 CAN HOLD A ONE IN ALL BITS
;*****
834 001424 005212 TST17: INC (R2) ; UPDATE TEST NUMBER
835 001426 022712 000017 CMP #17, (R2) ; SEQUENCE ERROR?
836 001432 001006 BNE TST20-10 ; BR TO ERROR HALT ON SEQ ERROR
837 001434 012706 000001 MOV #1, R6 ; SET BIT 0
838 001440 000241 CLC ; CLEAR C-BIT
839 001442 006106 REG6: ROL R6 ; ROTATE 1 POSITION
840 001444 103376 BCC REG6 ; ALL DONE
841 001446 001404 BEQ TST20
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 772
846 001450 012742 000022 MOV #22, -(R2) ; MOVE TO MAILBOX # ***** 22 *****
847 001454 005242 INC -(R2) ; SET MSGTYP TO FATAL ERROR
848 001456 000000 HALT ; FAILURE WITH R6
; OR SEQUENCE ERROR

```

850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906

```

:*****
:SBTTL PSW TESTS
:
: THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
: PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSE AND THAT THE
: PSW ADDRESSING LOGIC IS FUNCTIONING. MOVE AND COMPARE INSTRUCTIONS
: ARE USED TO TEST THAT THE PSE 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 20 TEST IF PSW WILL HOLD ZEROES
:*****
TST20: INC (R2) ;UPDATE TEST NUMBER
: CMP #20,(R2) ;SEQUENCE ERROR?
: BNE TST21-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #STBOT,R6
: MOV #0,#PS ;SET PSW TO ZERO
: TST #PS ;SUCCESSFUL
: BEQ TST21
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
: CONDITIONAL BRANCH INST. AND (====
: REPLACE THE MOVE INSTRUCTION (====
: WHICH FOLLOWS W/ 770 (====
: MOV #23,-(R2) ;MOVE TO MAILBOX # ***** 23 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;PSW NOT 0
: OR SEQUENCE ERROR
:*****
:TEST 21 TEST IF PSW WILL HOLD ONES AND ZEROES
:*****
TST21: INC (R2) ;UPDATE TEST NUMBER
: CMP #21,(R2) ;SEQUENCE ERROR?
: BNE TST22-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #252,#PS ;MOVE ALT. ONES AND ZEROES TO PSW
: CMP #PS,#252 ;SUCCESSFUL?
: BEQ TST22
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
: CONDITIONAL BRANCH INST. AND (====
: REPLACE THE MOVE INSTRUCTION (====
: WHICH FOLLOWS W/ 771 (====
: MOV #24,-(R2) ;MOVE TO MAILBOX # ***** 24 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;PSW NOT 252
: OR SEQUENCE ERROR
:*****
:TEST 22 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
:*****

```

```

001460 005212
001462 022712 000020
001466 001010
001470 012706 000500
001474 012737 000000 177776
001502 005737 177776
001506 001404

001510 012742 000023
001514 005242
001516 000000

001520 005212
001522 022712 000021
001526 001007
001530 012737 000252 177776
001536 023727 177776 000252
001544 001404

001546 012742 000024
001552 005242
001554 000000

```

```

907 001556 005212          TST22: INC      (R2)          ; UPDATE TEST NUMBER
908 001560 022712 000022  CMP      #22,(R2)       ; SEQUENCE ERROR?
909 001564 001007          BNE     TST23-10       ; BR TO ERROR HALT ON SEQ ERROR
910 001566 012737 000105 177776 MOV     #105,2#PS      ; MOVE ALT. ONES AND ZEROES TO PSW
911 001574 023727 177776 000105 CMP     2#PS,#105      ; SUCCESSFUL?
912 001602 001404          BEQ     TST23          ;
913          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
914          ; CONDITIONAL BRANCH INST. AND                <====
915          ; REPLACE THE MOVE INSTRUCTION                <====
916          ; WHICH FOLLOWS W/ 771                          <====
917 001604 012742 000025          MOV     #25,-(R2)      ; MOVE TO MAILBOX # ***** 25 *****
918 001610 005242          INC     -(R2)         ; SET MSGTYP TO FATAL ERROR
919 001612 000000          HALT                    ; PSW NOT 105
920          ; OR SEQUENCE ERROR

```

```

;*****
;TEST 23      TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
;*****

```

```

925 001614 005212          TST23: INC      (R2)          ; UPDATE TEST NUMBER
926 001616 022712 000023  CMP     #23,(R2)       ; SEQUENCE ERROR?
927 001622 001007          BNE     TST24-10       ; BR TO ERROR HALT ON SEQ ERROR
928 001624 012737 000357 177776 MOV     #357,2#PS      ; MOVE ONES TO PSW
929 001632 023727 177776 000357 CMP     2#PS,#357      ; SUCCESSFUL
930 001640 001404          BEQ     TST24          ;
931          ; TC SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
932          ; CONDITIONAL BRANCH INST. AND                <====
933          ; REPLACE THE MOVE INSTRUCTION                <====
934          ; WHICH FOLLOWS W/ 771                          <====
935 001642 012742 000026          MOV     #26,-(R2)      ; MOVE TO MAILBOX # ***** 26 *****
936 001646 005242          INC     -(R2)         ; SET MSGTYP TO FATAL ERROR
937 001650 000000          HALT                    ; PSW NOT 357
938          ; OR SEQUENCE ERROR

```

958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994

001652 005212
001654 022712 000024
001660 001012
001662 000241
001664 012737 000001 000000
001672 006137 000030
001676 022737 000002 000000
001704 001404

001706 012742 000027
001712 005242
001714 000000

001716 005212
001720 022712 000025
001724 001017
001726 012737 000000 000000
001734 000261
001736 006137 000000
001742 103014

001744 012742 000030
001750 005242
001752 000000

001754 022737 000001 000000
001762 001404

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 24 SHIFT BIT 0 TO BIT 1

TST24: INC (R2) ; UPDATE TEST NUMBER
CMP #24, (R2) ; SEQUENCE ERROR?
BNE TST25-10 ; BR TO ERROR HALT ON SEQ ERROR
CLC ; CLEAR CARRY BIT
MOV #1, A0 ; LOAD A 1
ROL A0 ; SHIFT LEFT
CMP #2, A0 ; SUCCESSFUL
BEQ TST25

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
MOV #27, -(R2) ; MOVE TO MAILBOX # ***** 27 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; BIT 1 NOT SET
; OR SEQUENCE ERROR

TEST 25 SHIFT CARRY INTO BIT 0

TST25: INC (R2) ; UPDATE TEST NUMBER
CMP #25, (R2) ; SEQUENCE ERROR?
BNE TST26-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #0, A0 ; CLEAR LOCATION
SEC ; SET CARRY
ROL A0 ; ROTATE CARRY BIT TO BIT 0
BCC TST26

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV #30, -(R2) ; MOVE TO MAILBOX # ***** 30 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CARRY CLEAR
; OR SEQUENCE ERROR
CMP #1, A0 ; BIT 0 SET
BEQ TST26

```

995
996
997
998
999 001764 012742 000031      MOV      #31,-(R2)
1000 001770 005242      INC      -(R2)
1001 001772 000000      HALT
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 761
; MOVE TO MAILBOX # ***** 31 *****
; SET MSGTYP TO FATAL ERROR
; BIT 0 NOT SET
; OR SEQUENCE ERROR

;*****
;TEST 26      LEFT SHIFT FROM BIT 0 TO C-BIT
;*****
1007 001774 005212      TST26:  INC      (R2)
1008 001776 022712 000026      CMP      #26,(R2)
1009 002002 001010      BNE     TST27-10
1010 002004 012737 000001 000000      MOV      #1,2#0
1011 002012 000241      CLC
1012 002014 006137 000000      SHL:    ROL      2#0
1013 002020 103375      BCC     SHL
1014 002022 001404      BEQ     TST27
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 770
; MOVE TO MAILBOX # ***** 32 *****
; SET MSGTYP TO FATAL ERROR
; LEFT SHIFTING LOGIC FAILED
; OR SEQUENCE ERROR

;*****
;TEST 27      SHIFT BIT 15 TO BIT 14
;*****
1027 002034 005212      TST27:  INC      (R2)
1028 002036 022712 000027      CMP      #27,(R2)
1029 002042 001012      BNE     TST30-10
1030 002044 012737 100000 000000      MOV      #100000,2#0
1031 002052 000241      CLC
1032 002054 006037 000000      ROR      2#0
1033 002060 022737 040000 000000      CMP      #40000,2#0
1034 002066 001404      BEQ     TST30
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 766
; MOVE TO MAILBOX # ***** 33 *****
; SET MSGTYP TO FATAL ERROR
; BIT 14 NOT SET
; OR SEQUENCE ERROR

;*****
;TEST 30      RIGHT SHIFT FROM BIT 15 TO C-BIT
;*****
1047 002100 005212      TST30:  INC      (R2)
1048 002102 022712 000030      CMP      #30,(R2)
1049 002106 001010      BNE     TST31-10
1050 002110 012737 100000 000000      MOV      #100000,2#0
; SET BIT 15

```

```

:051 002116 000241
:052 002120 005037 000000
:053 002124 103375
:054 002126 001404
:055
:056
:057
:058
:059 002130 012742 000034
:060 002134 005242
:061 002136 000000
:062

```

S-P:

```

CLC
ROR
BCC
BEG
J#0
SHR
TST3!

```

```

: CLEAR C-BIT
: ROTATE RIGHT ONE POSITION
: BRANCH IF C-BIT CLEAR

```

```

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

```

```

: MOVE TO MAILBOX # ***** 34 *****
: SET MSGTYP TO FATAL ERROR
: RIGHT SHIFT LOGIC FAILED
: OR SEQUENCE ERROR

```

```

(====
(====
(====
(====

```

SECT. 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 31 TEST BRANCHES AROUND Z-BIT

1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081 002140 005212
1082 002142 022712 000031
1083 002146 001014
1084
1085 002150 000257
1086 002152 000264
1087 002154 001001
1088 002156 001404
1089
1090
1091
1092
1093 002160
1094 002160 012742 000035
1095 002164 005242
1096 002166 000000
1097
1098 002170 000277
1099 002172 000244
1100 002174 001401
1101 002176 001004
1102
1103
1104
1105
1106 002200
1107 002200 012742 000036
1108 002204 005242
1109 002206 000000
1110

TST31: INC (R2) ;UPDATE TEST NUMBER
 CMP #31,(R2) ;SEQUENCE ERROR?
 BNE TST32-10 ;BR TO ERROR HALT ON SEQ ERROR
 ;FIRST WITH Z-BIT ON
 CCC ;CC=0100: JUST Z-BIT
 SEZ
 BNE BRZ1 ;CHECK OPPOSITE CONDITION
 BEQ BRZ2
 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
 ; CONDITIONAL BRANCH INST. AND <====
 ; REPLACE THE MOVE INSTRUCTION <====
 ; WHICH FOLLOWS W/ 774 <====

ERZ1: MOV #35,-(R2) ;MOVE TO MAILBOX # ***** 35 *****
 INC -(R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;IMPROPER BR W/ Z=1
 ;CHECK WITH Z-BIT OFF

BRZ2: SCC ;CC=1011: ALL BUT Z-BIT
 CLZ
 BEQ BRZ3
 BNE TST32
 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
 ; CONDITIONAL BRANCH INST. AND <====
 ; REPLACE THE MOVE INSTRUCTION <====
 ; WHICH FOLLOWS W/ 764 <====

BRZ3: MOV #36,-(R2) ;MOVE TO MAILBOX # ***** 36 *****
 INC -(R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;IMPROPER BR W/ Z=0
 ; OR SEQUENCE ERROR

1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157

002210 005212
002212 022712 000032
002216 001014

002220 000257
002222 000270
002224 100001
002226 100404

002230
002230 012742 000037
002234 005242
002236 000000

002240 000277
002242 000250
002244 100401
002246 100004

002250
002250 012742 000040
002254 005242
002256 000000

```
*****
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 32 TEST BRANCHES AROUND N-BIT
*****
TST32: INC (R2) ;UPDATE TEST NUMBER
CMP #32,(R2) ;SEQUENCE ERROR?
BNE TST33-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH N-BIT ON
;CC=1000: JUST N-BIT
CCC
SEN
BPL BRN1 ;CHECK OPPOSITE CONDITION
BMI BRN2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====

BRN1: MOV #37,-(R2) ;MOVE TO MAILBOX # ***** 37 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ N=1
;CHECK WITH N-BIT OFF

BRN2: SCC ;CC=0111
CLN
BMI BRN3 ;CHECK OPPOSITE CONDITION
BPL TST33
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====

BRN3: MOV #40,-(R2) ;MOVE TO MAILBOX # ***** 40 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ N=0
; OR SEQUENCE ERROR
```

1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204

002260 005212
002262 022712 000033
002266 001014
002270 000257
002272 000262
002274 102001
002276 102404
002300
002300 012742 000041
002304 005242
002306 000000
002310 000277
002312 000242
002314 102401
002316 102004
002320
002320 012742 000042
002324 005242
002326 000000

```
*****
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 33 TEST BRANCHES AROUND V-BIT
*****
TST33: INC (R2) ;UPDATE TEST NUMBER
CMP #33,(R2) ;SEQUENCE ERROR?
BNE TST34-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH V-BIT ON
CCC ;CC=0010: JUST V-BIT
SEV
BVC BRV1 ;CHECK OPPOSITE CONDITION
BVS BRV2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====
BRV1: MOV #41,-(R2) ;MOVE TO MAILBOX # ***** 41 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ V=1
;CHECK WITH V-BIT OFF
BRV2: SCC ;CC=1101: ALL BVT V-BIT
CLV
BVS BRV3 ;CHECK OPPOSITE CONDITION
BVC TST34
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
BRV3: MOV #42,-(R2) ;MOVE TO MAILBOX # ***** 42 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ V=0
; OR SEQUENCE ERROR
```

1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251

002330 005212
002332 022712 000034
002336 001014
002340 000257
002342 000261
002344 103001
002346 103404
002350
002350 012742 000043
002354 005242
002356 000000
002360 000277
002362 000242
002364 102401
002366 100404
002370
002370 012742 000044
002374 005242
002376 000000

```
*****
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 34 TEST BRANCHES AROUND C-BIT
*****
TST34: INC (R2) ;UPDATE TEST NUMBER
CMP #34,(R2) ;SEQUENCE ERROR?
BNE TST35-10 ;BR TO ERROR HALT ON SEG ERROR
;FIRST WITH C-BIT ON
CCC ;CC=0001: JUST C-BIT
SEC
BCC BRC1 ;CHECK OPPOSITE CONDITION
BCS BRC2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====
BRC1: MOV #43,-(R2) ;MOVE TO MAILBOX # ***** 43 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ C=1
;CHECK WITH V-BIT OFF
BRC2: SCC ;CC=1110
CLV
BVS BRC3 ;CHECK OPPOSITE CONDITION
BMI TST35
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
BRC3: MOV #44,-(R2) ;MOVE TO MAILBOX # ***** 44 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ C=0
; OR SEQUENCE ERROR
```

1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307

: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 35 TEST MODE 0 USING SOP INST.

TST35: INC (R2) ;UPDATE TEST NUMBER
CMP #35,(R2) ;SEQUENCE ERROR?
BNE TST36-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;TRY THE CLEAR INST.
BEQ SOP0A

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 776 <====

MOV #45,-(R2) ;MOVE TO MAILBOX # ***** 45 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP0A: INC R0 ;TRY THE INCREMENT INST.
NEG R0 ;TRY THE NEGATE INST.
BMI SOP0B

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

002400 005212
002402 022712 000035
002406 001017
002410 005000
002412 001404

002414 012742 000045
002420 005242
002422 000000
002424 005200
002426 005400
002430 100404

```

1308                                     ; REPLACE THE MOVE INSTRUCTION <====
1309                                     ; WHICH FOLLOWS W/ 767 <====
1310 002432 012742 000046                MOV #46, -(R2) ; MOVE TO MAILBOX # ***** 46 *****
1311 002436 005242                        INC -(R2) ; SET MSGTYP TO FATAL ERROR
1312 002440 000000                        HALT ; NEGATE DID NOT SET N-BIT
1313 002442 005100                SOP08: COM R0 ; TRY COMPLEMENT INST.
1314 002444 001404                BEQ TST36
1315                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1316                                     ; CONDITIONAL BRANCH INST. AND <====
1317                                     ; REPLACE THE MOVE INSTRUCTION <====
1318                                     ; WHICH FOLLOWS W/ 761 <====
1319 002446 012742 000047                MOV #47, -(R2) ; MOVE TO MAILBOX # ***** 47 *****
1320 002452 005242                        INC -(R2) ; SET MSGTYP TO FATAL ERROR
1321 002454 000000                        HALT ; CUMMULATIVE RESULT OF CLR, INC, NEG AND COM INSTS. FAILED
1322                                     ; OR SEQUENCE ERROR
1323
1324
1325 *****
1326
1327 THIS TEST INTRODUCES THE REMAINING SOP INSTRUCTIONS AND TESTS
1328 THEM IN MODE 0. THE PURPOSE IS TO PROVIDE A BASELINE OF
1329 INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS. SINCE THE MICROCODE FOR
1330 THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
1331 SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
1332 FUNCTIONING.
1333 *****
1334 TEST 36 TEST REMAINDER OF SOP INSTS IN MODE 0
1335 *****
1336
1337 002456 005212 000036                TST36: INC (R2) ; UPDATE TEST NUMBER
1338 002460 022712                        CMP #36, (R2) ; SEQUENCE ERROR?
1339 002464 001020                        BNE TST37-10 ; BR TO ERROR HALT ON SEQ ERROR
1340 002466 005000                        CLR R0 ; INITIALIZE
1341 002470 005300                        DEC R0 ; TRY DECREMENT INST.
1342 002472 100404                        BMI SOP0C
1343                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1344                                     ; CONDITIONAL BRANCH INST. AND <====
1345                                     ; REPLACE THE MOVE INSTRUCTION <====
1346                                     ; WHICH FOLLOWS W/ 775 <====
1347 002474 012742 000050                MOV #50, -(R2) ; MOVE TO MAILBOX # ***** 50 *****
1348 002500 005242                        INC -(R2) ; SET MSGTYP TO FATAL ERROR
1349 002502 000000                        HALT ; N-BIT NOT SET ON DEC
1350 002504 000261                SOP0C: SEC ; INITIALIZE CARRY
1351 002506 005500                        ADC R0 ; TRY ADD CARRY INST
1352 002510 001006                        BNE SOP0D
1353 002512 000261                SEC ; INITIALIZE CARRY
1354 002514 005600                        SBC R0 ; TRY SUBTRACT-CARRY INST
1355 002516 100003                        BPL SOP0D
1356 002520 005400                        NEG R0
1357 002522 005300                        DEC R0
1358 002524 001404                        BEQ TST37
1359                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1360                                     ; CONDITIONAL BRANCH INST. AND <====
1361                                     ; REPLACE THE MOVE INSTRUCTION <====
1362                                     ; WHICH FOLLOWS W/ 760 <====
1363 002526                SOP0D:

```

1364 002526 012742 000051
1365 002532 005242
1366 002534 000000
1367

MOV #51 -(R2)
INC -(R2)
HALT

; MOVE TO MAILBOX # ***** 51 *****
; SET MSGTYP TO FATAL ERROR
; CUMMULATIVE RESULT OF ADC, SBC, NEG AND DEC INSTS. FAILE
; OR SEQUENCE ERROR

1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402

002536 005212
002540 022712 000037
002544 001012
002546 :05C00
002550 001404

002552 012742 000052
002556 005242
002560 000000
002562 105100
002564 100002
002566 105200
002570 001404

002572 012742 000053
002576 005242
002600 000000

```
*****
: THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
: THE MODE 0 BYTE MICROCODE IS TESTED. THE METHOD AND SEQUENCE
: OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
*****
: TEST 37 TEST MODE 0 EVEN BYTE USING SOP INST
*****
:ST37: INC (R2) ;UPDATE TEST NUMBER
: CMP #37,(R2) ;SEQUENCE ERROR?
: BNE TST40-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLRB R0 ;TRY CLEARING EVEN BYTE OF REGISTER
: BEQ SOPB0A
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
: CONDITIONAL BRANCH INST. AND (====
: REPLACE THE MOVE INSTRUCTION (====
: WHICH FOLLOWS W/ 776 (====
: MOV #52,-(R2) ;MOVE TO MAILBOX # ***** 52 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;CLRB DID NOT SET Z-BIT
SOPB0A: COMB R0 ;TRY SETTING EVEN BYTE OF REGISTER
: BPL SOPB0B
: INCB R0 ;TRY INCREMENTING EVEN BYTE OF REGISTER)
: BEQ TST40
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
: CONDITIONAL BRANCH INST. AND (====
: REPLACE THE MOVE INSTRUCTION (====
: WHICH FOLLOWS W/ 766 (====
SOPB0B: MOV #53,-(R2) ;MOVE TO MAILBOX # ***** 53 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
: OR SEQUENCE ERROR
```


1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483

002652 005212
002654 022712 000041
002660 001017
002662 005000
002664 005010
002666 005110
002670 105010
002672 001404

002674 012742 000056
002700 005242
002702 000000
002704 005210
002706 100004
002710 105410
002712 100002
002714 105210
002716 001404

002720 012742 000057
002724 005242
002726 000000

```
*****
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 41 TEST MODE 1 EVEN BYTE USING SOP INST
*****
TST41: INC (R2) ;UPDATE TEST NUMBER
CMP #41,(R2) ;SEQUENCE ERROR?
BNE TST42-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;INITIALIZE R0
CLR (R0) ;INITIALIZE LOC. 0
COM (R0)
CLRB (R0) ;TRY TO CLEAR BYTE 0
BEQ SOPB1A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
MOV #56,-(R2) ;MOVE TO MAILBOX # ***** 56 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLRB DID NOT SET Z-BIT
SOPB1A: INC (R0) ;INCREMENT TO TEST WORD
BPL SOPB1B
NEGB (R0) ;NEGATE: ODD BYTE=377
BPL SOPB1B
INCB (R0) ;INCREMENT ODD BYTE=0
BEQ TST42
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
SOPB1B: MOV #57,-(R2) ;MOVE TO MAILBOX # ***** 57 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST
; OR SEQUENCE ERROR
*****
```

144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

002730 005212
002732 022712 000042
002734 001021
002740 005000
002742 005010
002744 005110
002746 005200
002750 105010
002752 001404

002754 012742 000060
002760 005242
002762 000000
002764 005300
002766 005210
002770 005200
002772 105410
002774 100002
002776 105210
003000 001404

003002 012742 000061
003006 005242
003010 000000

```
*****  
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 UNADDRESSEC  
BYTE IS NOT ALTERED BY THE INSTRUCTION.  
*****  
TEST 42 TEST MODE 1 ODD BYTE USING SOP INST  
*****  
TST42: INC (R2) ;UPDATE TEST NUMBER  
CMP #42,(R2) ;SEQUENCE ERROR?  
BNE TST43-10 ;BR TO ERROR HALT ON SEQ EPRCR  
CLR R0 ;INITIALIZE R0  
CLR (R0) ;INITIALIZE LOC. 0  
COM (R0)  
INC R0 ;R0=ODD BYTE  
CLRB (R0) ;TRY TO CLEAR BYTE 1  
BEQ SOPBIC  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 772 <====  
  
MOV #60,-(R2) ;MOVE TO MAILBOX # ***** 60 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLRB DID NOT SET Z-BIT  
SOPBIC: DEC R0 ;RO=WORD ADDR.  
INC (R0) ;INCREMENT TO TEST WORD  
INC R0 ;RO=ODD BYTE  
NEGB (R0) ;TRY TO NEGATE BYTE 1  
BPL SOPB1D  
INCB (R0) ;TRY TO INCREMENT BYTE 1  
BEQ TST43  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====  
  
SOPB1D: MOV #61,-(R2) ;MOVE TO MAILBOX # ***** 61 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.  
; OR SEQUENCE ERROR
```

```

1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543 003012 005212
1544 003014 022712 000043
1545 003020 001023
1546 003022 005000
1547 003024 105100
1548 003026 005200
1549 003030 005010
1550 003032 005110
1551 003034 005020
1552 003036 001404
1553
1554
1555
1556
1557 003040 012742 000062
1558 003044 005242
1559 003046 000000
1560 003050 005300
1561 003052 005300
1562 003054 005120
1563 003056 100004
1564 003060 005300
1565 003062 005300
1566 003064 005220
1567 003066 001404
1568
1569
1570
1571
1572 003070
1573 003070 012742 000063
1574 003074 005242
1575 003076 000000
1576

```

```

*****
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 43 TEST MODE 2 USING SOP INST.
*****
TST43: INC (R2) ;UPDATE TEST NUMBER
CMP #43,(R2) ;SEQUENCE ERROR?
BNE TST44-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400
COMB R0
INC R0
CLR (R0) ;CLEAR 400
COM (R0) ;INITIALIZE: 400=-1
CLR (R0)+ ;TRY CLEARING WITH MODE 2
BEQ SOPZA
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV #62,-(R2) ;MOVE TO MAILBOX # ***** 62 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR INST DID NOT SET Z-BIT
SOPZA: DEC R0 ;RESET R0
DEC R0
COM (R0)+ ;TRY COMPLEMENTING WITH MODE 2
BPL SOP2B
DEC R0 ;RESET R0
DEC R0
INC (R0)+ ;TRY INCREMENTING WITH MODE 2
BEQ TST44
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
SOP2B: MOV #63,-(R2) ;MOVE TO MAILBOX # ***** 63 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST
; OR SEQUENCE ERROR

```

1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623

003100 005212
003102 022712 000044
003106 001022
003110 005000
003112 105100
003114 005200
003116 005010
003120 005110
003122 105020
003124 001404

003126 012742 000064
003132 005242
003134 000000
003136 005300
003140 005210
003142 105420
003144 100003
003146 005300
003150 105220
003152 001404

003154 012742 000065
003154 005242
003162 000000

```
*****
THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
ADDRESS EVEN BYTES.  RO IS SET TO 400 AND USED TO INITIALIZE LOCATION
400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
MODE 2.
RO 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 44      TEST MODE 2 EVEN BYTE USING SOP INST.
*****
TST44:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #44,(R2)     ;SEQUENCE ERROR?
        BNE     TST45-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO           ;SET RO=400
        COMB    RO
        INC     RO
        CLR     (RO)         ;CLEAR 400
        COM     (RO)         ;INITIALIZE: 400=-1
        CLRB   (RO)+        ;TRY TO CLEAT 400 W/MODE 2
        BEQ    SOPB2A
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ;          CONDITIONAL BRANCH INST. AND <====
        ;          REPLACE THE MOVE INSTRUCTION <====
        ;          WHICH FOLLOWS W/ 771 <====
        MOV     #64, -(R2)   ;MOVE TO MAILBOX # ***** 64 *****
        INC     -(R2)
        HALT
SOPB2A: DEC     RO           ;SET MSGTYP TO FATAL ERROR
        INC     (RO)         ;CLR DID NOT SET Z-BIT
        NEGB   (RO)+        ;RESULT RO=400
        BPL    SOPB2B       ;INC 400 TO TEST WORD
        ; TRY TO NEGATE EVEN BYTE
        DEC     RO           ;RESET RO=400
        INCB   (RO)+        ;TRY INCREMENT OF EVEN BYTE
        BEQ    TST45
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ;          CONDITIONAL BRANCH INST. AND <====
        ;          REPLACE THE MOVE INSTRUCTION <====
        ;          WHICH FOLLOWS W/ 756 <====
SOPB2B: MOV     #65, -(R2)   ;MOVE TO MAILBOX # ***** 65 *****
        INC     -(R2)
        HALT
        ; SET MSGTYP TO FATAL ERROR
        ; TEST CUMMULATIVE RESULT OF ABOVE INST.
        ; OR SEQUENCE ERROR
```

TEST MODE 2 ODD BYTE USING SOP INST.

1624
1625
1626
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668

003164 005212
003166 022712 000045
003172 001025
005174 005000
003178 105100
003200 005200
003202 005010
003204 005110
003206 005200
003210 105020
003212 001404

003214 012742 000066
003220 005242
003222 000000
003224 005300
003226 005300
003230 005220
003232 005300
003234 105420
003236 100003
003240 005300
003242 105220
003244 001404

003246 012742 000067
003252 005242
003254 000000

```
*****
THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
TEST.  HERE.  THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
*****
TEST 45      TEST MODE 2 ODD BYTE USING SOP INST.
*****
ST45:  INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #45,(R2)      ;SEQUENCE ERROR?
      BNE     TST46-10      ;BR TO ERROR HALT ON SEQ ERROR.
      CLR     RO            ;SET RO=400
      COMB    RO
      INC     RO
      CLR     (RO)          ;CLEAR LOC 400
      COM     (RO)          ;INITIALIZE: 400=-1
      INC     RO            ;RO=ODD BYTE
      CLRB   (RO)+         ;TRY TO CLEAR ODD BYTE
      BEQ    SOPB2C
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
      ;          CONDITIONAL BRANCH INST. AND
      ;          REPLACE THE MOVE INSTRUCTION
      ;          WHICH FOLLOWS W/ 770
      ;          ***** 66 *****
      MOV     #66,-(R2)     ;MOVE TO MAILBOX # ***** 66 *****
      INC     -(R2)
      HALT
SOPB2C: DEC     RO
      DEC     RO
      INC     (RO)+        ;INCREMENT WORD
      DEC     RO           ;POINT TO ODD BYTE
      NEGB   (RO)+        ;TRY TO NEGATE ODD BYTE
      BPL    SOPB2D
      DEC     RO
      INCB   (RO)+        ;RESET RO TO ODD BYTE
      BEQ    TST46        ;TRY TO INCREMENT ODD BYTE
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
      ;          CONDITIONAL BRANCH INST. AND
      ;          REPLACE THE MOVE INSTRUCTION
      ;          WHICH FOLLOWS W/ 753
      ;          ***** 67 *****
      MOV     #67,-(R2)     ;MOVE TO MAILBOX # ***** 67 *****
      INC     -(R2)
      HALT
      ;TEST CUMMULATIVE RESULT OF ABOVE INST.
      ; OR SEQUENCE ERROR
*****
```

1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717

003256 005212
003260 022712 000046
003264 001020
003266 005000
003270 105100
003272 005200
003274 005010
003276 005030
003300 001404

003302 012742 000070
003306 005242
003310 000000
003312 005300
003314 005300
003316 005130
003320 100002
003322 005230
003324 001404

003326 012742 000071
003326 005242
003332 000000
003334

```
*****
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.
RO IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0. THEN RO
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 46 TEST MODE 3 USING SOP INST.
*****
TST46: INC (R2) ;UPDATE TEST NUMBER
CMP #46,(R2) ;SEQUENCE ERROR?
BNE TST47-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC RO
CLR (RO) ;CLEAR LOC 400
CLR @ (RO)+ ;TRY TO CLEAR LOC 0 USING MODE 3
BEQ SOP3A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 ***** <====
; MOVE TO MAILBOX # ***** 70 *****
; SET MSGTYP TO FATAL ERROR
; CLR DID NOT SET Z-BIT
; RESET RO=400
SOP3A: DEC RO
DEC RO
COM @ (RO)+ ;TRY TO COMPLEMENT LOC 0 OF MODE 3
BPL SOP3B
INC @ (RO)+ ;TRY TO INCREMENT LOC 0 W/MODE 3
BEQ TST47
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 760 ***** <====
SOP3B: MOV #71,-(R2) ;MOVE TO MAILBOX # ***** 71 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED
; OR SEQUENCE ERROR
```

1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770

003336 005212
003340 022712 000047
003344 001026
003346 005004
003350 105104
003352 005204
003354 005000
003356 005010
003360 005110
003362 105034
003364 001404

012742 000072
005242
006000
005304
005304
005234
100006
105434
100004
005304
005304
105234
001404

003422 012742 000073
003422 005242
003430 000000

```
*****
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 47 TEST MODE 3 EVEN BYTE USING SOP INST.
*****
TST47: INC (R2) ;UPDATE TEST NUMBER
CMP #47,(R2) ;SEQUENCE ERROR?
BNE TST50-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R4 ;SET R4=400
COMB R4
INC R4
CLR R0 ;INITIALIZE LOC. 0=-1
CLR (R0)
COM (R0)
CLRB @R4+ ;TRY TO CLEAR EVEN BYTE
BEQ SOPB3A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 <====
MOV #72,-(R2) ;MOVE TO MAILBOX # ***** 72 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLRB DID NOT SET Z-BIT
SOPB3A: DEC R4 ;RESET POINTER
DEC R4
INC @R4+ ;TRY INCREMENTING WORD
BPL SOPB3B ;TRY TO NEGATE EVEN BYTE
NEGB @R4+
BPL SOPB3B
DEC R4
DEC R4
INCB @R4+ ;TRY TO INCREMENT EVEN BYTE
BEQ TST50
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 752 <====
SOPB3B: MOV #73,-(R2) ;MOVE TO MAILBOX # ***** 73 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED
; OR SEQUENCE ERROR
```

1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822

003432 005212
003434 022712 000050
003440 001024
003442 005000
003444 105100
003446 005200
003450 005030
003452 005130
003454 105030
003456 001404

012742 000074
003464 005242
003466 000000
003470 005300
003472 005300
003474 005300
003476 005300
003500 005230
003502 105430
003504 100002
003506 105230
003510 001404

003512 012742 000075
003516 005242
003520 000000

```
*****  
; 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 50 TEST MODE 3 ODD BYTE USING SOP INST.  
*****  
TST50: INC (R2) ; UPDATE TEST NUMBER  
CMP #50,(R2) ; SEQUENCE ERROR?  
BNE TST51-10 ; BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ; SET R0=400  
COMB R0  
INC R0  
CLR @ (R0)+ ; INITIALIZE  
COM @ (R0)+ ; LOC 0=-1 R0=404  
CLRB @ (R0)+ ; TRY TO CLEAR ODD BYTE  
BEQ SOPB3C  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 771  
; MOVE TO MAILBOX # ***** 74 *****  
; SET MSGTYP TO FATAL ERROR  
; CLRB DID NOT SET Z-BIT  
; RESET R0  
  
SOPB3C: DEC R0 ; POINT TO EVEN BYTE ADDR.  
DEC R0  
DEC R0  
DEC R0  
INC @ (R0)+ ; INCREMENT WORD  
NEGB @ (R0)+ ; TRY TO NEGATE ODD BYTE  
BPL SOPB3D  
INCB @ (R0)+ ; TRY TO INCREMENT ODD BYTE  
BEQ TST51  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 754  
  
SOPB3D: MOV #75,-(R2) ; MOVE TO MAILBOX # ***** 75 *****  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; CUMMULATIVE RESULT OF ABOVE INSTS FAILED  
; OR SEQUENCE ERROR
```

1823
1824
1825
1826
1827 003522 005212
1828 003524 022712 000051
1829 003530 001021
1830 003532 005000
1831 003534 105100
1832 003536 005200
1833 003540 005040
1834 003542 001404
1835
1836
1837
1838
1839 003544 012742 000076
1840 003550 005242
1841 003552 000000
1842 003554 005200
1843 003556 005200
1844 003560 005140
1845 003562 100004
1846 003564 005200
1847 003566 005200
1848 003570 005240
1849 003572 001404
1850
1851
1852
1853
1854 003574
1855 003574 012742 000077
1856 003600 005242
1857 003602 000000
1858

```
*****  
;TEST 51 TEST MODE 4 USING SOP INSTS  
*****  
TST51: INC (R2) ;UPDATE TEST NUMBER  
CMP #51,(R2) ;SEQUENCE ERROR?  
BNE TST52-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;SET RO=400  
COMB RO  
INC RO  
CLR -(RO) ;TRY TO CLEAR USING MODE 4  
BEQ SOP4A  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 773 <====  
MOV #76,-(R2) ;MOVE TO MAILBOX # ***** 76 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOP4A: INC RO ;RESET RO  
INC RO  
COM -(RO) ;TRY TO COMPLEMENT USING MODE 4  
BPL SOP4B  
INC RO ;MOVE POINTER  
INC -(RO)  
BEQ TST52  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====  
SOP4B: MOV #77,-(R2) ;MOVE TO MAILBOX # ***** 77 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST.  
; OR SEQUENCE ERROR
```

1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907

003604 005212
003606 022712 000052
003612 001017
003614 005000
003616 005020
003620 105400
003622 005050
003624 001404

003626 012742 000100
003632 005242
003634 000000
003636 005200
003640 005200
003642 005150
003644 100002
003646 005250
003650 001404

003652
003652 012742 000101
003656 005242
003660 000000

```
*****
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
RO 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 RO 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 52 TEST MODE 5 USING SOP INSTS
*****
TST52: INC (R2) ;UPDATE TEST NUMBER
CMP #52,(R2) ;SEQUENCE ERROR?
BNE TST53-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=376
CLR (RO)+
NEGB RO
CLR 2-(RO) ;TRY TO CLEAR LOC 0 W/MODE 5
BEQ SOP5A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
MOV #100,-(R2) ;MOVE TO MAILBOX # ***** 100 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP5A: INC RO ;RESET RO
INC RO
COM 2-(RO) ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
BPL SOP5B
INC 2-(RO) ;TRY TO INCREMENT LOC. 0 W/MODE 5
BEQ TST53
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
SOP5B: MOV #101,-(R2) ;MOVE TO MAILBOX # ***** 101 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
; OR SEQUENCE ERROR
```


1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988

003742 005212
003744 022712 000054
003750 001020
003752 005000
003754 105100
003756 005200
003760 005070 000002
003764 001404

012742 000104
005242
000000
005170 000002
100003
005270 000002
001404

012742 000105
005242
000000

THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS. IT USES
THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
RO 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 54 TEST MODE 7 USING SOP INST.

ST54: INC (R2) ;UPDATE TEST NUMBER
CMP #54,(R2) ;SEQUENCE ERROR?
BNE TST55-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC RO
CLR 02(RO) ;TRY TO CLEAR LOC. 0 W/MODE 7
BEQ SOP7A

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
MOV #104,-(R2) ;MOVE TO MAILBOX # ***** 104 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP7A: COM 02(RO) ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
BPL SOP7B
INC 02(RO) ;TRY TO INCREMENT LOC. 0 W/MODE 7
BEQ TST55

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 750 <====
SOP7B: MOV #105,-(R2) ;MOVE TO MAILBOX # ***** 105 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
; OR SEQUENCE ERROR

1989
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000
 2001 004022 005212
 2002 004024 022712 000055
 2003 004030 001017
 2004 004032 005027
 2005 004034 177777
 2006 004036 001404
 2007
 2008
 2009
 2010
 2011 004040 012742 000106
 2012 004044 005242
 2013 004046 000000
 2014 004050 005237 004034
 2015 004054 005467 177754
 2016 004060 100003
 2017 004062 005277 000012
 2018 004066 001405
 2019
 2020
 2021
 2022
 2023 004070
 2024 004070 012742 000107
 2025 004074 005242
 2026 004076 000000
 2027
 2028 004100 004034

```

*****
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 55 TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
*****
TST55: INC (R2) ;UPDATE TEST NUMBER
        CMP #55,(R2) ;SEQUENCE ERROR?
        BNE SOPB ;BR TO ERROR HALT ON SEQ ERROR
        CLR (R7)+ ;CLEAR NEXT LOCATION: (SOPX)
SOPX: -1 ;USE MODE 27
        BEQ SOPA

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 775 <====
        MOV #106,-(R2) ;MOVE TO MAILBOX # ***** 106 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CLR DID NOT SET Z-BIT
SOPA: INC #SOPX ;INC SOPX W/MODE 37
        NEG SOPX ;NEGATE SOPX W/MODE 67
        BPL SOPB
        INC #SOPXAD ;INC SOPX W/MODE 77
        BEQ TST56

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
SOPB: MOV #107,-(R2) ;MOVE TO MAILBOX # ***** 107 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;INC DID NOT SET Z-BIT
; OR SEQUENCE ERROR
SOPXAD: SOPX ;INDIRECT ADDRESS OF SOPX
    
```

2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060

004102 005212
004104 022712 000056
004110 001010
004112 005000
004114 000277
004116 000244
004120 005700
004122 102403
004124 100402
004126 103401
004130 001404

004132
004132 012742 000110
004136 005242
004140 000000

THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
USING MODE 0. R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION. A TST INSTRUCTION
IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION
CODES.

TEST 56 TEST MODE 0 SOP NON-MODIFYING

ST56: INC (R2) ;UPDATE TEST NUMBER
CMP #56,(R2) ;SEQUENCE ERROR?
BNE TST57-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;INITIALIZE R0=0
SCC ;SET CC=1011
CLZ
TST R0 ;TRY TST W/ MODE 0
BVS SNM0A ;CHECK THAT CC=0100
BMT SNM0A
BCS SNM0A
BEQ TST57

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

SNM0A: MOV #110,-(R2) ;MOVE TO MAILBOX # ***** 110 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CONDITION CODES NOT SET PROPERLY
; OR SEQUENCE ERROR

TEST MODE 0 SOP NON-MODIFYING

2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092

004142 005212
004144 022712 000057
004150 001010
004152 005000
004154 105100
004156 000277
004160 000250
004152 105700
004164 102402
004166 101401
004170 100404

004172 012742 000111
004172 005242
004200 000000

THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0
RO IS SET TO 377 AND COMPLEMENT OF THE EXPECTED CONDITION CODES
IS LOADED IN PSW. A TSTB INSTRUCTION IS EXECUTED AND THE RESULTS
ARE CHECKED WITH SEVERAL CONDITIONAL BRANCH INSTRUCTIONS.
THIS VERIFIES THAT THE PROPER BYTE WAS TESTED.

TEST 57 TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING

TST57: INC (R2) ; UPDATE TEST NUMBER
CMP #57, (R2) ; SEQUENCE ERROR?
BNE TST60-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR RC ; INITIALIZE
COMB RO ; RO=377
SCC ; SET CC=0111
CLN
TSTB RO ; TRY TST EVEN BYTE
BVS SNMBOA ; CHECK CC=1000
BLOS SNMBOA
BMI TST60

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

SNMBOA: MOV #111, -(R2) ; MOVE TO MAILBOX # ***** 111 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CONDITION CODES NOT SET PROPERLY
; OR SEQUENCE ERROR

2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125

004202 005212
004204 022712 000060
004210 001011
004212 005000
004214 005010
004216 000277
004220 000244
004222 005710
004224 102403
004226 103402
004230 100401
004232 001404

004234
004234 012742 000112
004240 005242
004242 000000

THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
RO 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 RO AND CONDITIONAL BRANCHES TEST
THE RESULTS.

TEST 60 TEST MODE 1 SOP NON-MODIFYING

TST60: INC (R2) ; UPDATE TEST NUMBER
CMP #60, (R2) ; SEQUENCE ERROR?
BNE TST61-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR RO ; POINT TO LOC 0
CLR (RO) ; CLEAR LOC 0
SCC ; INITIALIZE
CLZ ; CC=1011
TST (RO) ; TRY TST W/ MODE 1
BVS SNM1A ; CHECK CC=0100
BCS SNM1A
BMI SNM1A
BEQ TST61

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====

SNM1A: MOV #112, -(R2) ; MOVE TO MAILBOX # ***** 112 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CC'S NOT SET PROPERLY
; OR SEQUENCE ERROR

TEST CODE 1 SOP NON-MODIFYING

2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174

004244 005212
004246 022712 000061
004252 001026
004254 005000
004256 005010
004260 105110
004262 000277
004264 000250
004266 105710
004270 102402
004272 101401
004274 100404

004276 012742 000113
004302 005242
004304 000030
004306 005000
004310 005200
004312 000277
004314 000244
004316 105710
004320 102403
004322 103402
004324 100401
004326 001404

004330 012742 000114
004334 005242
004336 000000

```
*****
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 61 TEST MODE 1 BYTE INST. NON-MODIFYING
*****
TST61: INC (R2) ;UPDATE TEST NUMBER
CMP #61,(R2) ;SEQUENCE ERROR?
BNE TST62-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;POINT TO LOC 0
CLR (R0) ;CLEAR LOC 0
COMB (R0) ;COMPLEMENT BYTE 0
SCC ;SET CC=0111
CLN
TSTB (R0) ;TRY TST ON EVEN BYTE
BVS SNMB1A
BLOS SNMB1A
BMI SNMB1B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====

SNMB1A: MOV #113,-(R2) ;MOVE TO MAILBOX # ***** 113 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT

SNMB1B: CLR R0
INC R0
SCC ;SET CC=1011
CLZ
TSTB (R0) ;TRY TO TST AN ODD BYTE
BVS SNMB1C ;CHECK CC=0100
BCS SNMB1C
BMI SNMB1C
BEQ TST62

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 752 <====

SNMB1C: MOV #114,-(R2) ;MOVE TO MAILBOX # ***** 114 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
; OR SEQUENCE ERROR
```

2175
2176
2177
2178
2179
2180
2181
2182
2183
2194
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216

004340 005212
004342 022712 000062
004346 001020
004350 005000
004352 005010
004354 000277
004356 000244
004360 005720
004362 102403
004364 103402
004366 100401
004370 001404

004372
004372 012742 000115
004376 005242
004400 000000
004402 005300
004404 005300
004406 001404

004410 012742 000116
004414 005242
004416 000000

```
*****
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 62 TEST MODE 2 WITH SOP NON-MODIFYING
*****
TST62: INC (R2) ;UPDATE TEST NUMBER
CMP #62,(R2) ;SEQUENCE ERROR?
BNE TST63-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;INITIALIZE RO=0
CLR (RO) ;CLEAR LOC 0
SCC ;SET CC=1011
CLZ
TST (RO)+ ;TRY TST W/ MODE 2
BVS SNM2A ;CHECK CC=0100
BCS SNM2A
BMI SNM2A
BEQ SNM2B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====

SNM2A: MOV #115,-(R2) ;MOVE TO MAILBOX # ***** 115 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
SNM2B: DEC RO ;RESET RO
DEC RO
BEQ TST63

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 760 <====

MOV #116,-(R2) ;MOVE TO MAILBOX # ***** 116 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MODE 2 DID NOT INC REG CORRECTLY
; OR SEQUENCE ERROR
```

2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272

004420 005212
004422 022712 000063
004426 001042
004430 005000
004432 005010
004434 105110
004436 000277
004440 000250
004442 105720
004444 102402
004446 101401
004450 100404

004452 012742 000117
004456 005242
004460 000000
004462 005300
004464 001404

004466 012742 000120
004472 005242
004474 000000
004476 005200
004500 000277
004502 000244
004504 105720
004506 102403
004510 103402
004512 100401
004514 001404

004516 012742 000121
004522 005242

```
*****
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 63 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
*****
TST63: INC (R2) ;UPDATE TEST NUMBER
CMP #63,(R2) ;SEQUENCE ERROR?
BNE TST64-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;CLEAR R0
CLR (R0) ;CLEAR LOC 0
COMB (R0) ;SET LOC 0=377
SCC ;SET CC=0111
CLN
TSTB (R0)+ ;TRY TST OF EVEN BYTE
BVS SNMB2A
BLOS SNMB2A
BMI SNMB2B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
SNMB2A: MOV #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT SET CORRECTLY
SNMB2B: DEC R0 ;DECREMENT R0
BEQ SNMB2C
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
SNMB2C: MOV #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MODE 2 DID NOT INC REG CORRECTLY
SNMB2D: INC R0 ;POINT TO ODD BYTE
SCC ;SET CC=1011
CLZ
TSTB (R0)+ ;TRY TST OF ODD BYTE
BVS SNMB2D ;CHECK CC'S=0100
BCS SNMB2D
BMI SNMB2D
BEQ SNMB2E
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITI. AL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 745 <====
SNMB2D: MOV #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
```

2273 004524 000000
2274 004526 005300
2275 004530 005300
2276 004532 001404
2277
2278
2279
2280
2281 004534 012742 000122
2282 004540 005242
2283 004542 000000
2284

SNMBZE: HALT
DEC RO
DEC RO
BEG TST64

:CC'S NOT CORRECT

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
: CONDITIONAL BRANCH INST. AND
: REPLACE THE MOVE INSTRUCTION
: WHICH FOLLOWS W/ 736
: MOVE TO MAILBOX # ***** 122 *****
: SET MSGTYP TO FATAL ERROR
: RO DID NOT INCREMENT PROPERLY
: OR SEQUENCE ERROR

2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328

004544 005212
004546 022712 000064
004552 001022
004554 005000
004556 005010
004560 105100
004562 005300
004564 000277
004566 000244
004570 005730
004572 102403
004574 103402
004576 100401
004600 001404

004602 012742 000123
004606 005242
004610 000000
004612 005300
004614 105100
004616 001404

004620 012742 000124
004624 005242
004626 000000

```
*****
: THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
: A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
: THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
: TST MODE 3 INSTRUCTION.
*****
: TEST 64 TEST MODE 3 W/ SOP NON-MODIFYING INSTS
*****
TST64: INC (R2) ; UPDATE TEST NUMBER
CMP #64,(R2) ; SEQUENCE ERROR?
BNE TST65-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR R0 ; R0=0
CLR (R0) ; CLEAR LOC 0
COMB R0 ; R0=376
DEC R0
SCC ; SET CC=1011
CLZ
TST 2(R0)+ ; TRY TST W/ MODE 3
BVS SNM3A ; CHECK CC=0100
BCS SNM3A
BMI SNM3A
BEQ SNM3B

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

SNM3A: MOV #123, -(R2) ; MOVE TO MAILBOX # ***** 123 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CC'S NOT CORRECT

SNM3B: DEC R0 ; R0=377
COMB R0 ; R0=0
BEQ TST65

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

MOV #124, -(R2) ; MOVE TO MAILBOX # ***** 124 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; MODE 3 DID NOT INC REG CORRECTLY
; OR SEQUENCE ERROR
```

2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384

004630 005212
004632 022712 000065
004636 001036
004640 005000
004642 005010
004644 105110
004646 105100
004650 005200
004652 005720
004654 000277
004656 000250
004660 105730
004662 102402
004664 101401
004666 100404

004670
004670 012742 000125
004674 005242
004676 000000
004700 000277
004702 000244
004704 105730
004706 102403
004710 103402
004712 100401
004714 001404

004716
004716 012742 000126
004722 005242
004724 000000
004726 005720
004730 005710
004732 100404

```
*****
: 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 65 TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
*****
TST65: INC (R2) ; UPDATE TEST NUMBER
      CMP #65,(R2) ; SEQUENCE ERROR?
      BNE TST66-10 ; BR TO ERROR HALT ON SEQ ERROR
      CLR RO ; RO=0
      CLR (RO) ; CLEAR LOC 0
      COMB (RO) ; LOC. 0 =377
      COMB RO
      INC RO
      TST (RO)+ ; RO=402
      SCC ; CC=0111
      CLN
      TSTB @ (RO)+ ; TRY TST OF EVEN BYTE
      BVS SNMB3A ; CHECK CC=1000
      BLOS SNMB3A
      BMI SNMB3B

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

SNMB3A: MOV #125,-(R2) ; MOVE TO MAILBOX # ***** 125 *****
      INC -(R2) ; SET MSGTYP TO FATAL ERROR
      HALT ; CC'S NOT CORRECT
SNMB3B: SCC ; SET CC=1011
      CLZ
      TSTB @ (RO)+ ; TRY TST OF ODD BYTE
      BVS SNMB3C ; CHECK CC=0100
      BCS SNMB3C
      BMI SNMB3C
      BEQ SNMB3D

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

SNMB3C: MOV #126,-(R2) ; MOVE TO MAILBOX # ***** 126 *****
      INC -(R2) ; SET MSGTYP TO FATAL ERROR
      HALT ; CC'S NOT CORRECT
SNMB3D: TST (RO)+ ; RO=410
      TST (RO)
      BMI TST66

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


2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475

005022 005212
005024 022712 000067
005030 001022
005032 005000
005034 005010
005036 005110
005040 105100
005042 005200
005044 000277
005046 000250
005050 005750
005052 102402
005054 101401
005056 100404

005060
005060 012742 000132
005064 005242
005066 000000
005070 005200
005072 105100
005074 001404

005076 012742 000133
005102 005242
005104 000000

```
*****
THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
IT USES A POINTER AT LOC. 376 TO TEST LOC. 0. RO IS SET
TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
RO IS CHECKED TO INSURE PROPER DECREMENTING.
*****
TEST 67 TEST MODE 5 W/ SOP NON-MODIFYING INSTS
*****
TST67: INC (R2) ;UPDATE TEST NUMBER
CMP #67,(R2) ;SEQUENCE ERROR?
BNE TST70-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;RO=0
CLR (RO) ;LOC 0=0
COM (RO) ;LOC 0=-1
COMB RO ;RO=377
INC RO ;RO=400
SCC ;SET CC=0111
CLN
TST @-(RO) ;TRY TST W/ MODE 5
BVS SNM5A ;CHECK CC=1000
BLOS SNM5A
BMI SNM5B

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

SNM5A: MOV #132,-(R2) ;MOVE TO MAILBOX # ***** 132 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT SET PROPERLY
SNM5B: INC RO ;RO=377
COMB RO ;RO=0
BEQ TST70

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

MOV #133,-(R2) ;MOVE TO MAILBOX # ***** 133 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MODE 5 DID NOT DEC RO CORRECTLY
; OR SEQUENCE ERROR
```


2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559

005170 005212
005172 022712 000071
005176 001021
005200 005000
005202 005010
005204 005110
005206 105100
005210 000277
005212 000250
005214 005770 000001
005220 102402
005222 101401
005224 100404

005226
005226 012742 000136
005232 005242
005234 000000
005236 105100
005240 001404

005242 012742 000137
005246 005242
005250 000000

```
*****
THIS TEST VERIFIES MODE 7 SOP NON-MODIFYING INSTRUCTIONS.
IT USES A POINTER TO LOC. 0 STORED AT LOC. 400 TO TST LOC. 0.
RO IS SET TO 377 AND LOC. 0 IS TESTED THRU THE POINTER AT 400 USING
RO AND AN OFFSET OF 1.
*****
TEST 71      TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
*****
TST71:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #71,(R2)     ;SEQUENCE ERROR?
        BNE     TST72-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO           ;RO=0
        CLR     (RO)         ;LOC 0=0
        COM     (RO)         ;LOC 0=-1
        COMB    RO           ;RO=377
        SCC     ;            ;CC=0111
        CLN     ;
        TST     21(R0)        ;TRY TST W/ MODE 7
        BVS     SNM7A         ;CHECK CC=1000
        BLOS    SNM7A
        BMI     SNM7B

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

SNM7A:  MOV      #136,-(R2)    ;MOVE TO MAILBOX # ***** 136 *****
        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
        HALT    ;            ;CC'S NOT CORRECT
        COMB    RO           ;RO=0
        BEG     TST72

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 757 <====

SNM7B:  MOV      #137,-(R2)    ;MOVE TO MAILBOX # ***** 137 *****
        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
        HALT    ;            ;TST MODE 7 INCORRECTLY CHANGED RC
;           OR SEQUENCE ERROR
```

2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613

005252 005212
005254 022712 000072
005260 001006
005262 005000
005264 005100
005266 005004
005270 060004
005272 005204
005274 001404

005276 012742 000140
005302 005242
005304 000000

THIS TEST VERIFIES MODE 0 DOUBLE OPERAND INSTRUCTIONS. IT SETS
DATA IN R0 AND R4 AND USES THE ADD INSTRUCTION TO TEST THE DCP
MICROCODE.

TEST 72 TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.

TST72: INC (R2) ; UPDATE TEST NUMBER
CMP #72,(R2) ; SEQUENCE ERROR?
BNE TST73-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR R0 ; R0=0
COM R0 ; R0=-1
CLR R4 ; R4=0
ADD R0,R4 ; TRY ADD: R4=-1
INC R4 ; R4=0
BEQ TST73

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
; MOVE TO MAILBOX # ***** 140 *****
; SET MSGTYP TO FATAL ERROR
; 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 73 MOV MODE 0 TO MODE 0

TST73: INC (R2) ; UPDATE TEST NUMBER
CMP #73,(R2) ; SEQUENCE ERROR?
BNE TST74-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR R0 ; R0=0
CLR R4 ; R4=0
COM R0 ; R0=-1
MOV R0,R4 ; TRY MOVE -1 TO R4
INC R4 ; INC R4
BEQ TST74

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
; MOVE TO MAILBOX # ***** 141 *****
; SET MSGTYP TO FATAL ERROR
; MOVE FAILED MODE 0 TO MODE 0
; OR SEQUENCE ERROR

2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626 005342 005212
2627 005344 022712 00007-
2628 005350 001051
2629 005352 005000
2630 005354 010004
2631 005356 001404
2632
2633
2634
2635
2636 005360 012742 000142
2637 005364 005242
2638 005366 000000
2639 005370 005200
2640 005372 005100
2641 005374 005104
2642 005376 040004
2643 005400 005304
2644 005402 001404
2645
2646
2647
2648
2649 005404 012742 000143
2650 005410 005242
2651 005412 000000
2652 005414 050004
2653 005416 005204
2654 005420 005204
2655 005422 001404
2656
2657
2658
2659
2660 005424 012742 000144
2661 005430 005242
2662 005432 000000
2663 005434 005000
2664 005436 105100
2665 005440 005004
2666 005442 005104
2667 005444 040004
2668 005446 060004
2669 005450 005204

```

*****
THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
WITH MODE 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 74          TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0
*****
ST74:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #74,(R2)     ;SEQUENCE ERROR?
        BNE     TST75-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0           ;R0=0
        MOV     R0,R4        ;TRY MOVE MODE 0,0
        BEQ     DOP0A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
;           CONDITIONAL BRANCH INST. AND
;           REPLACE THE MOVE INSTRUCTION
;           WHICH FOLLOWS W/ 775
;           <====
;           <====
;           <====
;           <====
; MOVE TO MAILBOX # ***** 142 *****
; SET MSGTYP TO FATAL ERROR
; Z-BIT NOT SET
; R0=1
; R0=177776
; R4=177777
; TRY BIC: R4=1
; R4=0
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
;           CONDITIONAL BRANCH INST. AND
;           REPLACE THE MOVE INSTRUCTION
;           WHICH FOLLOWS W/ 763
;           <====
;           <====
;           <====
;           <====
; MOVE TO MAILBOX # ***** 143 *****
; SET MSGTYP TO FATAL ERROR
; BIC CLEAR RESULT INCORRECT
; TRY BIS: R4=177777
; R4=0
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
;           CONDITIONAL BRANCH INST. AND
;           REPLACE THE MOVE INSTRUCTION
;           WHICH FOLLOWS W/ 753
;           <====
;           <====
;           <====
;           <====
; MOVE TO MAILBOX # ***** 144 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF BIS INCORRECT
; R0=0
; R0=377
; R4=0
; R4=177777
; R4=177400
; TRY ADD: R4=177777
; R4=0
DOP0A:  INC      R0
        COM     R0
        COM     R4
        BIC     R0,R4
        DEC     R4
        BEQ     DOP0B
DOP0B:  BIS      R0,R4
        INC     R4
        INC     R4
        BEQ     DOP0C
DOP0C:  MOV      #144,-(R2)
        INC     -(R2)
        HALT
        CLR     R0
        COMB   R0
        CLR     R4
        COM     R4
        BIC     R0,R4
        ADD     R0,R4
        INC     R4

```

2670	005452	001404		BEQ	DOP00				
2671									
2672									
2673									
2674									
2675	005454	012742	000145	MOV	#145, -(R2)				
2676	005460	005242		INC	-(R2)				
2677	005462	000000		HALT					
2678	005464	160004							
2679	005466	105404							
2680	005470	005204							
2681	005472	001404		BEQ	TST75				
2682									
2683									
2684									
2685									
2686	005474	012742	000146	MOV	#146, -(R2)				
2687	005500	005242		INC	-(R2)				
2688	005502	000000		HALT					
2689									

DOP00:

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 737
; MOVE TO MAILBOX # ***** 145 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF ADD INCORRECT
; 177401=R4
; R4=177777
; RD=0

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 727
; MOVE TO MAILBOX # ***** 146 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF SUB INCORRECT
; OR SEQUENCE ERROR

```

2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700 005504 005212
2701 005506 022712 000075
2702 005512 001042
2703 005514 005000
2704 005516 005004
2705 005520 005204
2706 005522 020400
2707 005524 003004
2708
2709
2710
2711
2712 005526 012742 000147
2713 005532 005242
2714 005534 000000
2715 005536 020004
2716 005540 002404
2717
2718
2719
2720
2721 005542 012742 000150
2722 005546 005242
2723 005550 000000
2724 005552 005200
2725 005554 020400
2726 005556 001404
2727
2728
2729
2730
2731 005560 012742 000151
2732 005564 005242
2733 005566 000000
2734 005570 005000
2735 005572 005100
2736 005574 005004
2737 005576 030004
2738 005600 001404
2739
2740
2741
2742
2743 005602 012742 000152
2744 005606 005242
2745 005610 000000

```

*****
THIS TEST VERIFIES MODE 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 75          TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0
*****
TST75:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #75,(R2)     ;SEQUENCE ERROR?
        BNE     TST76-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0           ;R0=0
        CLR     R4           ;R4=0
        INC     R4           ;R4=1
        CMP     R4,R0       ;TRY COMPARE R4 TO R0
        BGT     DNM1
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 773 <====
        MOV     #147,-(R2)    ;MOVE TO MAILBOX # ***** 147 *****
        INC     -(R2)
        HALT
DNM1:   CMP     R0,R4       ;CC'S NOT CORRECT FOR CMP
        BLT     DNM2       ;TRY COMPARE R0 TO R4
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 765 <====
        MOV     #150,-(R2)    ;MOVE TO MAILBOX # ***** 150 *****
        INC     -(R2)
        HALT
DNM2:   INC     R0
        CMP     R4,R0       ;CC'S NOT CORRECT FOR CMP
        BEQ     DNM3       ;R4=0
        ;TRY COMPARE R4=1 TO R0=1
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 756 <====
        MOV     #151,-(R2)    ;MOVE TO MAILBOX # ***** 151 *****
        INC     -(R2)
        HALT
DNM3:   CLR     R0
        COM     R0
        CLR     R4
        BIT     R0,R4       ;CC'S NOT CORRECT (Z=1) FOR CMP
        BEQ     DNM4       ;R0=0
        ;R0=177777
        ;R4=0
        ;TRY BIT R0 TO R4
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 745 <====
        MOV     #152,-(R2)    ;MOVE TO MAILBOX # ***** 152 *****
        INC     -(R2)
        HALT
;CC'S NOT CORRECT FOR BIT

```


2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2910
2811
2812
2813

005666 005212
005670 022712 000077
005674 001007
005676 005000
005700 005010
005702 005110
005704 005004
005706 151004
005710 105104
005712 001404

005714 012742 000155
005720 005242
005722 000000

```
*****
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 77 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
*****
TST77: INC (R2) ;UPDATE TEST NUMBER
CMP #77,(R2) ;SEQUENCE ERROR?
BNE TST100-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
CLR R4 ;R4=0
BISB (R0),R4 ;TRY MODE 1- EVEN BYTE W/ DOP
CCMB R4 ;R4=0
BEQ TST100
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
; MOVE TO MAILBOX # ***** 155 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF BISB IS INCORRECT
; OR SEQUENCE ERROR
```

```

2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825 005724 005212
2826 005726 022712 000100
2827 005732 001007
2828 005734 005000
2829 005736 005010
2830 005740 005110
2831 005742 005004
2832 005744 105104
2833 005746 121004
2834 005750 001404
2835
2836
2837
2838
2839 005752 012742 000156
2840 005756 005242
2841 005760 000000
2842

```

```

*****
: 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 100 TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
*****
TST100: INC (R2) ;UPDATE TEST NUMBER
CMP #100,(R2) ;SEQUENCE ERROR?
BNE TST101-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC 0=0
COM (R0) ;LOC 0=177777
CLR R4 ;R4=0
COMB R4 ;R4=377
CMPB (R0),R4 ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
BEQ TST101
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
; CONDITIONAL BRANCH INST. AND <===
; REPLACE THE MOVE INSTRUCTION <===
; WHICH FOLLOWS W/ 771 <===
; MOVE TO MAILBOX # ***** 156 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF CMPB INCORRECT
; OR SEQUENCE ERROR

```

2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887

005762 005212
005764 022712 000101
005770 001020
005772 005000
005774 005010
005776 105110
006000 005110
006002 005004
006004 005104
006006 111004
006010 005704
006012 001404

006014 012742 000157
006020 005242
006022 000000
006024 005110
006026 111004
006030 100404

006032 012742 000160
006036 005242
006040 000000

```
*****
THIS TEST VERIFIES MODE 1,0 MOV B INSTRUCTIONS
WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
R4 IS SET TO -1. MOV B 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 101 TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
*****
TST101: INC (R2) ;UPDATE TEST NUMBER
CMP #101,(R2) ;SEQUENCE ERROR?
BNE TST102-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC 0=0
COMB (R0) ;LOC 0=177400
COM (R0)
CLR R4 ;R4=0
COM R4 ;R4=177777
MOVB (R0),R4 ;R4=0
TST R4 ;CHECK SIGN OF WORD
BEQ DOP1

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
MOV #157,-(R2) ;MOVE TO MAILBOX # ***** 157 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOVB SHOULD SIGN X-TEND
DOP1: COM (R0) ;LOC 0=177777
MOVB (R0),R4 ;DO MOV B W/ EVEN BYTE
BMI TST102

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 760 <====
MOV #160,-(R2) ;MOVE TO MAILBOX # ***** 160 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOVB SHOULD SIGN X-TEND
; OR SEQUENCE ERROR
```

```

2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899 006042 005212
2900 006044 022712 000102
2901 006050 001010
2902 006052 005000
2903 006054 005010
2904 006056 005004
2905 006060 005204
2906 006062 105114
2907 006064 151410
2908 006066 005210
2909 006070 001404
2910
2911
2912
2913
2914 006072 012742 000161
2915 006076 005242
2916 006100 000000
2917

```

```

:*****
:
:      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 102      TEST MODE 1-ODD BYTE W/ DOP INSTS.
:*****
TST102: INC      (R2)      ;UPDATE TEST NUMBER
        CMP      #102,(R2) ;SEQUENCE ERROR?
        BNE     TST103-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0      ;R0=0
        CLR     (R0)    ;LOC. 0=0
        CLR     R4      ;R4=0
        INC     R4      ;R4=1
        COMB    (R4)    ;LOC. 0=177400
        BISB   (R4),(R0);TRY TO BIS LOW ORDER BITS W/ MODE 1
        INC     (R0)    ;CHECK RESULT
        BEQ     TST103
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
:           CONDITIONAL BRANCH INST. AND <====
:           REPLACE THE MOVE INSTRUCTION <====
:           WHICH FOLLOWS W' 770 <====
: MOVE TO MAILBOX # ***** 161 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF BISB INCORRECT
: OR SEQUENCE ERROR

```

2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955

006102 005212
006104 022712 000103
006110 001015
006112 005000
006114 005010
006116 005110
006120 012004
006122 005204
006124 001404

006126 012742 000162
006132 005242
006134 000000
006136 005300
006140 005300
006142 001404

006144 012742 000163
006150 005242
006152 000000

TST103: INC (R2)
CMP #103,(R2)
BNE TST104-10
CLR RO
CLR (RO)
COM (RO)
MOV (RO)+,R4
INC R4
BEQ DOP2

DOP2: DEC RO
DEC RO
BEQ TST104

MOV #162,-(R2)
INC -(R2)
HALT

MOV #163,-(R2)
INC -(R2)
HALT

: THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS. LOC. 0 IS SET TO -1.
: RO 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 103 TEST MODE 2 W/ DOP INSTS.

: UPDATE TEST NUMBER
: SEQUENCE ERROR?
: BR TO ERROR HALT ON SEQ ERROR
: RO=0
: LOC. 0=0
: LOC. 0=177777
: TRY MOVE MODE 2,0
: CHECK R4

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 772 <====
: MOVE TO MAILBOX # ***** 162 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF MOV INST INCORRECT
: TEST RO AFTER MODE 2

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 763 <====
: MOVE TO MAILBOX # ***** 163 *****
: SET MSGTYP TO FATAL ERROR
: REGISTER NOT INCREMENTED IN MODE 2
: OR SEQUENCE ERROR

2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994

006154 005212
006156 022712 000104
006162 001016
006164 005000
006166 010010
006170 005110
006172 142010
006174 105737 000001
006200 001404

006202 012742 000164
006206 005242
006210 000000
006212 105137 000000
006216 001404

006220 012742 000165
006224 005242
006226 000000

```
*****
; 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 104 TEST MODE 2 - EVEN BYTE W/ DOP INST.
*****
TST104: INC (R2) ; UPDATE TEST NUMBER
; CMP #104,(R2) ; SEQUENCE ERROR?
; BNE TST105-10 ; BR TO ERROR HALT ON SEQ ERROR
; CLR R0 ; R0=0
; MOV R0,(R0) ; LOC. 0=0
; COM (R0) ; LOC. 0=177777
; BICB (R0)+,(R0) ; TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
; TSTB @#1 ; CHECK RESULT
; BEQ DOPB2A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
; MOV #164,-(R2) ; MOVE TO MAILBOX # ***** 164 *****
; INC -(R2) ; SET MSGTYP TO FATAL ERROR
; HALT ; BICB DESTINATION INCORRECT
DOPB2A: COMB @#0 ; CHECK BICB SOURCE
; BEQ TST105
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
; MOV #165,-(R2) ; MOVE TO MAILBOX # ***** 165 *****
; INC -(R2) ; SET MSGTYP TO FATAL ERROR
; HALT ; BICB SOURCE INCORRECTLY CHANGED
; OR SEQUENCE ERROR
```


3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144

006412 005212
006414 022712 000110
006420 001011
006422 012737 052652 000000
006430 005000
006432 153700 000001
006436 022700 000125
006442 001404

006444 012742 000172
006450 005242
006452 000000

005212
022712 000111
001015
012700 006536
014037 006536
064037 006536
144037 006536
154037 006537
024037 006536
001411

```
*****
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: RO = 125.
*****
TEST 110 TEST MODE 3 - ODD BYTE W/ DOP INSTS.
*****
TST110: INC (R2) ; UPDATE TEST NUMBER
CMP #110, (R2) ; SEQUENCE ERROR?
BNE TST111-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #52652, @#0 ; MOVE 1'S AND 0'S PATTERN TO LOC 0
CLR RO ; RO=0
BISB @#1, RO ; TRY RO=152 W/ MODE 3 - ODD BYTE
CMP #125, RO ; RO=125?
BEQ TST111
; TO SCOPE. CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 767 (====
MOV #172, -(R2) ; MOVE TO MAILBOX # ***** 172 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; BISB W/ MODE 3 - ODD BYTE FAILED
; OR SEQUENCE ERROR
*****
```

```
*****
THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
TABLE OF OPERANDS. THE TABLE OF OPERANDS AND THE WORK LOCATION IS
STORED FOLLOWING THE TEST CODE. A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
VALUE. THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
GO UNDETECTED. WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
ODD ADDRESSES ARE USED IN THE TEST. THE LISTING SHOWS THE
EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
*****
TEST 111 TEST MODE 4 W/ DOP INSTS.
*****
TST111: INC (R2) ; UPDATE TEST NUMBER
CMP #111, (R2) ; SEQUENCE ERROR?
BNE DOP4 ; BR TO ERROR HALT ON SEQ ERROR
MOV #TBL1, RO ; INITIALIZE RO
MOV -(RO), @#TBL1 ; TBL1=125252
ADD -(RO), @#TBL1 ; TBL1=000377
BICB -(RO), @#TBL1 ; TBL1=000252
BISB -(RO), @#TBL1+1 ; TBL1=125252
CMP -(RO), @#TBL1 ; CHECK RESULT
BEQ TST112
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
*****
```

```

3145 006516
3146 006516 012742 000173
3147 006522 005242
3148 006524 000000
3150
3151
3152 006526 125252
3153 006530 052652
3154 006532 053125
3155 006534 125252
3156 006536 000000
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170 006540 005212
3171 006542 022712 000112
3172 006546 001015
3173 006550 012700 006624
3174 006554 015037 006536
3175 006560 065037 006536
3176 006564 145037 006536
3177 006570 155037 006537
3178 006574 025037 006536
3179 006600 001411
3180
3181
3182
3183
3184 006602
3185 006602 012742 000174
3186 006606 005242
3187 006610 000000
3188
3189 006612 006526
3190 006614 006530
3191 006616 006531
3192 006620 006532
3193 006622 006534

```

```

DOP4:
MOV #173, -(R2) ; MOVE TO MAILBOX # ***** 173 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; RESULT OF MODE 4 INSTS. INCORRECT
; OR SEQUENCE ERROR

```

```

125252
52652
53125
125252
TBL1: 0

```

```

*****
THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
THE DATA TABLE USED IN THE PREVIOUS TEST. THE TEST IS IDENTICAL TO
THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
TABLE AND MODE 5 ADDRESSING. (SEE PREVIOUS TEST).
*****

```

```

*****
TEST 112 TEST MODE 5 W/ DOP INSTS.
*****

```

```

TST112: INC (R2) ; UPDATE TEST NUMBER
CMP #112, (R2) ; SEQUENCE ERROR?
BNE DOP5 ; BR TO ERROR HALT ON SEQ ERROR
MOV #TBL2+2, R0 ; INITIALIZE R0
MOV @-(R0), @#TBL1 ; TBL1=125252
ADD @-(R0), @#TBL1 ; TBL1=000377
BICB @-(R0), @#TBL1 ; TBL1=000252
BISB @-(R0), @#TBL1+1 ; TBL1=125252
CMP @-(R0), @#TBL1 ; CHECK RESULT
BEQ TST113

```

```

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

```

```

DOP5:
MOV #174, -(R2) ; MOVE TO MAILBOX # ***** 174 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; RESULT OF MODE 5 INSTS. INCORRECT
; OR SEQUENCE ERROR

```

```

TBL1-10
TBL1-6
TBL1-5
TBL1-4
TBL2: TBL1-2

```

```

3204
3205
3206
3207
3208 006624 005212
3209 006626 022712 000113
3210 006632 001022
3211 006634 012700 006532
3212 006640 016037 000002 006536
3213 006646 066037 000000 006536
3214 006654 146037 177777 006536
3215 006662 156037 177776 006537
3216 006670 026037 177774 006536
3217 006676 001404
3218
3219
3220
3221
3222 006700 012742 000175
3223 006704 005242
3224 006706 000000
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240 006710 005212
3241 006712 022712 000114
3242 006716 001022
3243 006720 012700 006616
3244 006724 017037 000004 006536
3245 006732 067037 000002 006536
3246 006740 147037 000000 006536
3247 006746 157037 177776 006537
3248 006754 027037 177774 006536
3249 006762 001404

```

```

*****
THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
THIS TIME THE DATA IS ACCESSED USING MODE 6. RO IS SET
TO POINT TO THE MIDDLE OF THE TABLE. THE TABLE IS ACCESSED FROM
BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
TESTS.
*****

```

```

*****
TEST 113 TEST MODE 6 W/ DOP INSTS.
*****
TST113: INC (R2) ;UPDATE TEST NUMBER
CMP #113 (R2) ;SEQUENCE ERROR?
BNE TST114-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #TBL1-4,RO ;INITIALIZE RO
MOV 2(RO),#TBL1 ;TBL1=125252
ADD 0(RO),#TBL1 ;TBL1=000377
BICB -1(RO),#TBL1 ;TBL1=000252
BISB -2(RO),#TBL1+1 ;TBL1=125252
CMP -4(RO),#TBL1 ;CHECK RESULT
BEQ TST114

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
; CONDITIONAL BRANCH INST. AND <===
; REPLACE THE MOVE INSTRUCTION <===
; WHICH FOLLOWS W/ 756 <===
MOV #175, -(R2) ;MOVE TO MAILBOX # ***** 175 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF MODE 6 INSTS. INCORRECT
; OR SEQUENCE ERROR

```

```

*****
THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
THE MODE 5 TESTS. THIS TIME THE DATA IS ACCESSED USING MODE 7.
RO IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
TEST. THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
IN THE MODE 7 INSTRUCTIONS. THE DATA RESULTS ARE IDENTICAL TO
THOSE EXPECTED IN THE MODE 5 TESTS.
*****

```

```

*****
TEST 114 TEST MODE 7 W/ DOP INSTS.
*****
TST114: INC (R2) ;UPDATE TEST NUMBER
CMP #114 (R2) ;SEQUENCE ERROR?
BNE TST115-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #TBL2-4,RO ;INITIALIZE RO
MOV 2(RO),#TBL1 ;TBL1=125252
ADD 2(RO),#TBL1 ;TBL1=000377
BICB 20(RO),#TBL1 ;TBL1=000252
BISB 2-2(RO),#TBL1+1 ;TBL1=125252
CMP 2-4(RO),#TBL1 ;CHECK RESULT
BEQ TST115

```

```

3250
3251
3252
3253
3254 006764 012742 000176      MOV      #176,-(R2)
3255 006770 005242      INC      -(R2)
3256 006772 000000      HALT
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 756
; MOVE TO MAILBOX # ***** 176 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF MODE 7 INSTS INCORRECT
; OR SEQUENCE ERROR

```

```

*****
THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
RO IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND
AN ROL INSTRUCTION IS EXECUTED WITH MODE 0. THE OPERATION IS CHECKED
BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
*****

```

```

*****
TEST 115      TEST ROTATE INSTRUCTIONS OF MODE 0
*****
TST115: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #115,(R2)   ;SEQUENCE ERROR?
          BNE     TST116-10    ;BR TO ERROR HALT ON SEQ ERROR
          MOV     #125252,RO    ;INITIALIZE DATA
          SEC     ;SET C-BIT
          ROL     RO           ;TRY ROL W/ MODE 0
          BVC     ROT0A        ;CC=0011
          BCC     ROT0A
          CMP     #052525,RO   ;CHECK DATA
          BEQ     ROT0B

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 767

```

```

3284 007026      ROT0A: MOV      #177,-(R2)   ;MOVE TO MAILBOX # ***** 177 *****
3285 007026 012742 000177      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
3286 007032 005242      HALT        ;ROL MODE 0 FAILED
3287 007034 000000      ROT0B: MOV     #125252,RO ;INITIALIZE DATA
3288 007036 012700 125252      SEC     ;SET C-BIT
3289 007042 000261      ROLB     RO      ;TRY ROL W/ MODE 0 EVEN BYTE
3290 007044 106100      BVC     ROT0C    ;CC=0011
3291 007046 102004      BCC     ROT0C
3292 007050 103003      CMP     #125125,RO ;CHECK DATA
3293 007052 022700 125125      BEQ     TST116

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 752

```

```

3298
3299 007060      ROT0C: MOV     #200,-(R2) ;MOVE TO MAILBOX # ***** 200 *****
3300 007060 012742 000200      INC     -(R2)   ;SET MSGTYP TO FATAL ERROR
3301 007064 005242      HALT        ;ROLB MODE 0 FAILED
3302 007066 000000      ;OR SEQUENCE ERROR
3303

```

3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359

007070 005212
007072 022712 000116
007076 001051
007100 005000
007102 012710 052525
007106 000241
007110 006110
007112 102005
007114 103404
007116 023727 000000 125252
007124 001404

007126
007126 012742 000201
007132 005242
007134 000000
007136 000261
007140 012710 125252
007144 106110
007146 102005
007150 103004
007152 022737 125125 000000
007160 001404

007162
007162 012742 000202
007166 005242
007170 000000
007172 012710 125252
007176 005000
007200 005200
007202 000261
007204 106110
007206 102005
007210 103004
007212 022737 052652 000000
007220 001404

```
*****
THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
THE DATA TO BE ROTATED IS IN LOC 0. R0 IS USED AS THE
ADDRESSING REGISTER. THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
THE C AND V BITS. THIS PROCEDURE IS THEN REPEATED TWICE MORE
TO TEST THE BYTE ROTATES. FIRST ON BYTE 0, THEN ON BYTE 1.
*****
TEST 116 TEST ROTATE INSTRUCTIONS W/ MODE 1
*****
TST116: INC (R2) ;UPDATE TEST NUMBER
CMP #116,(R2) ;SEQUENCE ERROR?
BNE TST117-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;POINT TO LOC. 0
MOV #52525,(R0) ;INITIALIZE DATA
CLC ;CLEAR C-BIT
ROL (R0) ;TRY ROL W/ MODE 1
BVC ROT1A ;CC=1010
BCS ROT1A
CMP #0,#125252 ;CHECK RESULT
BEQ ROT1B

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

ROT1A: MOV #201,-(R2) ;MOVE TO MAILBOX # ***** 201 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROL MODE 1 FAILED

ROT1B: SEC
MOV #125252,(R0) ;INITIALIZE DATA
ROLB (R0) ;TRY ROLB W/ MODE 1 EVEN BYTE
BVC ROT1C ;CC=1011
BCC ROT1C
CMP #125125,#0 ;TEST RESULT
BEQ ROT1D

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

ROT1C: MOV #202,-(R2) ;MOVE TO MAILBOX # ***** 202 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROLB W/ MODE 1 EVEN BYTE FAILED

ROT1D: MOV #125252,(R0)
CLR R0 ;POINT TO ODD BYTE
INC R0
SEC ;SET C-BIT
ROLB (R0) ;TRY ROLB W/ MODE 1 ODD BYTE
BVC ROT1E ;CC=0011
BCC ROT1E
CMP #052652,#0 ;CHECK DATA
BEQ TST117
```

3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415

007222 012742 000203
007222 005242
007226 005242
007230 000000

007232 005212 000117
007234 022712
007240 001057
007242 005000
007244 012710 173737
007250 000241
007252 006120
007254 103007
007256 022737 167676 000000
007264 001003
007266 005300
007270 005300
007272 001404

007274 012742 000204
007300 005242
007302 000000
007304 005000
007306 012710 004040
007312 000241
007314 106120
007316 103406
007320 022737 004100 000000
007326 001002
007330 005300
007332 001404

007334 012742 000205

ROT1E: MOV #203, -(R2)
INC -(R2)
HALT

TST117: INC (R2)
CMP #117, (R2)
BNE TST120-10
CLR R0
MOV #173737, (R0)
CLC
ROL (R0)+
BCC ROT2A
CMP #167676, @#0
BNE ROT2A
DEC R0
DEC R0
BEQ ROT2B

ROT2A: MOV #204, -(R2)
INC -(R2)
HALT

ROT2B: CLR R0
MOV #4040, (R0)
CLC
ROLB (R0)+
BCS ROT2C
CMP #4100, @#0
BNE ROT2C
DEC R0
BEQ ROT2D

ROT2C: MOV #205, -(R2)

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 727 <====

: MOVE TO MAILBOX # ***** 203 *****
: SET MSGTYP TO FATAL ERROR
: 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 117 TEST ROTATE INSTRUCTIONS W/ MODE 2
: *****
: UPDATE TEST NUMBER
: SEQUENCE ERROR?
: BR TO ERROR HALT ON SEQ ERROR
: POINT TO LOC 0
: INITIALIZE DATA
: CLEAR C-BIT
: TRY ROL W/ MODE 2
: CHECK C-BIT
: CHECK DATA
: BRANCH IF RESULT INCORRECT
: TEST R0

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

: MOVE TO MAILBOX # ***** 204 *****
: SET MSGTYP TO FATAL ERROR
: ROL W/ MODE 2 FAILED
: POINT TO LOC 0
: INITIALIZE DATA
: CLEAR C-BIT
: TRY ROLB W/ MODE 2 EVEN BYTE
: CHECK C-BIT
: CHECK DATA
: BRANCH IF DATA INCORRECT
: CHECK R0

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 743 <====

: MOVE TO MAILBOX # ***** 205 *****

3416	007340	005242		INC	-(R2)	: SET MSGTYP TO FATAL ERROR	
3417	007342	000000		HALT		: ROLB W/ MODE 2 EVEN BYTE FAILED	
3418	007344	005000		ROT2D: CLR	RO	: POINT TO LOC 0	
3419	007346	012710	004040	MOV	#4040, (RO)	: INITIALIZE DATA	
3420	007352	005200		INC	RO	: POINT TO ODD BYTE OF DATA	
3421	007354	000261		SEC		: SET C-BIT	
3422	007356	106120		ROLB	(RO)+	: TRY ROL W/ MODE 2 ODD BYTE	
3423	007360	103407		BCS	ROT2E	: CHECK C-BIT	
3424	007362	022737	010440 000000	CMP	#10440, @#0	: CHECK DATA	
3425	007370	001003		BNE	ROT2E	: BRANCH IF DATA INCORRECT	
3426	007372	005300		JEC	RO	: CHECK RO	
3427	007374	005300		DEC	RO		
3428	007376	001404		SEQ	TST120		
3429						: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
3430						: CONDITIONAL BRANCH INST. AND	<====
3431						: REPLACE THE MOVE INSTRUCTION	<====
3432						: WHICH FOLLOWS W/ 721	<====
3433	007400			ROT2E:			
3434	007400	012742	000206	MOV	#206, -(R2)	: MOVE TO MAILBOX # ***** 206 *****	
3435	007404	005242		INC	-(R2)	: SET MSGTYP TO FATAL ERROR	
3436	007406	000000		HALT		: ROLB W/ MODE 2 ODD BYTE FAILED	
3437						: OR SEQUENCE ERROR	

3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493

007410 005212
007412 022712 000120
007416 001051
007420 012737 052525 000000
007426 000261
007430 006137 000000
007434 103404
007436 022737 125253 000000
007444 001404

007446
007446 012742 000207
007452 005242
007454 000000
007456 012737 125252 000000
007464 000241
007466 106137 000000
007472 103004
007474 023727 000000 125124
007502 001404

007504
007504 012742 000210
007510 005242
007512 000000
007514 012737 125252 000000
007522 000261
007524 106137 000001
007530 103004
007532 022737 052652 000000
007540 001404

007542
007542 012742 000211
007546 005242
007550 000000

```
*****
: 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 120 TEST ROTATE INSTRUCTIONS /W MODE 3
*****
TST120: INC (R2) ; UPDATE TEST NUMBER
        CMP #120,(R2) ; SEQUENCE ERROR?
        BNE TST121-10 ; BR TO ERROR HALT ON SEQ ERROR
        MOV #52525,a#0 ; INITIALIZE DATA IN LOC 0
        SEC ; SET C-BIT
        ROL a#0 ; TRO ROL W/ MODE 3
        BCS ROT3A ; CHECK C-BIT
        CMP #125253,a#0 ; CHECK DATA
        BEQ ROT3B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
;
ROT3A: MOV #207,-(R2) ; MOVE TO MAILBOX # ***** 207 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; ROL W/ MODE 3 FAILED
ROT3B: MOV #125252,a#0 ; INITIALIZE DATA
        CLC ; CLEAR C-BIT
        ROLB a#0 ; TRY ROL W/ MODE 3 EVEN BYTE
        BCC ROT3C ; CHECK C-BIT
        CMP a#0,#125124 ; CHECK DATA
        BEQ ROT3D
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 746 <====
;
ROT3C: MOV #210,-(R2) ; MOVE TO MAILBOX # ***** 210 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; ROL W/ MODE 3 EVEN BYTE FAILED
ROT3D: MOV #125252,a#0 ; INITIALIZE DATA IN LOC. 0
        SEC ; SET C-BIT
        ROLB a#1 ; TRY ROL W/ MODE 3 ODD BYTE
        BCC ROT3E ; CHECK C-BIT
        CMP #052652,a#0 ; CHECK DATA
        BEQ TST121
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 727 <====
;
ROT3E: MOV #211,-(R2) ; MOVE TO MAILBOX # ***** 211 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; ROL W/ MODE 3 ODD BYTE FAILED
```

LU6

3494

; OR SEQUENCE ERROR

3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550

007552 005212
007554 022712 000121
007560 001016
007562 012737 070707 000000
007570 012700 000002
007574 000261
007576 006140
007600 103406
007602 022737 161617 000000
007610 001002
007612 005700
007614 001404

007616
007616 012742 000212
007622 005242
007624 000000

007626 005212
007630 022712 000122
007634 001021
007636 012737 007710 000000
007644 012700 000002
007650 012767 107070 000032
007656 000241
007660 006150
007662 103006

```
*****
;
; THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS. THE DATA IS
; STORED IN LOC. 0. RO IS SET TO 2 AND THE CARRY IS SET. AN ROL MODE 4
; IS USED TO ROTATE LOCATION 0 USING RO. THE DATA IS CHECKED
; AND THE C AND V BITS ARE TESTED. THE PROPER DECREMENTING OF
; RO IS VERIFIED.
;
;*****
;TEST 121 TEST MODE 4 W/ ROTATE INSTRUCTIONS
;*****
TST121: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
CMP #121,(R2) ;BR TO ERROR HALT ON SEQ ERROR
BNE TST122-10 ;INITIALIZE DATA IN LOC. 0
MOV #070707,R0 ;INITIALIZE RO AS PCINTER
MOV #2,RO ;SET C-BIT
SEC ;TRY ROL W/ MODE 4
ROL -(RO) ;CHECK C-BIT
BCS ROT4 ;CHECK DATA
CMP #161617,R0 ;BRANCH IF DATA INCORRECT
BNE ROT4 ;CHECK MODE 4 REGISTER
TST RO
BEQ TST122
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
; CONDITIONAL BRANCH INST. AND <===
; REPLACE THE MOVE INSTRUCTION <===
; WHICH FOLLOWS W/ 762 <===
```

```
ROT4: MOV #212,-(R2) ;MOVE TO MAILBOX # ***** 212 *****
;SET MSGTYP TO FATAL ERROR
INC -(R2) ;ROL MODE 4 FAILED
HALT ; 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).
; RO IS SET TO 2. THE CARRY IS CLEARED AND A MODE 5 ROL
; IS EXECUTED USING RO AS AN ADDRESSING REGISTER. THE DATA IS
; CHECKED, THE C AND V BITS TESTED, AND RO CHECKED FOR PROPER
; DECREMENTING.
;
;*****
```

```
*****
;TEST 122 TEST MODE 5 W/ ROTATE INSTRUCTIONS
;*****
TST122: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
CMP #122,(R2) ;BR TO ERROR HALT ON SEQ ERROR
BNE ROT5 ;MOVE POINTER TO LOC. 0
MOV #ROTX,R0 ;SET MODE 5 REG. TO LOC. 0
MOV #2,RO ;INITIALIZE DATA
MOV #107070,ROTX ;CLEAR C-BIT
CLC ;TRY ROL W/ MODE 5
ROL R0-(R0) ;CHECK C-BIT
BCC ROT5
```


3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649

007760 005212
007762 022712 000124
007766 001016
007770 012737 052525 007710
007776 012737 007710 010034
010004 000241
010006 006177 000022
010012 103404
010014 023727 007710 125252
010022 001405

010024
010024 012742 000215
010030 005242
010032 000000

010034 000000

010036 005212
010040 022712 000125
010044 001013
010046 012700 177400
010052 000300
010054 100404

010056 012742 000216
010062 005242
010064 000000

THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST). THE ROL INSTRUCTION
ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
(ROTXAD) FOLLOWING THE TEST CODE.

TEST 124 TEST MODE 7 W/ ROTATE INSTRUCTIONS

TST124: INC (R2) ; UPDATE TEST NUMBER
CMP #124, (R2) ; SEQUENCE ERROR?
BNE ROT7 ; BR TO ERROR HALT ON SEQ ERROR
MOV #52525, @ROTX ; INITIALIZE DATA
MOV @ROTX, @ROTXAD ; INITIALIZE ADDRESS POINTER
CLC ; CLEAR C-BIT
ROL @ROTXAD ; TRY ROL W/ MODE 7
BCS ROT7 ; CHECK C-BIT
CMP @ROTX, #125252 ; CHECK DATA
BEG TST125

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 762

ROT7: MOV #215, -(R2) ; MOVE TO MAILBOX # ***** 215 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; ROL W/ MODE 7 FAILED
; OR SEQUENCE ERROR

ROTXAD: 0

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

TEST 125 TEST MODE 0 W/ SWAB INST.

TST125: INC (R2) ; UPDATE TEST NUMBER
CMP #125, (R2) ; SEQUENCE ERROR?
BNE TST126-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #177400, RO ; MOVE TEST PATTERN TO RO
SWAB RO ; TRY SWAB MODE 0
BMI S80

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 774

MOV #216, -(R2) ; MOVE TO MAILBOX # ***** 216 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; SWAB DID NOT SET CC'S CORRECT

3650	010066	022700	000377	SBC:	CMP	#377, RD
3651	010072	001404			BEO	TST126
3652						
3653						
3654	010074	012742	000217		MOV	#217, -(R2)
3655	010100	005242			INC	-(R2)
3656	010102	000000			HALT	
3657						
3658						
3659						

;CHECK RESULT
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS w/ 765 <====
: MOVE TO MAILBOX # ***** 217 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF SWAB MODE G FAILED
: OR SEQUENCE ERROR

```

3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671 010104 005212
3672 010106 022712 000126
3673 010112 001011
3674 010114 012737 125652 000000
3675 010122 005000
3676 010124 000310
3677 010126 022737 125253 000000
3678 010134 001404
3679
3680
3681
3682
3683 010136 012742 000220
3684 010142 005242
3685 010144 000000
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699 010146 005212
3700 010150 022712 000127
3701 010154 001020
3702 010156 012737 125152 000000
3703 010164 005000
3704 010166 000320
3705 010170 022737 065252 000000
3706 010176 001404
3707
3708
3709
3710
3711 010200 012742 000221
3712 010204 005242
3713 010206 000000
3714 010210 162700 000002
3715 010214 001404

```

```

*****
THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION. THE TEST
PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE ADDRESSING
REGISTER IN THE MODE 1 SWAB. THE DATA RESULTS ARE CHECKED WITH
A COMPARE.
*****

```

```

*****
TEST 126 TEST MODE 1 W/ SWAB INST
*****

```

```

TST126: INC (R2) ;UPDATE TEST NUMBER
CMP #126,(R2) ;SEQUENCE ERROR?
BNE TST127-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125652,R#0 ;MOVE TEST PATTERN TO LOC. 0
CLR R0 ;R0=0
SWAB (R0) ;TRY SWAB MODE 1
CMP #125253,R#0 ;CHECK RESULT
BEQ TST127

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 767 (====
MOV #220,-(R2) ;MOVE TO MAILBOX # ***** 220 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF SWAB MODE 1 FAILED
; OR SEQUENCE ERROR

```

```

*****
THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION. THE TEST
PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE MODE
2 ADDRESSING REGISTER. THE RESULTS ARE CHECKED WITH A COMPARE.
R0 IS CHECKED FOR PROPER DECREMENTING.
*****

```

```

*****
TEST 127 TEST MODE 2 W/ SWAB INST
*****

```

```

TST127: INC (R2) ;UPDATE TEST NUMBER
CMP #127,(R2) ;SEQUENCE ERROR?
BNE TST130-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125152,R#0 ;MOVE TEST PATTERN TO LOC. 0
CLR R0 ;R0=0
SWAB (R0)+ ;TRY SWAB MODE 2
CMP #65252,R#0 ;CHECK RESULT
BEQ SB2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS \====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 767 (====
MOV #221,-(R2) ;MOVE TO MAILBOX # ***** 221 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF SWAB MODE 0 FAILED
SB2: SUB #2,R0 ;CHECK EFFECT OF REG.
BEQ TST130

```


3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788

010270 005212
010272 022712 000131
010276 001020
010300 012737 125652 000000
010306 012700 000002
010312 000340
010314 022737 125253 000000
010322 001404

010324 012742 000224
010330 005242
010332 000000
010334 005700
010336 001404

010340 012742 000225
010344 005242
010346 000000

```
*****
      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 131      TEST MODE 4 W/ SWAB INST
*****
TST131: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #131,(R2)   ;SEQUENCE ERROR?
          BNE     TST132-10    ;BR TO ERROR HALT ON SEQ ERROR
          MOV     #125652,2#0   ;MOVE TEST PATTERN TO LOC. 0
          MOV     #2,R0        ;SET UP REGISTER POINTER
          SWAB   -(R0)         ;TRY SWAB MODE 4
          CMP     #125253,2#0  ;CHECK RESULT
          BEQ    SB4
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
          ;          CONDITIONAL BRANCH INST. AND
          ;          REPLACE THE MOVE INSTRUCTION
          ;          WHICH FOLLOWS W/ 766
          ;          MOVE TO MAILBOX # ***** 224 *****
          ;          SET MSGTYP TO FATAL ERROR
          ;          RESULT OF SWAB INCORRECT
          ;          CHECK EFFECT ON REG.
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
          ;          CONDITIONAL BRANCH INST. AND
          ;          REPLACE THE MOVE INSTRUCTION
          ;          WHICH FOLLOWS W/ 760
          ;          MOVE TO MAILBOX # ***** 225 *****
          ;          SET MSGTYP TO FATAL ERROR
          ;          REGISTER VALUE INCORRECT
          ;          OR SEQUENCE ERROR
          ;          *****

SB4:     TST     R0
          BEQ    TST132
          TST132
```

```

3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802 010350 005212
3803 010352 022712 000132
3804 010356 001021
3805 010360 012700 010436
3806 010364 012767 125125 000040
3807 010372 000350
3808 010374 022767 052652 000030
3809 010402 001404
3810
3811
3812
3813
3814 010404 012742 000226
3815 010410 005242
3816 010412 000000
3817 010414 020027 010434
3818 010420 001406
3819
3820
3821
3822
3823 010422
3824 010422 012742 000227
3825 010426 005242
3826 010430 000000
3827
3828 010432 000000
3829 010434 010432
3830

```

```

*****
THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION. THE TEST USES
TWO LOCATIONS FOLLOWING THE TEST CODE. SBSX HOLDS THE DATA;
SBSXAD IS A POINTER TO THE DATA LOCATION. THE DATA IS MOVED TO
SBSX AND RO IS SET TO TWO PLUS THE ADDRESS OF SBSXAD FOLLOWING
THE MODE 5 SWAB SBSX IS CHECKED FOR THE PROPER DATA. RO IS
CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
*****
TEST 132 TEST MODE 5 W/ SWAB INST.
*****
TST132: INC (R2) ;UPDATE TEST NUMBER
CMP #132,(R2) ;SEQUENCE ERROR?
BNE SBS ;BR TO ERROR HALT ON SEQ ERROR
MOV #SBSXAD+2,RO ;SET UP POINTER TO WORK LOCATION
MOV #125125,SBSX ;MOVE PATTERN TO WORK LOCATION
SWAB @-(RO) ;TRY SWAB MODE 5
CMP #52652,SBSX ;CHECK RESULT
BEQ SBSA
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
; MOVE TO MAILBOX # ***** 226 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF SWAB INCORRECT
; CHECK RESULT OF REG.
SBSA: CMP RO,#SBSXAD
BEQ TST133
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
SBS: MOV #227,-(R2) ;MOVE TO MAILBOX # ***** 227 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;REGISTER VALUE INCORRECT
; OR SEQUENCE ERROR
;WORK LOCATION
SBSX: 0
SBSXAD: SBSX

```

```

3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844 010436 005212
3845 010440 022712 000133
3846 010444 001013
3847 010446 012767 125125 000030
3848 010454 012700 010476
3849 010460 000360 000006
3850 010464 022760 052652 000006
3851 010472 001405
3852
3853
3854
3855
3856 010474
3857 010474 012742 000230
3858 010500 005242
3859 010502 000000
3860
3861 010504 000000
3862

```

```

*****
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 133 TEST MODE 6 W/ SWAB INST.
*****
TST133: INC (R2) ;UPDATE TEST NUMBER
CMP #133,(R2) ;SEQUENCE ERROR?
BNE SB6 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125125,SB6X ;MOVE PATTERN TO WORK LOCATION
MOV #SB6X-6,R0 ;MOVE OFFSET POINTER TO R0
SWAB 6(R0) ;TRY SWAB W/ MODE 6
CMP #52652,6(R0) ;CHECK RESULT
BEQ TST134

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

SB6: MOV #230,-(R2) ;MOVE TO MAILBOX # ***** 230 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF SWAB INCORRECT
; OR SEQUENCE ERROR
SB6X: 0 ;WORK LOCATION

```

3862
 3864
 3865
 3866
 3867
 3868
 3869
 3870
 3871
 3872
 3873
 3874
 3875
 3876
 3877 010506 005212
 3878 010510 022712 000134
 3879 010514 001013
 3880 010516 012767 177400 000030
 3881 010524 012700 010464
 3882 010530 000370 000072
 3883 010534 027027 000072 000377
 3884 010542 001406
 3885
 3886
 3887
 3888
 3889 010544
 3890 010544 012742 000231
 3891 010550 005242
 3892 010552 000000
 3893
 3894 010554 000000
 3895 010556 010554
 3896

```

*****
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. RO 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 134 TEST MODE 7 W/ SWAB INST.
*****
TST134: INC (R2) ;UPDATE TEST NUMBER
          CMP #134,(R2) ;SEQUENCE ERROR?
          BNE SB7 ;BR TO ERROR HALT ON SEQ ERROR
          MOV #177400,SB7X ;MOVE PATTERN TO WORK LOCATION
          MOV #SB7XAD-72,RO ;MOVE OFFSET POINTER TO RO
          SWAB @72(RO) ;TRY SWAB MODE 7
          CMP @72(RO),#377 ;CHECK RESULTS
          BEQ TST135

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

SB7: MOV #231,-(R2) ;MOVE TO MAILBOX # ***** 231 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB INCORRECT
          OR SEQUENCE ERROR
SB7X: 0 ;WORK LOCATION
SB7XAD: SB7X ;POINTER TO WORK LOCATION

```

3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952

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 135 TEST THE JMP INSTRUCTION IN ALL MODES

```

TST135: INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #135, (R2)    ;SEQUENCE ERROR?
        BNE     JMPCK+6        ;BR TO ERROR HALT ON SEQ ERROR
        CLR     JMPSEQ         ;ESTABLISH A SEQUENCE CHECKER
        MOV     #JMP2, R0      ;SET R0=JUMP TARGET
        JMP     (R0)           ;TRY JMP MODE 1
JMP3:   CMP     #.+2, R0       ;CHECK RESULT OF MODE 2 JUMP
        BEQ     JMP3A

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 770 <====
        MOV     #232, -(R2)    ;MOVE TO MAILBOX # ***** 232 *****
        INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
        HALT     .             ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
JMP3A:  CMP     JMPSEQ, #1     ;MAKE SURE JUMPS ARE IN SEQUENCE: JMPSEQ=1?
        BEQ     JMP3B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 760 <====

```

```

010560 005212
010562 022712 000135
010566 001150
010570 005067 000326
010574 012700 010654
010600 000110
010602 022700 010604
010606 001404

010610 012742 000232
010614 005242
010616 000000
010620 026727 000276 000001
010626 001404

```

3953	010630	012742	000233		MOV	#233, -(R2)	; MOVE TO MAILBOX # ***** 233 *****
3954	010634	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
3955	010636	000000			HALT		; SHOULD BE HERE FROM JMP MODE 2 ONLY
3956	010640	012700	010652	JMP3B:	MOV	#IIMP4, R0	; POINT R0 TO INDIRECT JMP ADDR.
3957	010644	005267	000252		INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
3958	010650	000130			JMP	2(R0)+	; TRY JMP MODE 3
3959	010652	010704		IIMP4:	JMP4		; ADDRESS INDIRECT JUMP
3960							
3961	010654	005767	000242	JMP2:	TST	JMPSEQ	; CHECK THAT JMPs ARE IN SEQUENCE: JMPSEQ=0?
3962	010660	001404			BEQ	JMP2A	
3963							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
3964							; CONDITIONAL BRANCH INST. AND (====
3965							; REPLACE THE MOVE INSTRUCTION (====
3966							; WHICH FOLLOWS W/ 743 (====
3967	010662	012742	000234		MOV	#234, -(R2)	; MOVE TO MAILBOX # ***** 234 *****
3968	010666	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
3969	010670	000000			HALT		; SHOULD BE HERE FROM JMP MODE 1 ONLY
3970	010672	005267	000224	JMP2A:	INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
3971	010676	012700	010602		MOV	#JMP3, R0	; SET R0=JUMP TARGET
3972	010702	000120			JMP	(R0)+	; TRY A JUMP MODE 2 TO "JMP3"
3973	010704	022700	010654	JMP4:	CMP	#IIMP4+2, R0	; CHECK RESULT OF REGISTER IN MODE 3 JUMP
3974	010710	001404			BEQ	JMP4A	
3975							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
3976							; CONDITIONAL BRANCH INST. AND (====
3977							; REPLACE THE MOVE INSTRUCTION (====
3978							; WHICH FOLLOWS W/ 727 (====
3979	010712	012742	000235		MOV	#235, -(R2)	; MOVE TO MAILBOX # ***** 235 *****
3980	010716	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
3981	010720	000000			HALT		; REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
3982	010722	022767	000002	000172	JMP4A:	CMP	#2, JMPSEQ
3983	010730	001404			BEQ	JMP4B	; CHECK JUMP SEQUENCE: JMPSEQ=2?
3984							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
3985							; CONDITIONAL BRANCH INST. AND (====
3986							; REPLACE THE MOVE INSTRUCTION (====
3987							; WHICH FOLLOWS W/ 717 (====
3988	010732	012742	000236		MOV	#236, -(R2)	; MOVE TO MAILBOX # ***** 236 *****
3989	010736	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
3990	010740	000000			HALT		; SHOULD BE ONLY FROM MODE 3 JUMP
3991	010742	012700	011012	JMP4B:	MOV	#JMP5+2, R0	; SET UP POINTER TO JUMP TARGET
3992	010746	005267	000150		INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
3993	010752	000140			JMP	-(R0)	; TRY JUMP MODE 4 TO "JMP4"
3994							
3995	010754	022767	000004	000140	JMP6:	CMP	#4, JMPSEQ
3996	010762	001404			BEQ	JMP6A	; CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
3997							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
3998							; CONDITIONAL BRANCH INST. AND (====
3999							; REPLACE THE MOVE INSTRUCTION (====
4000							; WHICH FOLLOWS W/ 702 (====
4001	010764	012742	000237		MOV	#237, -(R2)	; MOVE TO MAILBOX # ***** 237 *****
4002	010770	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
4003	010772	000000			HALT		; SHOULD BE HERE ONLY FROM MODE 5 JUMP
4004	010774	012700	011442	JMP6A:	MOV	#JMP7+376, R0	; SET UP OFFSET POINTER TO JUMP TARGET
4005	011000	005267	000116		INC	JMPSEQ	; UPDATE JUMP SEQUENCE
4006	011004	000160	177402		JMP	-376(R0)	; TRY MODE 6 JUMP
4007							
4008	011010	022767	000003	000104	JMP5:	CMP	#3, JMPSEQ
							; CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?

4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067 011124 005212
4068 011126 022712 000136
4069 011132 001001
4070 011134 000402
4071 011136 000137 011572
4072
4073 011142 012706 000500
4074 011146 012700 011254
4075 011152 005037 011552
4076 011156 005031
4077 011160 005131
4078 011162 004110
4079
4080
4081 011164
4082 011164 012742 000243
4083 011170 005242
4084 011172 000000
4085
4086 011174 022737 000001 011552
4087 011202 001014
4088 011204 020127 011336
4089 011210 001011
4090 011212 022706 000476
4091 011216 001006
4092 011220 022716 125252
4093 011224 001003
4094 011226 022700 011176
4095 011232 001404
4096
4097
4098
4099
4100 011234
4101 011234 012742 000244
4102 011240 005242

```

*****
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 136 TEST JSR INSTRUCTION W/ ALL MODES
*****
TST136: INC (R2) ;UPDATE TEST NUMBER
        CMP #136,(R2) ;SEQUENCE ERROR?
        BNE JSR0 ;BR TO ERROR HALT ON SEQ ERROR
        BR JSR1
JSR0: JMP @#JSRCK1
JSR1: MOV #STBOT,R6 ;SET STACK POINTER
      MOV #JSR2,R0 ;SET TARGET ADDRESS
      CLR @#JSRSEQ ;INITIALIZE SEQUENCE CHECKER
      CLR R1 ;INITIALIZE R1
      COM R1
      JSR R1,(R0) ;TRY JSR MODE 1
      ; TO SCOPE: REPLACE THE MOVE INSTRUCTION (====
      ; FOLLOWING W/ 774 (====
JSR1A: MOV #243,-(R2) ;MOVE TO MAILBOX # ***** 243 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;JSR MODE 1 FAILED
JSR3: CMP #1,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=1?
      BNE JSR3A ;BRANCH IF OUT OF SEQUENCE
      CMP R1,#JSR4 ;PROPER PC SAVED?
      BNE JSR3A ;BRANCH IF PC WRONG
      CMP #STBOT-2,R6 ;STACK POINTER DECREMENTED?
      BNE JSR3A ;BRANCH IF SP WRONG
      CMP #125252,(R6) ;REG SAVED ON STACK?
      BNE JSR3A ;BRANCH IF REG. NOT SAVED
      CMP #JSR3+2,R0 ;MODE 2 INCREMENT CORRECT?
      BEQ JSR3B
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
      ; CONDITIONAL BRANCH INST. AND (====
      ; REPLACE THE MOVE INSTRUCTION (====
      ; WHICH FOLLOWS W/ 740 (====
JSR3A: MOV #244,-(R2) ;MCVE TO MAILBOX # ***** 244 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR

```



```

4159 011432 005237 011552 JSR6B: INC 2#JSRSEQ ;UPDATE SEQUENCE CHECKER
4160 011436 004167 000000 JSR R1,JSR7 ;TRY JSR MODE 6
4161 011442 022767 000003 000102 JSR5: CMP 3#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=3?
4162 011450 001006 JSR5A ;BRANCH IF OUT OF SEQUENCE
4163 011452 022701 011376 CMP 4#JSR6,R1 ;PROPER PC SAVED?
4164 011456 001003 BNE JSR5A ;BRANCH IF PC WRONG
4165 011460 022700 011442 CMP 5#JSR5,R0 ;CHECK MODE 4 REGISTER
4166 011464 001404 BEQ JSR5B
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 623 (====
4171 011466 JSR5A:
4172 011466 012742 000250 MOV 250,-(R2) ;MOVE TO MAILBOX # ***** 250 *****
4173 011472 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4174 011474 000000 HALT ;JSR MODE 4 MALFUNCTIONED
4175 011476 005237 011552 JSR5B: INC 2#JSRSEQ ;UPDATE SEQUENCE CHECKER
4176 011502 012700 011550 MOV 4#JSR6AD+2,R0 ;POINT R0 TO TARGET ADDRESS
4177 011506 004150 JSR R1,4-(R0) ;TRY JSR MODE 5
4178
4179 011510 022737 000005 011552 JSR7: CMP 5,2#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=5?
4180 011516 001003 BNE JSR7A ;BRANCH IF OUT OF SEQUENCE
4181 011520 022701 011442 CMP 4#JSR5,R1 ;PROPER PC SAVED?
4182 011524 001404 BEQ JSR7B
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 603 (====
4187 011526 JSR7A:
4188 011526 012742 000251 MOV 251,-(R2) ;MOVE TO MAILBOX # ***** 251 *****
4189 011532 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4190 011534 000000 HALT ;JSR MODE 6 FAILED
4191 011536 005237 011552 JSR7B: INC 2#JSRSEQ ;UPDATE SEQUENCE CHECKER
4192 011542 004177 000002 JSR R1,2#JSRCKAD ;TRY JSR MODE 7
4193
4194 JSR6AD: JSR6 ;MODE 5 TARGET ADDRESS
4195 011550 011554 JSRCKAD: JSRCK ;MODE 7 TARGET ADDRESS
4196 011552 000000 JSRSEQ: 0 ;SEQUENCE CHECKER
4197
4198 011554 022767 000006 177770 JSRCK: CMP 6,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=6?
4199 011562 001003 BNE JSRCK1 ;BRANCH IF OUT OF SEQUENCE
4200 011564 022701 011546 CMP 4#JSR6AD,R1 ;PROPER PC SAVED?
4201 011570 001404 BEQ TEST137
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
; CONDITIONAL BRANCH INST. AND (====
; REPLACE THE MOVE INSTRUCTION (====
; WHICH FOLLOWS W/ 561 (====
4206 011572 JSRCK1:
4207 011572 012742 000252 MOV 252,-(R2) ;MOVE TO MAILBOX # ***** 252 *****
4208 011576 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4209 011600 000000 HALT ;JSR MODE 7 MALFUNCTIONED
; OR SEQUENCE ERROR
;
;*****
;

```

THIS TEST VERIFIES THE RTS INSTRUCTION. THE STACK POINTER IS INITIALIZED AND A TEST PATTERN STORED ON STACK. R0 IS LOADED WITH RETURN ADDRESS. AN RTS IS EXECUTED, AND, AT THE TARGET ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE STACK.

 TEST 137 TEST RTS INSTRUCTION

4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244

```

011602 005212
011604 022712 000137
011610 001016
011612 012706 000500
011616 012746 052525
011622 012700 011640
011626 000200

011630 012742 000253
011634 005242
011636 000000
011640 022700 052525
011644 001404

011646 012742 000254
011652 005242
011654 000000
  
```

```

TEST137: INC (R2) ; UPDATE TEST NUMBER
          CMP #137,(R2) ; SEQUENCE ERROR?
          BNE TST140-10 ; BR TO ERROR HALT ON SEQ ERROR
          MOV #STBOT,R6 ; INITIALIZE STACK POINTER
          MOV #52525,-(R6) ; INITIALIZE TOP OF STACK
          MOV #RTS1,R0 ; INITIALIZE RETURN REGISTER
          RTS R0 ; TRY RTS THROUGH R0
          ; TO SCOPE: REPLACE THE MOVE INSTRUCTION (====
          ; FOLLOWING W/ 770 (====
          MOV #253,-(R2) ; MOVE TO MAILBOX # ***** 253 *****
          INC -(R2) ; SET MSGTYP TO FATAL ERROR
          HALT ; RTS FAILED
          RTS1: CMP #52525,R0 ; CHECK THAT R0 RESTORED FROM STACK
          BEQ TST140
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (====
          ; CONDITIONAL BRANCH INST. AND (====
          ; REPLACE THE MOVE INSTRUCTION (====
          ; WHICH FOLLOWS W/ 762 (====
          MOV #254,-(R2) ; MOVE TO MAILBOX # ***** 254 *****
          INC -(R2) ; SET MSGTYP TO FATAL ERROR
          HALT ; RTS MALFUNCTIONED
          ; OR SEQUENCE ERROR
  
```

4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300

011656 005212
011660 022712 000140
011664 001022
011666 000277
011670 000251
011672 012700 100000
011676 101402
011700 102401
011702 100404

011704
011704 012742 000255
011710 005242
011712 000000

011714 000277
011716 000244
011720 012700 000000
011724 101002
011726 102401
011730 100004

011732
011732 012742 000256
011736 005242
011740 000000

011742 005212

```
*****
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 140 TEST MOV INSTRUCTION
*****
TST140: INC (R2) ;UPDATE TEST NUMBER
CMP #140,(R2) ;SEQUENCE ERROR?
BNE TST141-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=0110
+CLN!CLC
MOV #100000,R0 ;CC=1000
BLOS MOV1
BVS MOV1
BMI MOV2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV1: MOV #255,-(R2) ;MOVE TO MAILBOX # ***** 255 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOV DID NOT SET CC'S CORRECTLY
MOV2: SCC ;CC=1011
CLZ
MOV #0,R0 ;CC=0101
BHI MOV3 ;C OR Z = 0?
BVS MOV3 ;V=1?
BPL TST141
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
MOV3: MOV #256,-(R2) ;MOVE TO MAILBOX # ***** 256 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOV DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR
*****
TEST 141 TEST BIT INSTRUCTION
*****
TST141: INC (R2) ;UPDATE TEST NUMBER
*****
```

```

4301 011744 022712 000141      CMP      #141,(R2)      ;SEQUENCE ERROR?
4302 011750 001024      BNE     TST142-10     ;BR TO ERROR HALT ON SEQ ERROR
4303 011752 012700 100001      MOV     #100001,R0
4304 011756 000277      SCC
4305 011760 000251      +CLN!CLC             ;CC=0110
4306 011762 032700 100000      BIT     #100000,R0    ;CC=1000
4307 011766 101402      BLOS   BIT1
4308 011770 102401      BVS   BIT1
4309 011772 100404      BMI   BIT2
4310
4311
4312
4313
4314 011774      BIT1:
4315 011774 012742 000257      MOV     #257,-(R2)    ;MOVE TO MAILBOX # ***** 257 *****
4316 012000 005242      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4317 012002 000000      HALT
4318
4319 012004 000277      BIT2: SCC             ;CC=1011
4320 012006 000244      CLZ
4321 012010 032700 077776      BIT     #77776,R0    ;CC=0101
4322 012014 101002      BHI   BIT3
4323 012016 102401      BVS   BIT3
4324 012020 100004      BPL   TST142
4325
4326
4327
4328
4329 012022      BIT3:
4330 012022 012742 000260      MOV     #260,-(R2)    ;MOVE TO MAILBOX # ***** 260 *****
4331 012026 005242      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4332 012030 000000      HALT
4333
4334
4335
4336
4337
4338 012032 005212      TST142: INC (R2)      ;UPDATE TEST NUMBER
4339 012034 022712 000142      CMP     #142,(R2)    ;SEQUENCE ERROR?
4340 012040 001024      BNE     TST143-10     ;BR TO ERROR HALT ON SEQ ERROR
4341 012042 012700 177777      MOV     #177777,R0
4342 012046 000277      SCC
4343 012050 000251      +CLN!CLC             ;CC=0110
4344 012052 042700 077777      BIC     #77777,R0    ;CC=1000
4345 012056 101402      BLOS   BIC1
4346 012060 102401      BVS   BIC1
4347 012062 100404      BMI   BIC2
4348
4349
4350
4351
4352 012064      BIC1:
4353 012064 012742 000261      MOV     #261,-(R2)    ;MOVE TO MAILBOX # ***** 261 *****
4354 012070 005242      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
4355 012072 000000      HALT
4356 012074 000277      BIC2: SCC             ;CC=1011

;*****
;TEST 142 TEST BIC INSTRUCTION
;*****
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 767

```


4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466

012214 005212
012216 022712 000144
012222 001037
012224 012700 077777
012230 000257
012232 000264
012234 005200
012236 101402
012240 100001
012242 102404

012244
012244 012742 000265
012250 005242
012252 000000
012254 052700 077777
012260 000261
012262 000244
012264 005200
012266 100403
012270 102402
012272 103001
012274 001404

012276
012276 012742 000266
012302 005242
012304 000000

012306 000277
012310 000241
012312 005200
012314 101402
012316 100401
012320 100004

THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
DEC INSTRUCTIONS. THESE INSTRUCTIONS BOTH EFFECT THE C AND V
BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
UPON THE DATA RESULTS. THE SAME PROCEDURE IS USED. THE CONDITION
CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
DIFFERENT COMBINATIONS OF THE C AND V BITS.

TEST 144 TEST INC INSTRUCTION

TST144: INC (R2) ;UPDATE TEST NUMBER
CMP #144,(R2) ;SEQUENCE ERROR?
BNE TST145-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #077777,R0 ;RO=077777
CCC ;CC=0100
SEZ ;
INC R0 ;CC=1010 RO=10000
BLOS INC1
BPL INC1
BVS INC2

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

INC1: MOV #265,-(R2) ;MOVE TO MAILBOX # ***** 265 *****
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INC DID NOT SET CC'S CORRECTLY
INC2: BIS #77777,R0 ;RO=177777
SEC ;CC=1011
CLZ ;
INC R0 ;CC=0101 RC=0
BMI INC3
BVS INC3
BCC INC3
BEQ INC4

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 753 <====
<====
<====
<====

INC3: MOV #266,-(R2) ;MOVE TO MAILBOX # ***** 266 *****
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INC DID NOT SET CC'S CORRECTLY

INC4: SCC ;CC=1110
CLC ;
INC R0 ;CC=0000 RO=1
BLOS INC5
BMI INC5
BPL TST145

```

4467
4468
4469
4470
4471 012322
4472 012322 012742 000267
4473 012326 005242
4474 012330 000000
4475
4476
4477
4478
4479
4480 012332 005212
4481 012334 022712 000145
4482 012340 001051
4483 012342 012700 000002
4484 012346 000277
4485 012350 005300
4486 012352 100403
4487 012354 001402
4488 012356 102401
4489 012360 103404
4490
4491
4492
4493
4494 012362
4495 012362 012742 000270
4496 012366 005242
4497 012370 000000
4498 012372 000261
4499 012374 000244
4500 012376 005300
4501 012400 101002
4502 012402 100401
4503 012404 102004
4504
4505
4506
4507
4508 012406
4509 012406 012742 000271
4510 012412 005242
4511 012414 000000
4512 012416 000277
4513 012420 000251
4514 012422 005300
4515 012424 101402
4516 012426 102401
4517 012430 100404
4518
4519
4520
4521
4522 012432

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 741 <====

INCS:
MOV #267, -(R2) ; MOVE TO MAILBOX # ***** 267 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; INC DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

;*****
;TEST 145 TEST DEC INSTRUCTION
;*****

TST145: INC (R2) ; UPDATE TEST NUMBER
CMP #145, (R2) ; SEQUENCE ERROR?
BNE TST146-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #2, R0 ; R0=2
SCC ; CC=1111
DEC R0 ; CC=0001 R0=1
BMI DEC1
BEQ DEC1
BVS DEC1
BCS DEC2

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

DEC1:
MOV #270, -(R2) ; MOVE TO MAILBOX # ***** 270 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; DEC DID NOT SET CC'S CORRECTLY
DEC2:
SEC ; CC=1011
CLZ
DEC R0 ; CC=0101 R0=0
BHI DEC3
BMI DEC3
BVC DEC4

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

DEC3:
MOV #271, -(R2) ; MOVE TO MAILBOX # ***** 271 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; DEC DID NOT SET CC'S CORRECTLY
DEC4:
SCC ; CC=0110
+CLN!CLC ; CC=1000 R0=177777
DEC R0
BLOS DEC5
BVS DEC5
BMI DEC6

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

DEC5:

4523	012432	012742	000272		MOV	#272, -(R2)	:MOVE TO MAILBOX # ***** 272 *****
4524	012436	005242			INC	-(R2)	:SET MSGTYP TO FATAL ERROR
4525	012440	000000			HALT		:DEC DID NOT SET CC'S CORRECTLY
4526	012442	042700	077777	DEC6:	BIC	#77777, R0	:R0=100000
4527	012446	000277			SCC		:CC=0101
4528	012450	000252			+CLN!CLV		
4529	012452	005300			DEC	R0	:CC=1011 R0=77777
4530	012454	100403			BMI	DEC7	:CC=0011
4531	012456	001402			BEQ	DEC7	
4532	012460	102001			BVC	DEC7	
4533	012462	103404			BOS	TST146	
4534							: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4535							: CONDITIONAL BRANCH INST. AND
4536							: REPLACE THE MOVE INSTRUCTION
4537							: WHICH FOLLOWS W/ 727
4538	012464			DEC7:			'====
4539	012464	012742	000273		MOV	#273, -(R2)	:MOVE TO MAILBOX # ***** 273 *****
4540	012470	005242			INC	-(R2)	:SET MSGTYP TO FATAL ERROR
4541	012472	000000			HALT		:DEC DID NOT SET CC'S CORRECTLY
4542							: OR SEQUENCE ERROR
4543							<====

4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599

012474 005212
012476 022712 000146
012502 001007
012504 000277
012506 000244
012510 005000
012512 100403
012514 102402
012516 103401
012520 001404

012522
012522 012742 000274
012526 005242
012530 000000

012532 005212
012534 022712 000147
012540 001022
012542 000277
012544 000244
012546 005700
012550 100403
012552 102402
012554 103401
012556 001404

012560
012560 012742 000275
012564 005242
012566 000000
012570 005300
012572 000277

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 146 TEST CLR INSTRUCTION

TST146: INC (R2) ;UPDATE TEST NUMBER
CMP #146,(R2) ;SEQUENCE ERROR?
BNE TST147-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
CLR RO ;CC=0100 RO=0
BMI CLR1
BVS CLR1
BCS CLR1
BEQ TST147

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

CLR1.
MOV #274,-(R2) ;MOVE TO MAILBOX # ***** 274 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

;TEST 147 TEST TST INSTRUCTION

TST147: INC (R2) ;UPDATE TEST NUMBER
CMP #147,(R2) ;SEQUENCE ERROR?
BNE TST150-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
TST RO ;CC=0100
BMI TEST1
BVS TEST1
BCS TEST1
BEQ TEST2

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

TEST1: MOV #275,-(R2) ;MOVE TO MAILBOX # ***** 275 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST DID NOT SET CC'S CORRECTLY
TEST2: DEC RO ;MAKE RO NEGATIVE
SCC ;CC=0111

```

4600 012574 000250          CLN
4601 012576 005700          TST          RO          ;CC=1000
4602 012600 101402          BLOS        TEST3
4603 012602 102401          BVS        TEST3
4604 012604 100404          BMI        TST150
4605
4606
4607
4608
4609 012606          TEST3:
4610 012606 012742 000276          MOV        #276, -(R2)      ;MOVE TO MAILBOX # ***** 276 *****
4611 012612 005242          INC        -(R2)          ;SET MSGTYP TO FATAL ERROR
4612 012614 000000          HALT
4613
4614
4615
4616
4617
4618 012616 005212          ;*****
4619 012620 022712 000150          ;TEST 150      TEST SWAB INSTRUCTION
4620 012624 001023          ;*****
4621 012626 012700 170000          TST150: INC        (R2)          ;UPDATE TEST NUMBER
4622 012632 000277          CMP        #150, (R2)      ;SEQUENCE ERROR?
4623 012634 000250          BNE        TST151-10      ;BR TO ERROR HALT ON SEQ ERROR
4624 012636 000300          MOV        #170000, RO     ;RO=170000
4625 012640 101402          SCC
4626 012642 102401          CLN          ;CC=0111
4627 012644 100404          SWAB       RO          ;CC=1000  RO=360
4628
4629
4630
4631
4632 012646          SWB1:
4633 012646 012742 000277          MOV        #277, -(R2)      ;MOVE TO MAILBOX # ***** 277 *****
4634 012652 005242          INC        -(R2)          ;SET MSGTYP TO FATAL ERROR
4635 012654 000000          HALT          ;SWAB DID NOT SET CC'S CORRECTLY
4636 012656 000277          SWB2: SCC
4637 012660 000244          CLZ
4638 012662 000300          SWAB       RO          ;CC=0100  RO=170000
4639 012664 102403          BVS        SWB3
4640 012666 103402          BCS        SWB3
4641 012670 100401          BMI        SWB3
4642 012672 001404          BEQ        TST151
4643
4644
4645
4646
4647 012674          SWB3:
4648 012674 012742 000300          MOV        #300, -(R2)      ;MOVE TO MAILBOX # ***** 300 *****
4649 012700 005242          INC        -(R2)          ;SET MSGTYP TO FATAL ERROR
4650 012702 000000          HALT
4651

```

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

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 77C

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 755

4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707

012704 005212
012706 022712 000151
012712 001062
012714 012700 040000
012720 000277
012722 062700 030000
012726 101402
012730 102401
012732 100004

012734
012734 012742 000301
012740 005242
012742 000000
012744 000264

012746 062700 010000
012752 101402
012754 102001
012756 100404

012760
012760 012742 000302
012764 005242
012766 000000
012770 000257
012772 000270
012774 062700 100000
013000 101002
013002 102001
013004 100004

013006

```
*****
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. THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.
*****
TEST 151      TEST ADD INSTRUCTION
*****
TST151: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #151,(R2)   ;SEQUENCE ERROR?
          BNE      TST152-10   ;BR TO ERROR HALT ON SEQ ERROR
          MOV      #40000,R0   ;RO=40000
          SCC      ;CC=1111
          ADD      #30000,R0   ;CC=0000 RO=70000
          BLOS    ADD1
          BVS     ADD1
          BPL     ADD2
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ;          CONDITIONAL BRANCH INST. AND <====
          ;          REPLACE THE MOVE INSTRUCTION <====
          ;          WHICH FOLLOWS W/ 770 <====
          ADD1:  MOV      #301,-(R2) ;MOVE TO MAILBOX # ***** 301 *****
          INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
          HALT    ;ADD DID NOT SET CC'S CORRECTLY
          ADD2:  SEZ      ;CC=0100
          ADD     #10000,R0 ;CC=1010 40=100000
          BLOS   ADD3
          BVC    ADD3
          BMI    ADD4
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ;          CONDITIONAL BRANCH INST. AND <====
          ;          REPLACE THE MOVE INSTRUCTION <====
          ;          WHICH FOLLOWS W/ 756 <====
          ADD3:  MOV      #302,-(R2) ;MOVE TO MAILBOX # ***** 302 *****
          INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
          HALT    ;ADD DID NOT SET CC'S CORRECTLY
          ADD4:  CCC      ;CC=1000
          SEN
          ADD     #100000,R0 ;CC=0111 RO=0
          BHI    ADD5
          BVC    ADD5
          BPL    ADD6
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ;          CONDITIONAL BRANCH INST. AND <====
          ;          REPLACE THE MOVE INSTRUCTION <====
          ;          WHICH FOLLOWS W/ 743 <====
          ADD5:
```

```

4708 013006 012742 000303      MOV      #303, -(R2)      ;MOVE TO MAILBOX # ***** 303 *****
4709 013012 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
4710 013014 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
4711 013016 062700 177777      ADD6:   ADD      #177777, R0 ;CC=1000  RO=177777
4712 013022 101402              BLOS    ADD7
4713 013024 102401              BVS     ADD7
4714 013026 100404              BMI     ADD8
4715
4716
4717
4718
4719 013030      ADD7:
4720 013030 012742 000304      MOV      #304, -(R2)      ;MOVE TO MAILBOX # ***** 304 *****
4721 013034 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
4722 013036 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
4723 013040 000277      ADD8:   SCC
4724 013042 000245      +CLC!CLZ                ;CC=1010  R=0
4725 013044 062700 000001      ADD      #1, R0
4726 013050 102403              BVS     ADD9
4727 013052 103002              BCC     ADD9
4728 013054 100401              BMI     ADD9
4729 013056 001404              BEQ     TST152
4730
4731
4732
4733
4734 013060      ADD9:
4735 013060 012742 000305      MOV      #305, -(R2)      ;MOVE TO MAILBOX # ***** 305 *****
4736 013064 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
4737 013066 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
4738
4739
4740
4741
4742
4743 013070 005212              ;*****
4744 013072 022712 000152      ;TEST 152 TEST ADC INSTRUCTION
4745 013076 001037              ;*****
4746 013100 012700 077777      TST152: INC      (R2)      ;UPDATE TEST NUMBER
4747 013104 000277              CMP      #152, (R2)      ;SEQUENCE ERROR?
4748 013106 000252              BNE     TST153-10        ;BR TO ERROR HALT ON SEQ ERROR
4749 013110 005500              MOV      #077777, R0
4750 013112 101402              SCC
4751 013114 102001              +CLN!CLV                ;CC=0101
4752 013116 100404              ADC      R0              ;CC=1010
4753
4754
4755
4756
4757 013120      ADC1:
4758 013120 012742 000306      MOV      #306, -(R2)      ;MOVE TO MAILBOX # ***** 306 *****
4759 013124 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
4760 013126 000000              HALT                    ;ADC DID NOT SET CC'S CORRECTLY
4761 013130 052700 077777      ADC2:   BIS      #77777, R0
4762 013134 000277              SCC
4763 013136 000244              CLZ

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 732 <====

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 716 <====

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

```

4764 013140 005500          ADC      R0          ;CC=0101  R0=0
4765 013142 101002          BHI      ADC3
4766 013144 102401          BVS      ADC3
4767 013146 100004          BPL      ADC4
4768
4769
4770
4771
4772 013150          ADC3:
4773 013150 012742 000307          MOV      #307, -(R2)          ;MOVE TO MAILBOX # ***** 307 *****
4774 013154 005242          INC      -(R2)              ;SET MSGTYP TO FATAL ERROR
4775 013156 000000          HALT
4776 013160 000277          ADC4:          SCC
4777 013162 000245          +CLZ!CLC          ;CC=1010
4778 013164 005500          ADC      R0          ;CC=0100
4779 013166 102403          BVS      ADC5
4780 013170 103402          BCS      ADC5
4781 013172 100401          BMI      ADC5
4782 013174 001404          BEQ      TST153
4783
4784
4785
4786
4787 013176          ADC5:
4788 013176 012742 000310          MOV      #310, -(R2)          ;MOVE TO MAILBOX # ***** 310 *****
4789 013202 005242          INC      -(R2)              ;SET MSGTYP TO FATAL ERROR
4790 013204 000000          HALT          ;ADC DID NOT SET CC'S CORRECTLY
4791
; OR SEQUENCE ERROR
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 754
<====
<====
<====
<====
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 741
<====
<====
<====
<====

```

 THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
 CMP, AND COM INSTRUCTIONS. EACH OF THESE INSTRUCTIONS GENERATE
 THE C AND V BITS IDENTICALLY. THE CONDITION CODES ARE PRESET,
 THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
 OF CONDITIONAL BRANCH INSTRUCTIONS. THIS PROCEDURE IS REPEATED
 SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
 COMBINATIONS OF THE C AND V BITS.

 TEST 153 TEST NEG INSTRUCTION

4806 013206 005212
 4807 013210 022712 000153
 4808 013214 001042
 4809 013216 012700 000001
 4810 013222 000277
 4811 013224 000251
 4812 013226 005400
 4813 013230 103003
 4814 013232 102402
 4815 013234 001401
 4816 013236 100404
 4817
 4818
 4819
 4820
 4821 013240
 4822 013240 012742 000311
 4823 013244 005242
 4824 013246 000000
 4825 013250 042700 077777
 4826 013254 000257
 4827 013256 000264
 4828 013260 005400
 4829 013262 102003
 4830 013264 103002
 4831 013266 001401
 4832 013270 100404
 4833
 4834
 4835
 4836
 4837 013272
 4838 013272 012742 000312
 4839 013276 005242
 4840 013300 000000
 4841 013302 005000
 4842 013304 000277
 4843 013306 000244
 4844 013310 005400
 4845 013312 102403
 4846 013314 103402
 4847 013316 001001

```

TST153: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #153,(R2)   ;SEQUENCE ERROR?
          BNE     TST154-10    ;BR TO ERROR HALT ON SEQ ERROR
          MOV     #1,R0
          SCC
          +CLN:CLC             ;CC=0110
          NEG     R0           ;CC=1001 RO=177777
          BCC     NEG1
          BVS     NEG1
          BEQ     NEG1
          BMI     NEG2

          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
          ; CONDITIONAL BRANCH INST. AND
          ; REPLACE THE MOVE INSTRUCTION
          ; WHICH FOLLOWS W/ 767
          <=====
          <=====
          <=====
          <=====

NEG1:    MOV     #311,-(R2)    ;MOVE TO MAILBOX # ***** 311 *****
          INC     -(R2)
          HALT
          ;SET MSGTYP TO FATAL ERROR
          ;NEG DID NOT SET CC'S CORRECTLY

NEG2:    BIC     #77777,R0
          CCC
          SEZ
          NEG     R0           ;CC=0100
          BVC     NEG3
          BCC     NEG3
          BEQ     NEG3
          BMI     NEG4

          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
          ; CONDITIONAL BRANCH INST. AND
          ; REPLACE THE MOVE INSTRUCTION
          ; WHICH FOLLOWS W/ 752
          <=====
          <=====
          <=====
          <=====

NEG3:    MOV     #312,-(R2)    ;MOVE TO MAILBOX # ***** 312 *****
          INC     -(R2)
          HALT
          ;SET MSGTYP TO FATAL ERROR
          ;NEG DID NOT SET CC'S CORRECTLY

NEG4:    CLR     R0
          SCC
          CLZ
          NEG     R0           ;CC=1011
          BVS     NEG5
          BCS     NEG5
          BNE     NEG5
  
```


4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949

013452
013452 012742 000316
013456 005242
013460 000000
013462 042700 040000
013466 000277
013470 022700 177777
013474 101402
013476 102401
013500 100004

013502
013502 012742 000317
013506 005242
013510 000000

013512 005212
013514 022712 000155
013520 001010
013522 012700 177777
013526 000257
013530 000265
013532 005100
013534 101002
013536 102401
013540 100004

013542
013542 012742 000320
013546 005242
013550 000000

CMP5:
MOV #316, -(R2)
INC -(R2)
HALT
CMP6:
BIC #40000, R0
SBC
CMP #-1, R0
BLJS CMP7
BVS CMP7
BPL TST155

CMP7:
MOV #317, -(R2)
INC -(R2)
HALT

TST155:
INC (R2)
CMP #155, (R2)
BNE TST156-10
MOV #-1, R0
CCC
+SEC!SEZ
COM R0
BHI COM1
BVS COM1
BPL TST156

COM1:
MOV #320, -(R2)
INC -(R2)
HALT

CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 734
<====
<====
<====
: MOVE TO MAILBOX # ***** 316 *****
: SET MSGTYP TO FATAL ERROR
: CMP DID NOT SET CC'S CORRECTLY
: CC=1111
: CC=0000
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
: CONDITIONAL BRANCH INST. AND
: REPLACE THE MOVE INSTRUCTION
: WHICH FOLLOWS W/ 720
<====
<====
<====
<====
: MOVE TO MAILBOX # ***** 317 *****
: SET MSGTYP TO FATAL ERROR
: CMP DID NOT SET CC'S CORRECTLY
: OR SEQUENCE ERROR
: *****
: TEST 155 TEST COM INSTRUCTION
: *****
: UPDATE TEST NUMBER
: SEQUENCE ERROR?
: BR TO ERROR HALT ON SEQ ERROR
: CC=1010
: CC=0101
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
: CONDITIONAL BRANCH INST. AND
: REPLACE THE MOVE INSTRUCTION
: WHICH FOLLOWS W/ 770
<====
<====
<====
<====
: MOVE TO MAILBOX # ***** 320 *****
: SET MSGTYP TO FATAL ERROR
: 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 156 TEST SUB INSTRUCTION

4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005

```

*****
TEST 156 TEST SUB INSTRUCTION
*****
*ST156: INC (R2) ;UPDATE TEST NUMBER
        CMP #156,(R2) ;SEQUENCE ERROR?
        BNE TST157-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #125252,R0
        CCC ;CC=1010
        +SEN!SEC
        SUB #125252,R0 ;CC=0101 R0=0
        BHI SUB1
        BVS SUB1
        BPL SUB2
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (=====
        ; CONDITIONAL BRANCH INST. AND (=====
        ; REPLACE THE MOVE INSTRUCTION (=====
        ; WHICH FOLLOWS W/ 767 (=====

SUB1: MOV #321,-(R2) ;MOVE TO MAILBOX # ***** 321 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;SUB DID NOT SET CC'S CORRECTLY

SUB2: BIS #100000,R0
      SCC ;CC=1101
      CLV
      SUB #77777,R0 ;CC=0010 R0=1
      BLOS SUB3
      BVC SUB3
      BPL SUB4
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (=====
      ; CONDITIONAL BRANCH INST. AND (=====
      ; REPLACE THE MOVE INSTRUCTION (=====
      ; WHICH FOLLOWS W/ 752 (=====

SUB3: MOV #322,-(R2) ;MOVE TO MAILBOX # ***** 322 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT

SUB4: COM R0 ;R0=177777
      SCC ;CC=11111

SUB5: SUB #100000,R0 ;CC=0000 R0=77777
      BLOS SUB5
      BVS SUB5
      BPL SUB6
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS (=====
      ; CONDITIONAL BRANCH INST. AND (=====

```

013552 005212
013554 022712 000156
013560 001055
013562 012700 125252
013566 000257
013570 000271
013572 162700 125252
013576 101002
013600 102401
013602 100004

013604
013604 012742 000321
013610 005242
013612 000000
013614 052700 100000
013620 000277
013622 000242
013624 162700 077777
013630 101402
013632 102001
013634 100004

013636
013636 012742 000322
013642 005242
013644 000000
013646 005100
013650 000277

013652 162700 100000
013656 101402
013660 102401
013662 100004

5062	014004	012742	000326		MOV	#326, -(R2)	; MOVE TO MAILBOX # ***** 326 *****	
5063	014010	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR	
5064	014012	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY	
5065	014014	000277		SBC4:	SCC		; CC=0111	
5066	014016	000250			CLN			
5067	014020	005600			SBC	RO	; CC=1001 RO=177777	
5068	014022	103003			SBC	SBC5		
5069	014024	102402			BVS	SBC5		
5070	014026	001401			BEQ	SBC5		
5071	014030	100404			BMI	SBC6		
5072							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<=====
5073							CONDITIONAL BRANCH INST. AND	<=====
5074							REPLACE THE MOVE INSTRUCTION	<=====
5075							WHICH FOLLOWS W/ 741	<=====
5076	014032			SBC5:				
5077	014032	012742	000327		MOV	#327, -(R2)	; MOVE TO MAILBOX # ***** 327 *****	
5078	014036	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR	
5079	014040	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY	
5080	014042	042700	077777	SBC6:	BIC	#77777, RO	; RO=100000	
5081	014046	000277			SCC		; CC=1101	
5082	014050	000242			CLV			
5083	014052	005600			SBC	RO	; CC=0010	
5084	014054	101402			BLOS	SBC7		
5085	014056	102001			BVC	SBC7		
5086	014060	100004			BPL	TST160		
5087							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<=====
5088							CONDITIONAL BRANCH INST. AND	<=====
5089							REPLACE THE MOVE INSTRUCTION	<=====
5090							WHICH FOLLOWS W/ 725	<=====
5091	014062			SBC7:				
5092	014062	012742	000330		MOV	#330, -(R2)	; MOVE TO MAILBOX # ***** 330 *****	
5093	014066	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR	
5094	014070	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY	
5095							; OR SEQUENCE ERROR	
5096								

TEST INSTRUCTION

5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152

014072 005212
014074 022712 000160
014100 001053
014102 012700 144000
014106 000257
014110 000266
014112 006100
014114 103003
014116 102402
014120 001401
014122 100404

014124
014124 012742 000331
014130 005242
014132 000000
014134 000277
014136 000243
014140 006100
014142 103003
014144 102002
014146 001401
014150 100004

014152
014152 012742 000332
014156 005242
014160 000000
014162 000277
014164 000250
014166 006100
014170 101402
014172 102401
014174 100004

THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
ROR, ASL AND ASR INSTRUCTIONS. SPECIAL DATA PATTERNS ARE LOADED
AND ROTATED SEVERAL TIMES FOR EACH TEST. THE CONDITION CODES
ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
CHECKED AFTER EACH ROTATION. THE FINAL CHECK IN EACH TEST IS
TO VERIFY THE COMMULATIVE DATA RESULT. THE DATA PATTERNS HAVE
BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.

TEST 160 TEST ROL INSTRUCTION

TST160: INC (R2) ; UPDATE TEST NUMBER
CMP #160, (R2) ; SEQUENCE ERROR?
BNE TST161-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #144000, R0 ; RO=144000
CCC ; CC=0110
+SEZ!SEV
ROL R0 ; CC=1001 RO=110000
BCC ROL1
BVS ROL1
BEQ ROL1
BMI ROL2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 767 <====

ROL1: MOV #331, -(R2) ; MOVE TO MAILBOX # ***** 331 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT
ROL2: SCC ; CC=1100
+CLV!CLC
ROL R0 ; CC=0011 RO=020000
BCC ROL3
BVC ROL3
BEQ ROL3
BPL ROL4

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

ROL3: MOV #332, -(R2) ; MOVE TO MAILBOX # ***** 332 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; ROL DID NOT SET CC'S CORRECTLY
ROL4: SCC ; CC=0111
CLN
ROL R0 ; CC=0000 RO=040001
BLOS ROL5
BVS ROL5
BPL ROL6

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


```

5209                                     ;          WHICH FOLLOWS W/ 754          =====
5210 014320                                ROR3:                                ;
5211 014320 012742 000336                MOV      #336, -(R2)                ; MOVE TO MAILBOX # ***** 336 *****
5212 014324 005242                        INC      -(R2)                    ; SET MSGTYP TO FATAL ERROR
5213 014326 000000                        HALT                                     ; ROR DID NOT SET CC'S CORRECTLY
5214 014330 000277                                ROR4:                                ; CC=1110
5215 014332 000241                        SCC                                     ;
5216 014334 006000                        CLC                                     ;
5217 014336 101403                        ROR      RO                        ; CC=0000  RO=020002
5218 014340 102402                        BLOS    ROR5                       ;
5219 014342 001401                        BVS     ROR5                       ;
5220 014344 100004                        BEQ     ROR5                       ;
5221                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
5222                                     ; CONDITIONAL BRANCH INST. AND           <====
5223                                     ; REPLACE THE MOVE INSTRUCTION         <====
5224                                     ; WHICH FOLLOWS W/ 741                  <====
5225 014346                                ROR5:                                ;
5226 014346 012742 000337                MOV      #337, -(R2)                ; MOVE TO MAILBOX # ***** 337 *****
5227 014352 005242                        INC      -(R2)                    ; SET MSGTYP TO FATAL ERROR
5228 014354 000000                        HALT                                     ; ROR DID NOT SET CC'S CORRECTLY
5229 014356 000257                                ROR6:                                ; CC=0101
5230 014360 000265                        CCC                                     ;
5231 014362 006000                        +SEC!SEZ                             ;
5232 014364 101402                        ROR      RO                        ; CC=1010  RO=110001
5233 014366 102001                        BLOS    ROR7                       ;
5234 014370 100404                        BVC     ROR7                       ;
5235                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
5236                                     ; CONDITIONAL BRANCH INST. AND           <====
5237                                     ; REPLACE THE MOVE INSTRUCTION         <====
5238                                     ; WHICH FOLLOWS W/ 727                  <====
5239 014372                                ROR7:                                ;
5240 014372 012742 000340                MOV      #340, -(R2)                ; MOVE TO MAILBOX # ***** 340 *****
5241 014376 005242                        INC      -(R2)                    ; SET MSGTYP TO FATAL ERROR
5242 014400 000000                        HALT                                     ; ROR DID NOT PRODUCE CORRECT RESULTS
5243                                     ; OR SEQUENCE ERROR
5244
5245 ;*****
5246 ;TEST 162      TEST ASL INSTRUCTION
5247 ;*****
5248 014402 005212                                TST162: INC      (R2)                ; UPDATE TEST NUMBER
5249 014404 022712 000162                CMP      #162, (R2)                ; SEQUENCE ERROR?
5250 014410 001054                        BNE     TST163-10                  ; BR TO ERROR HALT ON SEQ ERROR
5251 014412 012700 144000                MOV      #144000, RO                ; RO=14000
5252 014416 000257                        CCC                                     ; CC=0110
5253 014420 000271                        +SEN!SEC                             ;
5254 014422 006300                        ASL     RO                        ; CC=1001  RO=110000
5255 014424 103003                        BCC     ASL1                       ;
5256 014426 102402                        BVS     ASL1                       ;
5257 014430 001401                        BEQ     ASL1                       ;
5258 014432 100404                        BMI     ASL2                       ;
5259                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
5260                                     ; CONDITIONAL BRANCH INST. AND           <====
5261                                     ; REPLACE THE MOVE INSTRUCTION         <====
5262                                     ; WHICH FOLLOWS W/ 767                  <====
5263 014434                                ASL1:                                ;
5264 014434 012742 000341                MOV      #341, -(R2)                ; MOVE TO MAILBOX # ***** 341 *****

```

```

5265 014440 005242      :INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
5266 014442 000000      :HALT
5267 014444 000277      ASL2:  :SCC
5268 014446 000243      :+CLV!CLC
5269 014450 006300      ASL      RO      ;CC=0011  RO=020000
5270 014452 103003      BCC      ASL3
5271 014454 102002      SVC      ASL3
5272 014456 001401      BEQ      ASL3
5273 014460 100004      BPL      ASL4
5274      :
5275      :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5276      :   CONDITIONAL BRANCH INST. AND <====
5277      :   REPLACE THE MOVE INSTRUCTION <====
5278      :   WHICH FOLLOWS W/ 754 <====
5278 014462      ASL3:
5279 014462 012742 000342      MOV      #342, -(R2) ;MOVE TO MAILBOX # ***** 342 *****
5280 014466 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
5281 014470 000000      :HALT
5282 014472 000277      ASL4:  :SCC
5283 014474 000250      :CLN
5284 014476 006300      ASL      RO      ;CC=0000  RO=040000
5285 014500 101402      BLOS     ASL5
5286 014502 102401      BVS      ASL5
5287 014504 100004      BPL      ASL6
5288      :
5289      :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5290      :   CONDITIONAL BRANCH INST. AND <====
5291      :   REPLACE THE MOVE INSTRUCTION <====
5292      :   WHICH FOLLOWS W/ 742 <====
5292 014506      ASL5:
5293 014506 012742 000343      MOV      #343, -(R2) ;MOVE TO MAILBOX # ***** 343 *****
5294 014512 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
5295 014514 000000      :HALT
5296 014516 000257      ASL6:  :CCC
5297 014520 000265      :+SEZ!SEC
5298 014522 006300      ASL      RO      ;CC=1010  RO=100000
5299 014524 103406      BCS      ASL7
5300 014526 001405      BEQ      ASL7
5301 014530 102004      BVC      ASL7
5302 014532 100003      BPL      ASL7
5303 014534 022700 100000      CMP      #100000, RO
5304 014540 001404      BEQ      TST163
5305      :
5306      :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5307      :   CONDITIONAL BRANCH INST. AND <====
5308      :   REPLACE THE MOVE INSTRUCTION <====
5309      :   WHICH FOLLOWS W/ 724 <====
5309 014542      ASL7:
5310 014542 012742 000344      MOV      #344, -(R2) ;MOVE TO MAILBOX # ***** 344 *****
5311 014546 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
5312 014550 000000      :HALT
5313      :
5314      :   ASL MALFUNCTIONED
5315      :   OR SEQUENCE ERROR
5315      :*****
5315      :TEST 163      TEST ASR INSTRUCTION
5317      :*****
5318 014552 005212      TST163: INC      (R2)      ;UPDATE TEST NUMBER
5319 014554 022712 000163      CMP      #163, (R2) ;SEQUENCE ERROR?
5320 014560 001060      BNE      TST164-10 ;BR TO ERROR HALT ON SEQ ERROR

```

```

5321 014562 012700 100023      MOV      #100023,R0      ;RO=100023
5322 014566 000277              SCC              ;CC=0110
5323 014570 000250              CLN
5324 014572 006200              ASR      R0          ;CC=1001  RP=140011
5325 014574 102403              BVS     ASR1
5326 014576 103002              BCC     ASR1
5327 014600 001401              BEQ     ASR1
5328 014602 100404              BMI     ASR2
5329
5330
5331
5332
5333 014604              ASR1:
5334 014604 012742 000345      MOV      #345,-(R2)    ;MOVE TO MAILBOX # ***** 345 *****
5335 014610 005242              INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
5336 014612 000000              HALT
5337 014614 042700 100000      ASR2:  BIC      #100000,R0 ;ASR DID NOT SET CC'S CORRECTLY
5338 014620 000277              SCC
5339 014622 000243              +CLV!CLC           ;RO=40011
5340 014624 006200              ASR      R0          ;CC=1100
5341 014626 102003              BVC     ASR3         ;CC=0011  RO=020004
5342 014630 103002              BCC     ASR3
5343 014632 001401              BEQ     ASR3
5344 014634 100004              BPL     ASR4
5345
5346
5347
5348
5349 014636              ASR3:
5350 014636 012742 000346      MOV      #346,-(R2)    ;MOVE TO MAILBOX # ***** 346 *****
5351 014642 005242              INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
5352 014644 000000              HALT
5353 014646 000277              ASR4:  SCC
5354
5355 014650 006200              ASR      R0          ;CC=1111
5356 014652 101403              BLOS    ASR5         ;CC=0000  RO=010002
5357 014654 102402              BVS     ASR5
5358 014656 001401              BEQ     ASR5
5359 014660 100004              BPL     ASR6
5360
5361
5362
5363
5364 014662              ASR5:
5365 014662 012742 000347      MOV      #347,-(R2)    ;MOVE TO MAILBOX # ***** 347 *****
5366 014666 005242              INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
5367 014670 000000              HALT
5368 014672 052700 100000      ASR6:  BIS      #100000,R0 ;ASR DID NOT SET CC'S CORRECTLY
5369 014676 000257              CCC
5370 014700 000265              +SEZ!SEC
5371 014702 006200              ASR      R0          ;RO=110002
5372 014704 101406              BLOS    ASR7         ;CC=0101
5373 014706 102005              BVC     ASR7
5374 014710 100004              BPL     ASR7
5375 014712 001403              BEQ     ASR7
5376 014714 022700 144001      CMP      #144001,R0   ;CHECK RESULT OF ASR'S

```

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 767 <====

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 752 <====

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 740 <====

5377 014720 001404
5378
5379
5380
5381
5382 014722
5383 014722 012742 000350
5384 014726 005242
5385 014730 000000
5386
5387
5388

BEG TST164

ASR7:

MOV #350, -(R2)
INC -(R2)
HALT

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 720 <====
: MOVE TO MAILBOX # ***** 350 *****
: SET MSGTYP TO FATAL ERROR
: ASR DID NOT FUNCTION CORRECTLY
: OR SEQUENCE ERROR

5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416 014732 005212
5417 014734 022712 000164
5418 014740 001062
5419 014742 012700 015332
5420 014746 012704 015370
5421 014752 012767 000015 000142
5422 014760 012067 000110
5423 014764 012401
5424 014766 012767 177777 000074
5425 014774 012703 000016
5426 015000 005267 000064
5427 015004 032701 100000
5428 015010 013705 177776
5429 015014 042705 177773
5430 015020 000165 015024
5431 015024 000167 000020
5432 015030 012767 015124 000042
5433 015036 012767 015106 000040
5434 015044 000167 000014
5435 015050 012767 015106 000022
5436 015056 012767 015124 000020
5437 015064 006101
5438
5439 015066 012737
5440 015070 000000
5441 015072 177776
5442 015074 000000
5443 015076 000137
5444 015100 000000

```
*****
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 164 TEST THE BRANCH ROM
*****
TST164: INC (R2) ;UPDATE TEST NUMBER
CMP #164,(R2) ;SEQUENCE ERROR?
BNE ER ;BR TO ERROR HALT ON SEQ ERROR
SETUP: MOV #BRTAB,R0 ;INITIALIZE BRANCH TABLE POINTER
MOV #YNTAB,R4 ;INITIALIZE YES/NO BRANCH MAP POINTER
MOV #15,BRCT ;INITIALIZE BRANCH TABLE COUNT
SETBR: MOV (R0)+,BRH ;GET NEXT BRANCH INST.
MOV (R4)+,R1 ;GET NEXT BRANCH MAP
MOV #-1,CC ;INITIALIZE CONDITION CODE VALUE
MOV #16,R3 ;INITIALIZE CONDITION CODE COUNT
SETCC: INC CC ;SET FOR NEXT CC VALUE
BIT #100000,R1 ;SEE IF SHOULD BR W/ THESE CC'S
MOV @#177776,R5 ;SIMULATE A JNE
BIC #177773,R5 ; (JUMP NOT EQUAL)
JMP .+4(R5) ; TO SET2BR
JMP SET2BR
5432 015030 012767 015124 000042 MOV #CONT,NBR ;SET TO CONTINUE IF NO BRANCH
5433 015036 012767 015106 000040 MOV #ER,YBR ;SET TO REPORT ERROR IF BRANCH
5434 015044 000167 000014 JMP AROUND ;GO AROUNDND OPPOSITE CONDITION
5435 015050 012767 015106 000022 SET2BR: MOV #ER,NBR ;SET TO REPORT ERROR IF NO BRANCH
5436 015056 012767 015124 000020 MOV #CONT,YBR ;SET TO CONTINUE IF BRANCH
5437 015064 006101 AROUND: ROL R1 ;UPDATE BIT MAP
5438
5439 015066 012737 CC: MOV (PC)+,@(PC)+ ;SET CONDITION CODE
5440 015070 000000 ;NEW CC VALUE GOES HERE
5441 015072 177776 BRH: 0
5442 015074 000000 ;BRANCH INST. GOES HERE
5443 015076 000137 JMP @ (PC)+ ;THIS JUMP IF NO BRANCH
5444 015100 000000 NBR: 0 ;WHERE TO GO IF NO BRANCH OCCURS
```

MAC 111 27 732
ES- AM BRANCH ROM

015102	000137
015104	000000
015106	012702
015108	012742
015110	005242
015112	000000
015114	000000
015116	005303
015118	013705
015120	042705
015122	000165
015124	000167
015126	005367
015128	013705
015130	042705
015132	000165
015134	000167

000304
000351

```

YBR: 0
BR: 0
MOV #STESTN,R2
MOV #351,-(R2)
INC -(R2)
HALT
BRCT: 0
COCT: DEC R3
MOV #177776,R5
BIC #177773,R5
JMP .+4(R5)
SETCC
BRCT
MOV #177776,R5
BIC #177773,R5
JMP .+4(R5)
JMP SETBR
  
```

```

: THIS JUMP IF BRANCH OCCURS
: WHERE TO GO IF BRANCH OCCURS
: RESTORE POINTER
: MOVE TO MAILBOX # ***** 351 *****
: SET MSGTYP TO FATAL ERROR
:
: CC'S DONE?
: SIMULATE A JNE
: (JUMP NOT EQUAL)
: TO SETCC
:
: BR'S DONE?
: SIMULATE A JNE
: (JUMP NOT EQUAL)
: TO SETBR
  
```

```

5462
5463
5464
5465 015172 005212
5466 015174 022712 000165
5467 015200 001037
5468 015202 005237 000306
5469 015206 105267 000076
5470 015212 001020
5471 015214 132767 000040 163077
5472 015222 001014
5473 015224 023727 000042 015264
5474 015232 001410
5475 015234 012700 015312
5476 015240 105737 177564
5477 015244 100375
5478 015246 112037 177566
5479 015252 001372
5480 015254 013700 000042
5481 015260 001405
5482 015262 000005
5483 015264 004710
5484 015266 000240
5485 015270 000240
5486 015272 000240
5487 015274 000167 163222
5488 015300
5489 015300 012742 000352
5490 015304 005242
5491 015306 000000
5492 015310 177777
5493 015312 047105 020104 043117
5494 015320 042040 045507 040501
5495 015326 005015 000

```

```

*****
TEST 165      END OF PASS SEQUENCE
*****
TEST165: INC      (R2)          ;UPDATE TEST NUMBER
            CMP      #165,(R2) ;SEQUENCE ERROR?
            BNE     EOP1       ;BR TO ERROR HALT ON SEQ EPCR
            INC     #SPASS
            INCB    PASSPT     ;SHOULD PRINT THIS PASS?
            BNE     ACT        ;NO
            BITB   #40,SEVM    ;WILL APT ALLOW PRINTING?
            BNE     ACT        ;NO
            CMP    #42,#SENDAD ;UNDER ACT AUTO ACCEPT?
            BEQ    ACT        ;IF SO SKIP PRINTOUT
            MOV    #MSG,RO     ;GET MSG ADDR.
            TSTB  #TPS        ;TTY READY
            BPL   WAIT        ;NO WAIT
            MOVB  (RO)+,#TPB  ;PRINT CHARACTER
            BNE   WAIT        ;NEXT IF NOT DONE.
            ACT:  MOV    #42,RO ;CHECK ACT
            BEQ   GOAGIN      ;KEEP GOING
            RESET
            JSR   PC,(RO)     ;ACT HOOKS
            NOP
            NOP
            NOP
            GOAGIN: JMP    RESTRT ;DO NEXT PASS
            EOP1:  MOV    #352,-(R2) ;MOVE TO MAILBOX # ***** 352 *****
            INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
            HALT  ;SEQUENCE ERROR
            PASSPT: -1
            MSG:  .ASCIZ .END OF DGKAA.<15><12>

```

```

5498      015332      .EVEN
5499      015332      000402      BRTAB:  BR      .+6
5500      015334      001002      BNE      .+6
5501      015336      001402      BEQ      .+6
5502      015340      002002      BGE      .+6
5503      015342      002402      BLT      .+6
5504      015344      003002      BGT      .+6
5505      015346      003402      BLE      .+6
5506      015350      100002      BPL      .+6
5507      015352      100402      BMI      .+6
5508      015354      101002      BHI      .+6
5509      015356      101402      BLOS     .+6
5510      015360      102002      BVC      .+6
5511      015362      102402      BVS      .+6
5512      015364      103002      BCC      .+6
5513      015366      103402      BCS      .+6
5514      000002      .RADIX  2
5515      015370      177777      YNTAB:  11111111111111111111
5516      015372      170360      1111000011110000
5517      015374      007417      0000111100001111
5518      015376      146063      1100110000110011
5519      015400      031714      0011001111001100
5520      015402      140060      1100000000110000
5521      015404      037717      0011111111001111
5522
5523      015406      177400      1111111100000000
5524      015410      000377      0000000011111111
5525      015412      120240      1010000010100000
5526      015414      057537      0101111101011111
5527      015416      146314      1100110011001100
5528      015420      031463      0011001100110011
5529      015422      125252      1010101010101010
5530      015424      052525      0101010101010101
5531
5532      000010      .RADIX  8
5533      015426      012737      015436  000024  PWRDN:  MOV      #PWRUP, @#24      ; SET UP FOR A POWER UP
5534      015434      000000      HALT
5535
5536      015436      012737      015426  000024  PWRUP:  MOV      #PWRDN, @#24      ; SET UP FOR A POWER FAIL
5537      015444      012706      000500      MOV      #STBOT, R6      ; SET UP STACK POINTER
5538      015450      132767      000040      162643      BITB     #40, $ENVM      ; SHOULD PRINT?
5539      015456      001010      BNE      PWR2           ; IF NOT: BR
5540      015460      012700      015504      MOV      #PFMES, R0      ; GET POWER FAIL MESSG.
5541      015464      105737      177564      WATE:   TSTB     @#TPS      ; TTY READY?
5542      015470      100375      BPL      WATE           ; IF NOT: BR
5543      015472      112037      177566      MOVB    (R0)+, @#TPB     ; PRINT NEXT CHAR.
5544      015476      001372      BNE      WATE           ; IF NOT DONE: BR
5545      015500      000137      000500      PWR2:   JMP      @#START      ; START PROGRAM AGAIN
5546
5547      015504      006412      047520  042527  PFMES:  .ASCIZ  <12><15>.POWER FAILURE.<12><15>
5548      015512      020122      040506  046111
5549      015520      051125      005105  000015
5550
5551      000001      .END

```

RBASE =	000000	424			
RCDM1 =	000000	424			
RCDM2 =	000000	424			
RCPUOP =	000000	424	439		
RCT =	015254	5470	5472	5474	5490#
RCC1 =	013120	4750	4751	4757#	
RCC2 =	013130	4752	4761#		
RCC3 =	013150	4765	4766	4772#	
RCC4 =	013160	4767	4776#		
RCC5 =	013176	4779	4780	4781	4787#
ROOM0 =	000000	424			
ROOM1 =	000000	424			
ROOM10 =	000000	424			
ROOM11 =	000000	424			
ROOM12 =	000000	424			
ROOM13 =	000000	424			
ROOM14 =	000000	424			
ROOM15 =	000000	424			
ROOM2 =	000000	424			
ROOM3 =	000000	424			
ROOM4 =	000000	424			
ROOM5 =	000000	424			
ROOM6 =	000000	424			
ROOM7 =	000000	424			
ROOM8 =	000000	424			
ROOM9 =	000000	424			
ROD1 =	012734	4672	4673	4679#	
ROD2 =	012744	4674	4683#		
ROD3 =	012760	4686	4687	4693#	
ROD4 =	012770	4688	4697#		
ROD5 =	013006	4700	4701	4707#	
ROD6 =	013016	4702	4711#		
ROD7 =	013030	4712	4713	4719#	
ROD8 =	013040	4714	4723#		
ROD9 =	013060	4726	4727	4728	4734#
RDEVCT =	000000	424	430		
RDEVN =	000000	424			
REMY =	000000	424	435		
RENVM =	000000	424	436		
RFATAL =	000000	424	427		
RMAOR1 =	000000	424			
RMAOR2 =	000000	424			
RMAOR3 =	000000	424			
RMAOR4 =	000000	424			
RMAAS1 =	000000	424			
RMAAS2 =	000000	424			
RMAAS3 =	000000	424			
RMAAS4 =	000000	424			
RMSGAD =	000000	424	432		
RMSGLG =	000000	424	433		
RMSGTY =	000000	424	426		
RMTYP1 =	000000	424			
RMTYP2 =	000000	424			
RMTYP3 =	000000	424			
RMTYP4 =	000000	424			
RPASS =	000000	424	429		

APRIOR=	000000	424				
AROUND	015064	5434	5437#			
ASL1	014434	5255	5256	5257	5263#	
ASL2	014444	5258	5267#			
ASL3	014462	5270	5271	5272	5278#	
ASL4	014472	5273	5282#			
ASL5	014506	5285	5286	5292#		
ASL6	014516	5287	5296#			
ASL7	014542	5299	5300	5301	5302	5309#
ASR1	014604	5325	5326	5327	5333#	
ASR2	014614	5328	5337#			
ASR3	014636	5341	5342	5343	5349#	
ASR4	014646	5344	5353#			
ASR5	014662	5356	5357	5358	5364#	
ASR6	014672	5359	5368#			
ASR7	014722	5372	5373	5374	5375	5382#
ASUREG=	000000	424	437			
ATESTN=	000000	424	429			
AUNIT =	000000	424	431			
AUSUR =	000000	424	438			
AVECT1=	000000	424				
AVECT2=	000000	424				
BIC1	012064	4345	4346	4352#		
BIC2	012074	4347	4356#			
BIC3	012112	4359	4360	4366#		
BISI	012154	4382	4383	4384	4390#	
BIS2	012164	4385	4394#			
BIS3	012204	4397	4398	4399	4405#	
BIT1	011774	4307	4308	4314#		
BIT2	012004	4309	4319#			
BIT3	012022	4322	4323	4329#		
BACT	015122	5421#	5451#	5457#		
BRC1	002350	1228	1234#			
BRC2	002360	1229	1239#			
BRC3	002370	1241	1247#			
BRH	015074	5422#	5442#			
BRN1	002230	1134	1140#			
BRN2	002240	1135	1145#			
BRN3	002250	1147	1153#			
BRTA8	015332	5419	5498#			
BRV1	002300	1181	1187#			
BRV2	002310	1182	1192#			
BRV3	002320	1194	1200#			
BRZ1	002160	1087	1093#			
BRZ2	002170	1088	1098#			
BRZ3	002200	1100	1106#			
BR1	000572	504	510#			
BR2	000602	505	514#			
BR3	000614	515	523#			
BR4	000622	524	530#			
BRS	000632	525	534#			
CC	015070	5424#	5426#	5440#		
CLR1	012522	4563	4564	4565	4571#	
CHP1	013364	4869	4870	4876#		
CHP2	013374	4871	4880#			
CHP3	013416	4884	4885	4891#		

3174	3175	3176	3177	3178	3211*	3212	3213	3214	3215	3216	3243*	3244*
3245	3246	3247	3248	3273*	3275*	3278	3288*	3290*	3293	3320*	3321*	3323*
3337*	3338*	3351*	3352*	3353*	3355*	3383*	3384*	3386*	3390*	3391*	3401*	3402*
3404*	3408*	3419*	3419*	3420*	3422*	3426*	3427*	3511*	3513*	3517	3546*	3549*
3553	3640*	3641*	3650	3675*	3676*	3703*	3704*	3714*	3767*	3768*	3778	3805*
3807*	3817	3848*	3849*	3850	3881*	3882*	3883	3936*	3937	3938	3956*	3958
3971*	3972	3973	3991*	3993	4004*	4006	4017*	4019	4031*	4033	4074*	4078
4094	4126*	4127	4142*	4143	4149	4165	4176*	4177	4228*	4229*	4235	4268*
4283*	4303*	4306	4321	4341*	4344*	4358*	4378*	4381*	4396*	4429*	4432*	4444*
4447*	4463*	4483*	4485*	4500*	4514*	4526*	4529*	4562*	4585	4598*	4601	4621*
4624*	4638*	4669*	4671*	4685*	4699*	4711*	4725*	4746*	4749*	4761*	4764*	4778*
4809*	4812*	4825*	4828*	4841*	4844*	4865*	4868	4880*	4883	4895*	4898	4911*
4913	4933*	4936*	4967*	4970*	4982*	4985*	4997*	5000*	5014*	5034*	5037*	5052*
5067*	5080*	5083*	5114*	5117*	5132*	5147*	5161*	5165	5183*	5186*	5201*	5216*
5231*	5251*	5254*	5269*	5284*	5298*	5303	5321*	5324*	5337*	5340*	5355*	5368*
5371*	5376	5419*	5422	5475*	5478	5480*	5483	5540*	5543			
737*	739*	4076*	4077*	4078*	4088	4105*	4109	4124*	4127*	4131	4143*	4147
4160*	4163	4177*	4181	4192*	4200	5423*	5427	5437*				
482*	493*	500*	501	511*	512*	520*	521*	531*	532*	540*	541*	562*
563	573*	574*	581*	582	592*	593*	600*	601	611*	612*	619*	620
629*	630*	661*	662	672*	673*	680*	681	690*	691*	698*	699	708*
709*	716*	717	726*	727*	734*	735	746*	747*	754*	755	757*	759*
766*	767*	768*	770*	774*	775	786*	787*	794*	795	806*	807*	814*
815	826*	827*	834*	835	846*	847*	870*	871	881*	882*	889*	890
899*	900*	907*	908	917*	918*	925*	926	935*	936*	958*	959	970*
971*	978*	979	989*	990*	999*	1000*	1007*	1008	1019*	1020*	1027*	1028
1039*	1040*	1047*	1048	1059*	1060*	1081*	1082	1094*	1095*	1107*	1108*	1128*
1129	1141*	1142*	1154*	1155*	1175*	1176	1188*	1189*	1201*	1202*	1222*	1223
1235*	1236*	1248*	1249*	1291*	1292	1300*	1301*	1310*	1311*	1319*	1320*	1337*
1338	1347*	1348*	1364*	1365*	1378*	1379	1387*	1388*	1399*	1400*	1415*	1416
1425*	1426*	1438*	1439*	1453*	1454	1465*	1466*	1479*	1480*	1497*	1498	1510*
1511*	1525*	1526*	1543*	1544	1557*	1558*	1573*	1574*	1591*	1592	1605*	1606*
1620*	1621*	1633*	1634	1648*	1649*	1665*	1666*	1687*	1688	1700*	1701*	1714*
1715*	1734*	1735	1749*	1750*	1767*	1768*	1788*	1789	1802*	1803*	1819*	1820*
1827*	1828	1839*	1840*	1855*	1856*	1878*	1879	1890*	1891*	1904*	1905*	1920*
1921	1932*	1933*	1944*	1945*	1961*	1962	1973*	1974*	1985*	1986*	2001*	2002
2011*	2012*	2024*	2025*	2041*	2042	2057*	2058*	2073*	2074	2089*	2090*	2105*
2106	2122*	2123*	2137*	2138	2154*	2155*	2171*	2172*	2186*	2187	2203*	2204*
2213*	2214*	2229*	2230	2246*	2247*	2255*	2256*	2271*	2272*	2281*	2282*	2296*
2297	2315*	2316*	2325*	2326*	2342*	2343	2362*	2363*	2377*	2378*	2387*	2388*
2402*	2403	2419*	2420*	2428*	2429*	2443*	2444	2462*	2463*	2472*	2473*	2487*
2488	2505*	2506*	2514*	2515*	2529*	2530	2547*	2548*	2556*	2557*	2570*	2571
2583*	2584*	2597*	2598	2610*	2611*	2626*	2627	2636*	2637*	2649*	2650*	2660*
2661*	2675*	2676*	2686*	2687*	2700*	2701	2712*	2713*	2721*	2722*	2731*	2732*
2743*	2744*	2753*	2754*	2768*	2769	2782*	2783*	2796*	2797	2810*	2811*	2825*
2826	2839*	2840*	2858*	2859	2874*	2875*	2884*	2885*	2899*	2900	2914*	2915*
2929*	2930	2942*	2943*	2952*	2953*	2969*	2970	2982*	2983*	2991*	2992*	3005*
3006	3020*	3021*	3030*	3031*	3046*	3047	3058*	3059*	3073*	3074	3085*	3086*
3100*	3101	3112*	3113*	3132*	3133	3147*	3148*	3170*	3171	3185*	3186*	3208*
3209	3222*	3223*	3240*	3241	3254*	3255*	3270*	3271	3285*	3286*	3300*	3301*
3317*	3318	3333*	3334*	3348*	3349*	3365*	3366*	3380*	3381	3398*	3399*	3415*
3416*	3434*	3435*	3449*	3450	3463*	3464*	3477*	3478*	3491*	3492*	3507*	3508
3524*	3525*	3542*	3543	3560*	3561*	3576*	3577	3590*	3591*	3605*	3606	3620*
3621*	3637*	3638	3647*	3648*	3656*	3657*	3671*	3672	3683*	3684*	3699*	3700
3711*	3712*	3720*	3721*	3736*	3737	3747*	3748*	3763*	3764	3775*	3776*	3784*
3785*	3802*	3803	3814*	3815*	3824*	3825*	3844*	3845	3857*	3858*	3877*	3878

R1 =:000001

R2 =:000002

SNM818	004306	2148	2157#		
SNM81C	004330	2162	2163	2164	2170#
SNM82A	004452	2238	2239	2245#	
SNM82B	004462	2240	2249#		
SNM82C	004476	2250	2258#		
SNM82D	004516	2262	2263	2264	2270#
SNM82E	004526	2265	2274#		
SNM83A	004670	2354	2355	2361#	
SNM83B	004700	2356	2365#		
SNM83C	004716	2368	2369	2370	2376#
SNM83D	004726	2371	2380#		
SNM0A	004132	2048	2049	2050	2056#
SNM1A	004234	2113	2114	2115	2121#
SNM2A	004372	2194	2195	2196	2202#
SNM2B	004402	2197	2206#		
SNM3A	004602	2306	2307	2308	2314#
SNM3B	004612	2309	2318#		
SNM4A	004776	2411	2412	2418#	
SNM4B	005006	2413	2422#		
SNM5A	005060	2454	2455	2461#	
SNM5B	005070	2456	2465#		
SNM6A	005144	2497	2498	2504#	
SNM6B	005154	2499	2508#		
SNM7A	005226	2539	2540	2546#	
SNM7B	005236	2541	2550#		
SOPA	004050	2006	2014#		
SOPB	004070	2003	2016	2023#	
SOPB0A	002562	1382	1390#		
SOPB0B	002572	1391	1398#		
SOPB1A	002704	1460	1468#		
SOPB1B	002720	1469	1471	1478#	
SOPB1C	002764	1505	1513#		
SOPB1D	003002	1517	1524#		
SOPB2A	003136	1600	1608#		
SOPB2B	003154	1611	1619#		
SOPB2C	003224	1643	1651#		
SOPB2D	003246	1656	1664#		
SOPB3A	003376	1744	1752#		
SOPB3B	003422	1755	1757	1766#	
SOPB3C	003470	1797	1805#		
SOPB7C	003512	1811	1818#		
SOPX	004034	2005#	2014*	2015*	2028
SOPX4D	004100	2017*	2028#		
SOPZ4	003050	1552	1560#		
SOP0A	002424	1295	1303#		
SOP0B	002442	1305	1313#		
SOP0C	002504	1342	1350#		
SOP0D	002526	1352	1355	1363#	
SOP1A	002630	1420	1428#		
SOP1B	002642	1429	1437#		
SOP2B	003070	1563	1572#		
SOP3A	003312	1695	1703#		
SOP3B	003326	1706	1713#		
SOP4A	003554	1834	1842#		
SOP4B	003574	1845	1854#		
SOP5A	003636	1885	1893#		

TST131	010270	3738	3742	3763#
TST132	010350	3765	3779	3802#
TST133	010436	3818	3844#	
TST134	010506	3851	3877#	
TST135	010560	3884	3932#	
TST136	011124	4037	4067#	
TST137	011602	4201	4223#	
TST14	001300	774#		
TST140	011656	4225	4236	4263#
TST141	011742	4265	4286	4300#
TST142	012032	4302	4324	4338#
TST143	012122	4340	4361	4375#
TST144	012214	4377	4400	4426#
TST145	012332	4428	4466	4480#
TST146	012474	4482	4533	4557#
TST147	012532	4559	4566	4580#
TST15	001334	776	781	794#
TST150	012616	4582	4604	4618#
TST151	012704	4620	4642	4666#
TST152	013070	4668	4729	4743#
TST153	013206	4745	4782	4806#
TST154	013332	4808	4848	4862#
TST155	013512	4864	4916	4930#
TST156	013552	4932	4939	4964#
TST157	013724	4966	5018	5031#
TST16	001370	796	801	814#
TST160	014072	5033	5086	5111#
TST161	014240	5113	5166	5180#
TST162	014402	5182	5234	5248#
TST163	014552	5250	5304	5318#
TST164	014732	5320	5377	5416#
TST165	015172	5465#		
TST17	001424	816	821	834#
TST2	000644	502	535	562#
TST20	001460	836	841	870#
TST21	001520	872	876	889#
TST22	001556	891	894	907#
TST23	001614	909	912	925#
TST24	001652	927	930	958#
TST25	001716	960	965	978#
TST26	001774	980	984	994
TST27	002034	1009	1014	1027#
TST3	000700	564	568	581#
TST30	002100	1029	1034	1047#
TST31	002140	1049	1054	1081#
TST32	002210	1083	1101	1128#
TST33	002260	1130	1148	1175#
TST34	002330	1177	1195	1222#
TST35	002400	1224	1242	1291#
TST36	002456	1293	1314	1337#
TST37	002536	1339	1358	1378#
TST4	000736	583	587	600#
TST40	002602	1380	1393	1415#
TST41	002652	1417	1432	1453#
TST42	002730	1455	1473	1497#
TST43	003012	1499	1519	1543#

1007#

2636	2637	2649	2650	2660	2661	2675	2676	2686	2687	2712	2713	2721
2722	2731	2732	2743	2744	2753	2754	2782	2783	2810	2911	2839	2840
2874	2875	2884	2885	2914	2915	2942	2943	2952	2953	2982	2983	2991
2992	3020	3021	3030	3031	3058	3059	3085	3086	3112	3113	3147	3148
3185	3186	3222	3223	3254	3255	3285	3286	3300	3301	3333	3334	3349
3349	3365	3366	3398	3399	3415	3416	3434	3435	3453	3464	3477	3478
3491	3492	3524	3525	3560	3561	3590	3591	3620	3621	3647	3648	3656
3657	3683	3684	3711	3712	3720	3721	3747	3748	3775	3776	3784	3785
3814	3815	3824	3825	3857	3858	3890	3891	3944	3945	3953	3954	3967
3968	3979	3980	3988	3989	4001	4002	4014	4015	4028	4029	4042	4043
4082	4083	4101	4102	4120	4121	4138	4139	4156	4157	4172	4173	4188
4189	4207	4208	4232	4233	4241	4242	4277	4278	4292	4293	4315	4316
4330	4331	4353	4354	4367	4368	4391	4392	4406	4407	4441	4442	4457
4458	4472	4473	4495	4496	4509	4510	4523	4524	4539	4540	4572	4573
4595	4596	4610	4611	4633	4634	4648	4649	4680	4681	4694	4695	4708
4709	4720	4721	4735	4736	4758	4759	4773	4774	4788	4789	4822	4823
4838	4839	4854	4855	4877	4878	4892	4893	4908	4909	4922	4923	4945
4946	4979	4980	4994	4995	5009	5010	5024	5025	5047	5048	5062	5063
5077	5078	5092	5093	5127	5128	5142	5143	5156	5157	5172	5173	5196
5197	5211	5212	5226	5227	5240	5241	5264	5265	5279	5280	5293	5294
5310	5311	5334	5335	5350	5351	5365	5366	5383	5384	5448	5449	5489
5490	487	495*										
\$ERROR=	000302											
\$ETABL	000320	434										
\$ETEND	000330	446	469									
\$FATAL	000302	427	487									
\$HIBTS	000330	464										
\$MAIL	000300	425	465	469								
\$MADR	000332	465										
\$MSGAD	000314	432										
\$MSGLG	000316	433										
\$MSGTY	000300	426	496*									
\$PASS	000306	429	490*	5468*								
\$PASTM	000336	467										
\$SVPC =	000400	414	419									
\$SWR =	000000	398										
\$SWREG	000322	437										
\$TESTM	000304	428	482	488	493	766	770	5447				
\$TN =	000166	398	497	503	535	559	565	568	578	584	587	597
		616	622	624	658	664	667	677	683	685	695	701
		719	721	731	737	741	751	757	771	777	781	791
		811	817	821	831	837	841	867	873	876	886	892
		910	912	922	928	930	955	961	965	975	981	984
		1010	1014	1024	1030	1034	1044	1050	1054	1078	1084	1101
		1148	1172	1178	1195	1219	1225	1242	1288	1294	1314	1334
		1375	1381	1393	1412	1418	1432	1450	1456	1473	1494	1500
		1546	1567	1588	1594	1614	1630	1636	1659	1684	1690	1708
		1761	1785	1791	1813	1824	1830	1849	1875	1881	1898	1917
		1958	1964	1979	1998	2004	2018	2038	2044	2051	2070	2076
		2108	2116	2134	2140	2165	2183	2189	2208	2226	2232	2276
		2320	2339	2345	2382	2399	2405	2423	2440	2446	2467	2484
		2526	2532	2551	2567	2573	2578	2594	2600	2605	2623	2629
		2703	2748	2765	2771	2777	2793	2799	2805	2822	2828	2834
		2879	2896	2902	2909	2926	2932	2947	2966	2972	2986	3002
		3043	3049	3053	3070	3076	3080	3097	3103	3107	3129	3135
		3173	3179	3205	3211	3217	3237	3243	3249	3267	3273	3294
												3314
												3320

\$ERROR= 000302
 \$ETABL 000320
 \$ETEND 000330
 \$FATAL 000302
 \$HIBTS 000330
 \$MAIL 000300
 \$MADR 000332
 \$MSGAD 000314
 \$MSGLG 000316
 \$MSGTY 000300
 \$PASS 000306
 \$PASTM 000336
 \$SVPC = 000400
 \$SWR = 000000
 \$SWREG 000322
 \$TESTM 000304
 \$TN = 000166

CROSS REFERENCE TABLE -- USER SYMBOLS

STSTM 000334
STSTM= 000304
SUNIT 000312
SUNIT 000340
SUSAR 000324
SX = 015202

3359	3377	3383	3428	3446	3452	3485	3504	3510	3518	3539	3545	3554
3573	3579	3584	3602	3608	3614	3634	3640	3651	3668	3674	3678	3696
3702	3715	3733	3739	3742	3760	3766	3779	3799	3805	3818	3841	3847
3851	3874	3880	3884	3929	3935	4037	4064	4070	4201	4220	4226	4236
4260	4266	4286	4297	4303	4324	4335	4341	4361	4372	4378	4400	4423
4429	4466	4477	4483	4533	4554	4560	4566	4577	4593	4604	4615	4621
4642	4663	4669	4729	4740	4746	4782	4803	4809	4848	4859	4865	4916
4927	4933	4939	4961	4967	5018	5028	5034	5086	5108	5114	5166	5177
5183	5234	5245	5251	5304	5315	5321	5377	5413	5419	5462	5468	
466												
488	494*											
431												
468												
438												
503	518	538	565	571	584	590	603	609	622	627	664	670
683	688	701	706	719	724	737	744	757	777	784	797	804
817	824	837	844	873	879	892	897	910	915	928	933	961
968	981	987	997	1010	1017	1030	1037	1050	1057	1084	1091	1104
1131	1138	1151	1178	1185	1198	1225	1232	1245	1294	1298	1308	1317
1340	1345	1361	1381	1385	1396	1418	1423	1435	1456	1463	1476	1500
1508	1522	1546	1555	1570	1594	1603	1617	1636	1646	1662	1690	1698
1711	1737	1747	1764	1791	1800	1816	1830	1837	1852	1881	1888	1901
1923	1930	1941	1964	1971	1982	2004	2009	2021	2044	2054	2076	2086
2108	2119	2140	2151	2168	2189	2200	2211	2232	2243	2253	2268	2279
2299	2312	2323	2345	2359	2374	2385	2405	2416	2426	2446	2459	2470
2490	2502	2512	2532	2544	2554	2573	2581	2600	2608	2629	2634	2647
2658	2673	2684	2703	2710	2719	2729	2741	2751	2771	2780	2799	2808
2828	2837	2861	2872	2882	2902	2912	2932	2940	2950	2972	2980	2989
3008	3018	3028	3049	3056	3076	3083	3103	3110	3135	3144	3173	3182
3211	3220	3243	3252	3273	3282	3297	3320	3330	3345	3362	3383	3395
3412	3431	3452	3460	3474	3488	3510	3521	3545	3557	3579	3587	3608
3617	3640	3645	3654	3674	3681	3702	3709	3718	3739	3745	3766	3773
3782	3805	3812	3821	3847	3854	3880	3887	3935	3942	3951	3965	3977
3986	3999	4012	4026	4040	4070	4098	4117	4135	4153	4169	4185	4204
4226	4239	4266	4274	4289	4303	4312	4327	4341	4350	4364	4378	4388
4403	4429	4438	4454	4469	4483	4492	4506	4520	4536	4560	4569	4583
4592	4607	4621	4630	4645	4669	4677	4691	4705	4717	4732	4746	4755
4770	4785	4809	4819	4835	4851	4865	4874	4889	4905	4919	4933	4942
4967	4976	4991	5006	5021	5034	5044	5059	5074	5089	5114	5124	5139
5153	5169	5183	5193	5208	5223	5237	5251	5261	5276	5290	5307	5321
5331	5347	5362	5380	5419	5468							
518	538	571	590	609	627	670	688	706	724	744	784	804
824	844	879	897	915	933	968	987	997	1017	1037	1057	1091
1104	1138	1151	1185	1198	1232	1245	1298	1308	1317	1345	1361	1385
1396	1423	1435	1463	1476	1508	1522	1555	1570	1603	1617	1646	1662
1698	1711	1747	1764	1800	1816	1837	1852	1888	1901	1930	1941	1971
1982	2009	2021	2054	2086	2119	2151	2168	2200	2211	2243	2253	2268
2279	2312	2323	2359	2374	2385	2416	2426	2459	2470	2502	2512	2544
2554	2581	2608	2634	2647	2658	2673	2684	2710	2719	2729	2741	2751
2780	2808	2837	2872	2882	2912	2940	2950	2980	2989	3018	3028	3056
3083	3110	3144	3182	3220	3252	3282	3297	3330	3345	3362	3395	3412
3431	3460	3474	3488	3521	3557	3587	3617	3645	3654	3681	3709	3718
3745	3773	3782	3812	3821	3854	3887	3942	3951	3965	3977	3986	3999
4012	4026	4040	4098	4117	4135	4153	4169	4185	4204	4239	4274	4289
4312	4327	4350	4364	4388	4403	4438	4454	4469	4492	4506	4520	4536
4569	4592	4607	4630	4645	4677	4691	4705	4717	4732	4755	4770	4785

SXX = 177720

COMMON	1														
ENCODING	1														
ERROR	1														
	398	510	514	530	534	568	587	606	624	657	685	703	721	741	767
	781	801	821	841	876	894	912	930	965	984	994	1014	1034	1054	1088
	1131	1135	1148	1182	1195	1229	1242	1295	1305	1314	1342	1358	1382	1393	1420
	1432	1460	1473	1505	1519	1552	1567	1600	1614	1643	1659	1695	1708	1744	1761
	1797	1813	1834	1849	1885	1898	1927	1938	1968	1979	2006	2018	2051	2083	2116
	2148	2165	2197	2208	2240	2250	2265	2276	2309	2320	2356	2371	2382	2413	2423
	2456	2467	2499	2509	2541	2551	2576	2605	2631	2644	2655	2670	2681	2707	2716
	2726	2738	2748	2777	2805	2834	2869	2879	2909	2937	2947	2977	2986	3015	3025
	3053	3080	3107	3141	3179	3217	3249	3279	3294	3327	3342	3359	3392	3409	3429
	3457	3471	3485	3518	3554	3584	3614	3642	3651	3678	3706	3715	3742	3770	3779
	3809	3818	3851	3884	3939	3948	3962	3974	3982	3996	4009	4023	4037	4081	4095
	4114	4122	4150	4166	4182	4201	4232	4236	4271	4286	4309	4324	4347	4361	4385
	4400	4435	4451	4466	4489	4503	4517	4533	4566	4589	4604	4627	4642	4674	4688
	4702	4714	4729	4752	4767	4782	4816	4832	4848	4871	4886	4902	4916	4939	4973
	4988	5003	5018	5041	5056	5071	5086	5121	5136	5150	5166	5190	5205	5220	5234
	5258	5273	5287	5304	5328	5344	5359	5377	5448	5488					
ESCAPE	1														
GETPRI	1														
GETSWR	1														
JNE	5387	5428	5453	5458											
LOOP	398	518	538	571	590	609	627	670	688	706	724	744	784	804	824
	844	879	897	915	933	968	987	997	1017	1037	1057	1091	1104	1138	1151
	1185	1198	1232	1245	1298	1308	1317	1345	1361	1385	1396	1423	1435	1463	1476
	1508	1522	1555	1570	1603	1617	1646	1662	1698	1711	1747	1764	1800	1816	1837
	1852	1888	1901	1930	1941	1971	1982	2009	2021	2054	2086	2119	2151	2168	2200
	2211	2243	2253	2268	2279	2312	2323	2359	2374	2385	2416	2426	2459	2470	2502
	2512	2544	2554	2581	2608	2634	2647	2658	2673	2684	2710	2719	2729	2741	2751
	2780	2808	2837	2872	2882	2912	2940	2950	2980	2989	3018	3028	3056	3083	3110
	3144	3182	3220	3252	3282	3297	3330	3345	3362	3395	3412	3431	3460	3474	3488
	3521	3557	3587	3617	3645	3654	3681	3709	3718	3745	3773	3782	3812	3821	3854
	3887	3942	3951	3965	3977	3986	3999	4012	4026	4040	4098	4117	4135	4153	4169
	4185	4204	4239	4274	4289	4312	4327	4350	4364	4388	4403	4438	4454	4469	4492
	4506	4520	4536	4569	4592	4607	4630	4645	4677	4691	4705	4717	4732	4755	4770
	4785	4819	4835	4851	4874	4889	4905	4919	4942	4976	4991	5006	5021	5044	5059
	5074	5089	5124	5139	5153	5169	5193	5208	5223	5237	5261	5276	5290	5307	5331
	5347	5362	5380												
MULT	1														
NEWST	1														
	811	398	497	559	578	597	616	658	677	695	713	731	751	771	791
	1288	831	867	886	904	922	955	975	1004	1024	1044	1078	1125	1172	1219
	1958	1334	1375	1412	1450	1494	1540	1588	1630	1684	1731	1785	1824	1875	1917
	2594	1998	2038	2070	2102	2134	2183	2226	2293	2339	2399	2440	2484	2526	2567
	3167	2623	2697	2765	2793	2822	2855	2896	2926	2966	3002	3043	3070	3097	3129
	3760	3205	3237	3267	3314	3377	3446	3504	3539	3573	3602	3634	3668	3696	3733
	4615	3799	3841	3874	3929	4064	4220	4260	4297	4335	4372	4423	4477	4554	4577
		4663	4740	4803	4859	4927	4961	5028	5108	5177	5245	5315	5413	5462	
POP	1														
PUSH	1														
REPORT	1														
SETPRI	1														
SETUP	1														
SKIP	1														
SLASH	1														
STARS	1														
	597	398	412	423	450	452	459	475	497	499	545	559	561	578	580
		599	616	618	634	658	660	677	679	695	697	713	715	731	733

CROSS REFERENCE TABLE -- MARCH 1976

751	753	771	773	791	793	811	813	831	833	852	867	869	886	888
924	906	922	924	940	955	957	975	977	1004	1006	1024	1026	1044	1046
1065	1078	1080	1112	1125	1127	1159	1172	1174	1206	1219	1221	1253	1271	1275
1288	1290	1325	1334	1336	1369	1375	1377	1404	1412	1414	1443	1450	1452	1485
1494	1496	1530	1540	1542	1578	1588	1590	1625	1630	1632	1670	1684	1686	1719
1731	1733	1772	1785	1787	1824	1826	1860	1875	1877	1909	1517	1919	1949	1958
1960	1990	1998	2000	2030	2038	2040	2062	2070	2072	2094	2102	2104	2127	2134
2136	2176	2183	2185	2218	2226	2228	2286	2293	2295	2330	2339	2341	2391	2399
2401	2433	2440	2442	2477	2484	2486	2519	2526	2528	2561	2567	2569	2588	2594
2596	2615	2623	2625	2691	2697	2699	2758	2765	2767	2787	2793	2795	2815	2822
2824	2844	2855	2857	2889	2896	2898	2919	2926	2928	2957	2966	2968	2995	3002
3004	3035	3043	3045	3063	3070	3072	3090	3097	3099	3117	3129	3131	3158	3167
3169	3195	3205	3207	3227	3237	3239	3259	3267	3269	3305	3314	3316	3370	3377
3379	3429	3446	3448	3496	3504	3506	3529	3539	3541	3566	3573	3575	3595	3602
3604	3627	3634	3636	3661	3668	3670	3689	3696	3698	3726	3733	3735	3753	3760
3762	3790	3799	3801	3832	3841	3843	3864	3874	3876	3898	3929	3931	4048	4064
4066	4213	4220	4222	4246	4260	4262	4297	4299	4335	4337	4372	4374	4412	4423
4425	4477	4479	4545	4554	4556	4577	4579	4579	4615	4617	4663	4665	4740	4742
4793	4803	4805	4859	4861	4927	4929	4951	4961	4963	5028	5030	5098	5108	5110
5177	5179	5245	5247	5315	5317	5390	5413	5415	5462	5464				

SWSU
 TYPBIN
 TYPDEC
 TYPNAM
 TYPNUM
 TYPOCS
 TYPOCT
 TYPTXT
 SSERCO

398	511	520	531	540	573	592	611	623	672	690	708	726	746	767
786	806	826	846	881	899	917	935	970	989	999	1019	1039	1059	1094
1107	1141	1154	1188	1201	1235	1248	1300	1310	1319	1347	1364	1387	1399	1425
1438	1465	1479	1510	1525	1557	1573	1605	1620	1648	1665	1700	1714	1749	1767
1802	1819	1839	1855	1890	1904	1932	1944	1973	1985	2011	2024	2057	2089	2122
2154	2171	2203	2213	2246	2255	2271	2281	2315	2325	2362	2377	2387	2419	2428
2462	2472	2505	2514	2547	2556	2583	2610	2636	2649	2660	2675	2686	2712	2721
2731	2743	2753	2782	2810	2839	2874	2884	2914	2942	2952	2982	2991	3020	3030
3058	3085	3112	3147	3185	3222	3254	3285	3300	3333	3348	3365	3398	3415	3434
3463	3477	3491	3524	3560	3590	3620	3647	3656	3683	3711	3720	3747	3775	3784
3814	3824	3857	3890	3944	3953	3967	3979	3988	4001	4014	4028	4042	4082	4101
4120	4138	4156	4172	4188	4207	4232	4241	4277	4292	4315	4330	4353	4367	4391
4406	4441	4457	4472	4495	4509	4523	4539	4572	4595	4610	4633	4648	4680	4694
4708	4720	4735	4758	4773	4788	4822	4838	4854	4877	4892	4908	4922	4945	4979
4994	5009	5024	5047	5062	5077	5092	5127	5142	5156	5172	5196	5211	5226	5240
5264	5279	5293	5310	5334	5350	5365	5383	5448	5489					
398	511	520	531	540	573	592	611	623	672	690	708	726	746	767
786	806	826	846	881	899	917	935	970	989	999	1019	1039	1059	1094
1107	1141	1154	1188	1201	1235	1248	1300	1310	1319	1347	1364	1387	1399	1425
1438	1465	1479	1510	1525	1557	1573	1605	1620	1648	1665	1700	1714	1749	1767
1802	1819	1839	1855	1890	1904	1932	1944	1973	1985	2011	2024	2057	2089	2122
2154	2171	2203	2213	2246	2255	2271	2281	2315	2325	2362	2377	2387	2419	2428
2462	2472	2505	2514	2547	2556	2583	2610	2636	2649	2660	2675	2686	2712	2721
2731	2743	2753	2782	2810	2839	2874	2884	2914	2942	2952	2982	2991	3020	3030
3058	3085	3112	3147	3185	3222	3254	3285	3300	3333	3348	3365	3398	3415	3434
3463	3477	3491	3524	3560	3590	3620	3647	3656	3683	3711	3720	3747	3775	3784
3814	3824	3857	3890	3944	3953	3967	3979	3988	4001	4014	4028	4042	4082	4101
4120	4138	4156	4172	4188	4207	4232	4241	4277	4292	4315	4330	4353	4367	4391

SSERNU

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

BCC	4251	1431	4749	4764	4778										
BCC	4252	2668	2776	3137	3175	3213	3245	4671	4685	4699	4711	4725			
BCC	4253	5269	5284	5298											
BCC	4254	5340	5355	5371											
BCC	740	760	780	800	820	840	984	1013	1053	1228	3277	3292	3340	3357	3387
BCC	3469	3483	3550	3582	4397	4450	4727	4813	4830	4900	5016	5052	5118	5132	5188
BCC	5203	5255	5270	5326	5342	5511									
BCC	1229	2050	2114	2163	2195	2263	2307	2369	3325	3405	3423	3455	3514	3612	4382
BCC	4489	4533	4565	4588	4640	4780	4846	5038	5053	5299	5512				
BCC	504	535	568	587	606	624	667	685	703	721	741	761	791	801	821
BCC	841	876	894	912	930	965	994	1014	1034	1054	1088	1100	1295	1314	1359
BCC	1382	1393	1420	1432	1460	1473	1505	1519	1552	1567	1600	1614	1643	1659	1695
BCC	1708	1744	1761	1797	1813	1834	1849	1885	1898	1927	1938	1968	1979	2006	2016
BCC	2051	2116	2165	2197	2208	2250	2265	2276	2309	2320	2371	2423	2467	2509	2551
BCC	2578	2605	2631	2644	2655	2670	2681	2726	2738	2777	2805	2834	2869	2909	2937
BCC	2947	2977	2986	3015	3025	3053	3080	3107	3141	3179	3217	3249	3279	3294	3327
BCC	3342	3359	3392	3409	3428	3457	3471	3485	3518	3554	3584	3614	3651	3678	3706
BCC	3715	3742	3770	3779	3809	3818	3851	3884	3939	3948	3962	3974	3983	3996	4009
BCC	4023	4037	4095	4114	4132	4150	4166	4182	4201	4236	4385	4399	4451	4487	4531
BCC	4566	4589	4642	4729	4782	4815	4831	4901	5017	5041	5056	5070	5120	5135	5166
BCC	5189	5204	5219	5257	5272	5300	5304	5327	5343	5358	5375	5377	5474	5481	5500
BGE	5501														
BGT	2707	5503													
BHI	4284	4322	4359	4501	4700	4765	4869	4937	4971	5507					
BIC	2642	2667	4344	4358	4526	4825	4911	5080	5337	5429	5454	5459			
BICB	2975	3138	3176	3214	3246										
BIS	2652	3051	4381	4396	4444	4761	4895	4982	5368						
BISB	2803	2907	3078	3105	3139	3177	3215	3247							
BIT	2737	2747	4306	4321	5427										
BITB	5471	5538													
BLE	5504														
BLCS	2082	2147	2239	2355	2412	2455	2498	2540	4269	4307	4345	4433	4464	4515	4602
BLCS	4625	4672	4686	4712	4750	4884	4914	4986	5001	5084	5148	5162	5217	5232	5285
BLCS	5356	5372	5508												
BLT	2716	5502													
BMI	1135	1147	1242	1305	1342	2049	2083	2115	2148	2164	2196	2240	2264	2308	2356
BMI	2370	2382	2413	2456	2499	2541	2748	2879	3642	4271	4309	4347	4384	4400	4448
BMI	4465	4486	4502	4517	4530	4563	4586	4604	4627	4641	4688	4714	4728	4752	4781
BME	4816	4832	4902	5018	5040	5055	5071	5121	5190	5234	5258	5328	5506		
BME	502	515	524	564	583	602	621	663	682	700	718	736	756	776	796
BME	816	836	872	891	909	927	960	980	1009	1029	1049	1083	1087	1101	1130
BME	1177	1224	1293	1339	1352	1380	1417	1455	1499	1545	1593	1635	1689	1736	1790
BME	1829	1880	1922	1963	2003	2043	2075	2107	2139	2188	2231	2298	2344	2404	2445
BME	2489	2531	2572	2599	2628	2702	2770	2798	2827	2860	2901	2931	2971	3007	3048
BME	3075	3102	3134	3172	3210	3242	3272	3319	3382	3389	3407	3425	3451	3509	3516
BME	3544	3552	3578	3607	3639	3673	3701	3738	3765	3804	3846	3879	3934	4069	4087
BME	4089	4091	4093	4108	4110	4112	4130	4146	4148	4162	4164	4180	4199	4225	4265
BME	4302	4340	4377	4428	4482	4559	4582	4620	4668	4745	4808	4847	4864	4932	4966
BPL	5033	5113	5182	5250	5320	5418	5467	5470	5472	5479	5499	5539	5544		
BPL	1134	1148	1355	1391	1429	1469	1471	1517	1563	1611	1656	1706	1755	1757	1811
BPL	1845	1896	1936	1977	2016	4286	4324	4361	4434	4466	4674	4702	4767	4848	4871
BPL	4886	4916	4939	4973	4988	5003	5086	5136	5150	5164	5205	5220	5273	5287	5302
BPL	5344	5359	5374	5477	5505	5542									
BPP	505	525	4070	5498											
BPP	1181	1195	3276	3291	3324	3339	3356	4503	4532	4687	4701	4751	4829	4885	4899
BPP	4387	5015	5085	5134	5163	5202	5233	5271	5301	5341	5373	5509			

S.S	1182	1194	1241	2048	2081	2113	2146	2162	2194	2238	2262	2306	2354	2368	2411
	2454	2497	2539	4270	4285	4308	4323	4346	4360	4383	4399	4435	4449	4488	4516
	4564	4587	4603	4626	4639	4673	4713	4726	4766	4779	4814	4845	4870	4915	4938
	4972	5002	5039	5054	5069	5119	5149	5187	5218	5256	5286	5325	5357	5510	
CCC	503	1085	1132	1179	1226	4430	4697	4826	4866	4896	4934	4968	5012	5115	5159
	5139	5229	5252	5296	5369										
CLC	738	758	778	798	818	838	961	1011	1031	1051	3322	3385	3403	3467	3548
	3610	4267	4305	4343	4380	4462	4513	4724	4777	4811	5051	5131	5215	5268	5339
CLM	1146	2079	2144	2236	2352	2452	2495	2537	4267	4305	4343	4380	4395	4513	4523
	4600	4623	4748	4811	5066	5146	5185	5283	5323						
CLR	1294	1340	1418	1419	1456	1457	1500	1501	1546	1549	1551	1594	1597	1636	1639
	1690	1693	1694	1737	1740	1741	1791	1794	1830	1833	1881	1882	1884	1923	1926
	1964	1967	2004	2044	2076	2108	2109	2140	2141	2157	2189	2190	2232	2233	2299
	2300	2345	2346	2405	2406	2446	2447	2490	2491	2532	2533	2573	2575	2600	2601
	2629	2663	2665	2703	2704	2734	2736	2771	2773	2774	2799	2800	2802	2828	2829
	2831	2861	2862	2865	2902	2903	2904	2932	2933	2972	3008	3009	3010	3077	3104
	3320	3352	3383	3401	3418	3675	3703	3935	4075	4076	4378	4562	4841		
CLRB	1381	1459	1504	1599	1642	1743	1796								
CLV	1193	1240	4528	4748	4882	4984	5082	5131	5268	5339					
CLZ	1099	2046	2111	2160	2192	2260	2304	2366	2409	4282	4320	4357	4446	4499	4561
	4584	4637	4724	4763	4777	4843	5036	5051							
CMP	501	563	582	586	601	605	620	623	662	681	684	699	702	717	720
	735	755	775	795	815	835	871	890	893	908	911	926	929	959	964
	979	993	1008	1028	1033	1048	1082	1129	1176	1223	1292	1338	1379	1416	1454
	1498	1544	1592	1634	1688	1735	1789	1828	1879	1921	1962	2002	2042	2074	2106
	2138	2187	2230	2297	2343	2403	2444	2488	2530	2571	2598	2627	2701	2706	2715
	2725	2769	2797	2826	2859	2900	2930	2970	3006	3047	3074	3079	3101	3106	3133
	3140	3171	3178	3209	3216	3241	3248	3271	3278	3293	3318	3326	3341	3358	3381
	3388	3406	3424	3450	3456	3470	3484	3508	3515	3543	3551	3577	3583	3606	3613
	3638	3650	3672	3677	3700	3705	3737	3741	3764	3769	3803	3808	3817	3845	3850
	3878	3883	3933	3938	3947	3973	3982	3995	4008	4022	4036	4068	4086	4088	4090
	4092	4094	4109	4111	4113	4129	4131	4145	4147	4149	4161	4163	4165	4179	4181
	4198	4200	4224	4235	4264	4301	4339	4376	4427	4481	4558	4581	4619	4667	4744
	4807	4863	4868	4883	4898	4913	4931	4965	5032	5112	5165	5181	5249	5303	5319
	5376	5417	5466	5473											
CMPB	2833														
COM	1313	1458	1502	1550	1562	1598	1640	1705	1742	1795	1844	1895	1935	1976	2407
	2448	2492	2534	2574	2602	2640	2641	2666	2735	2772	2801	2830	2864	2866	2877
	2934	2974	3011	4077	4936	4997									
COMB	1390	1547	1595	1637	1691	1738	1792	1831	1924	1965	2077	2142	2234	2301	2319
	2347	2348	2449	2466	2493	2508	2535	2550	2664	2804	2832	2863	2906	2985	3012
DEC	1341	1357	1428	1513	1560	1561	1564	1565	1608	1612	1651	1652	1654	1657	1703
	1704	1752	1753	1758	1759	1805	1806	1807	1808	2206	2207	2249	2274	2275	2302
	2318	2643	2746	2945	2946	3390	3391	3408	3426	3427	4485	4500	4514	4529	4598
	5452	5457													
HALT	409	513	522	533	542	575	594	613	631	674	692	710	728	748	769
	788	808	828	848	883	901	919	937	972	991	1001	1021	1041	1061	1096
	1109	1143	1156	1190	1203	1237	1250	1302	1312	1321	1349	1366	1389	1401	1427
	1440	1467	1481	1512	1527	1559	1575	1607	1622	1650	1667	1702	1716	1751	1769
	1804	1821	1841	1857	1892	1906	1934	1946	1975	1987	2013	2026	2059	2091	2124
	2156	2173	2205	2215	2248	2257	2273	2283	2317	2327	2364	2379	2389	2421	2430
	2464	2474	2507	2516	2549	2558	2585	2612	2638	2651	2662	2677	2688	2714	2723
	2733	2745	2755	2784	2812	2841	2876	2886	2916	2944	2954	2984	2993	3022	3032
	3060	3087	3114	3149	3187	3224	3256	3287	3302	3335	3350	3367	3400	3417	3436
	3465	3479	3493	3526	3562	3592	3622	3649	3658	3685	3713	3722	3749	3777	3786
	3816	3826	3859	3892	3946	3955	3969	3981	3990	4003	4016	4030	4044	4084	4103

4122	4140	4158	4174	4190	4209	4234	4243	4279	4294	4317	4332	4355	4369	4392
4408	4443	4459	4474	4497	4511	4525	4541	4574	4597	4612	4635	4650	4682	4696
4710	4722	4737	4760	4775	4790	4824	4840	4856	4879	4894	4910	4924	4947	4981
4996	5011	5026	5049	5064	5079	5094	5129	5144	5158	5174	5198	5213	5228	5242
5266	5281	5295	5312	5336	5352	5367	5385	5450	5491	5534				
500	512	521	532	541	562	574	581	593	600	612	619	630	661	673
680	691	698	709	716	727	734	747	754	768	774	787	794	807	814
827	834	847	870	882	889	900	907	918	925	936	958	971	978	990
1000	1007	1020	1027	1040	1047	1060	1081	1095	1108	1128	1142	1155	1175	1183
1202	1222	1236	1249	1291	1301	1303	1311	1320	1337	1348	1365	1378	1388	1400
1415	1426	1439	1453	1466	1468	1480	1497	1503	1511	1514	1515	1526	1543	1548
1558	1565	1574	1591	1596	1606	1609	1621	1633	1638	1641	1649	1653	1666	1687
1692	1701	1707	1715	1734	1739	1750	1754	1768	1788	1793	1803	1809	1820	1827
1832	1840	1842	1843	1846	1847	1848	1856	1878	1891	1893	1894	1897	1905	1920
1925	1933	1937	1945	1961	1966	1974	1978	1986	2001	2012	2014	2017	2025	2041
2058	2073	2090	2105	2123	2137	2155	2158	2172	2186	2204	2214	2229	2247	2256
2258	2272	2282	2296	2316	2326	2342	2349	2363	2378	2388	2402	2420	2429	2443
2450	2463	2465	2473	2487	2506	2515	2529	2548	2557	2570	2577	2584	2597	2604
2611	2626	2637	2639	2650	2653	2654	2661	2669	2676	2680	2687	2700	2705	2713
2722	2724	2732	2744	2754	2768	2775	2783	2796	2811	2825	2840	2858	2875	2885
2899	2905	2908	2915	2929	2936	2943	2953	2969	2983	2992	3005	3014	3021	3031
3046	3052	3059	3073	3086	3100	3113	3132	3148	3170	3186	3208	3223	3240	3255
3270	3286	3301	3317	3334	3349	3353	3366	3380	3399	3416	3420	3435	3449	3464
3478	3492	3507	3525	3542	3561	3576	3591	3605	3621	3637	3648	3657	3671	3684
3699	3712	3721	3736	3748	3763	3776	3785	3802	3815	3825	3844	3858	3877	3891
3932	3945	3954	3957	3968	3970	3980	3989	3992	4002	4005	4015	4018	4029	4032
4043	4067	4083	4102	4104	4121	4125	4139	4141	4157	4159	4173	4175	4189	4191
4208	4223	4233	4242	4263	4278	4293	4300	4316	4331	4338	4354	4368	4375	4392
4407	4426	4432	4442	4447	4458	4463	4473	4480	4496	4510	4524	4540	4557	4573
4580	4596	4611	4618	4634	4649	4666	4681	4695	4709	4721	4736	4743	4759	4774
4789	4806	4823	4839	4855	4862	4878	4893	4909	4923	4930	4946	4964	4980	4995
5010	5025	5031	5048	5063	5078	5093	5111	5128	5143	5157	5173	5180	5197	5212
5227	5241	5248	5265	5280	5294	5311	5318	5335	5351	5366	5384	5416	5426	5449
5465	5468	5490												
INC8	1392	1472	1518	1613	1658	1760	1812	5469						
JMP	479	483	3937	3958	3972	3993	4006	4019	4033	4071	5430	5431	5434	5443
	5455	5456	5460	5461	5487	5545								
JSR	4078	4105	4127	4143	4160	4177	4192	5483						
MOV	481	482	489	490	491	492	493	494	495	496	511	520	531	540
	573	584	592	603	611	622	629	665	672	683	690	701	708	719
	737	746	757	766	767	770	777	786	797	806	817	826	837	846
	874	881	892	899	910	917	928	935	962	970	981	989	999	1010
	1030	1039	1050	1059	1094	1107	1141	1154	1188	1201	1235	1248	1300	1310
	1347	1364	1387	1399	1425	1438	1465	1479	1510	1525	1557	1573	1605	1620
	1665	1700	1714	1749	1767	1802	1819	1839	1855	1890	1904	1932	1944	1973
	2011	2024	2057	2089	2122	2154	2171	2203	2213	2246	2255	2271	2281	2315
	2362	2377	2387	2419	2428	2462	2472	2505	2514	2547	2556	2583	2603	2610
	2636	2649	2660	2675	2686	2712	2721	2731	2743	2753	2782	2810	2839	2874
	2914	2935	2942	2952	2973	2982	2991	3020	3030	3049	3050	3058	3076	3085
	3112	3135	3136	3147	3173	3174	3185	3211	3212	3222	3243	3244	3254	3273
	3288	3300	3321	3333	3337	3348	3351	3365	3384	3398	3402	3415	3419	3434
	3463	3466	3477	3480	3491	3510	3511	3524	3545	3546	3547	3560	3579	3590
	3609	3620	3640	3647	3656	3674	3683	3702	3711	3720	3739	3747	3766	3775
	3784	3805	3806	3814	3824	3847	3848	3857	3880	3881	3890	3936	3944	3956
	3967	3971	3979	3988	3991	4001	4004	4014	4017	4028	4031	4042	4073	4082
	4101	4120	4123	4124	4126	4138	4142	4156	4172	4176	4188	4207	4226	4228

	4232	4241	4268	4277	4283	4292	4303	4315	4330	4341	4353	4367	4391	4406	4429
	4441	4457	4472	4483	4495	4509	4523	4539	4572	4595	4610	4621	4633	4648	4669
	4680	4694	4708	4720	4735	4746	4758	4773	4788	4809	4822	4838	4854	4865	4877
	4880	4892	4908	4922	4933	4945	4967	4979	4994	5009	5024	5034	5047	5062	5077
	5092	5114	5127	5142	5156	5172	5183	5196	5211	5226	5240	5251	5264	5279	5293
	5310	5321	5334	5350	5365	5383	5419	5420	5421	5422	5423	5424	5425	5429	5432
	5433	5435	5436	5439	5447	5448	5453	5458	5475	5480	5489	5533	5536	5537	5540
MOVB	2867	2878	3013	5478	5543										
NEG	1304	1356	2015	4812	4828	4844									
NEGB	1470	1516	1610	1655	1756	1810	1883	2679							
NOP	403	5484	5485	5486											
RESET	5482														
ROL	739	759	779	799	819	839	963	983	1012	3275	3323	3386	3454	3513	3549
	3581	3611	5117	5132	5147	5161	5437								
ROLB	3290	3338	3355	3404	3422	3468	3482								
ROR	1032	1052	5186	5201	5216	5231									
RTS	4229														
SBC	1354	5037	5052	5067	5083										
SCC	1098	1145	1192	1239	2045	2078	2110	2143	2159	2191	2235	2259	2303	2351	2365
	2408	2451	2494	2536	4266	4281	4304	4319	4342	4356	4379	4394	4461	4484	4512
	4527	4560	4583	4599	4622	4636	4670	4723	4747	4762	4776	4810	4842	4881	4912
	4983	4998	5035	5050	5065	5081	5130	5145	5184	5214	5267	5282	5322	5338	5353
SEC	982	1227	1350	1353	1430	3274	3289	3336	3354	3421	3453	3481	3512	3580	4445
	4498	4867	4935	4969	5160	5230	5253	5297	5370						
SEN	1133	4698	4867	4969	5200	5253									
SEV	1180	5116													
SEZ	523	1086	4431	4683	4827	4897	4935	5013	5116	5160	5200	5230	5297	5370	
SUB	2678	3714	4970	4985	5000	5014									
SWAB	3641	3676	3704	3740	3768	3807	3849	3882	4624	4638					
TST	567	666	875	2047	2112	2193	2305	2350	2380	2381	2410	2422	2453	2496	2538
	2868	3023	3024	3517	3553	3778	3961	4107	4585	4601					
TSTB	2080	2145	2161	2237	2261	2353	2367	2976	5476	5541					
.ASCIZ	5493	5547													
.BYTE	435	436													
.ENABL	1	399													
.END	5551														
.ENDC	417	419	448	514	523	534	544	577	596	615	633	676	694	712	730
	750	770	790	810	830	850	885	903	921	939	974	993	1003	1023	1043
	1063	1097	1111	1144	1158	1191	1205	1238	1252	1303	1313	1323	1350	1368	1390
	1403	1428	1442	1468	1483	1513	1529	1560	1577	1608	1624	1651	1669	1703	1718
	1752	1771	1805	1823	1842	1859	1893	1908	1935	1948	1976	1989	2014	2028	2061
	2093	2126	2157	2175	2206	2217	2249	2258	2274	2285	2318	2329	2365	2380	2391
	2422	2432	2465	2476	2508	2518	2550	2560	2587	2614	2639	2652	2663	2678	2690
	2715	2724	2734	2746	2757	2786	2814	2843	2877	2888	2918	2945	2956	2985	2995
	3023	3034	3062	3089	3116	3151	3189	3226	3258	3288	3304	3336	3351	3369	3401
	3418	3438	3466	3480	3495	3528	3564	3594	3624	3650	3660	3687	3714	3724	3751
	3778	3788	3817	3828	3861	3894	3947	3956	3970	3982	3991	4004	4017	4031	4046
	4085	4104	4123	4141	4159	4175	4191	4211	4235	4245	4280	4296	4318	4334	4356
	4371	4394	4410	4444	4460	4476	4498	4512	4526	4543	4576	4598	4614	4636	4651
	4683	4697	4711	4723	4739	4761	4776	4792	4825	4841	4858	4880	4895	4911	4926
	4949	4982	4997	5012	5027	5050	5065	5080	5096	5130	5145	5159	5176	5199	5214
	5229	5244	5267	5282	5296	5314	5337	5353	5368	5387	5451	5492			
.EVEN	424	51													
.IF	415	417	446	448	511	515	523	531	535	568	587	606	624	667	685
	703	721	741	767	781	801	821	841	876	894	912	930	965	984	994
	1014	1034	1054	1088	1097	1101	1135	1144	1148	1182	1191	1195	1229	1238	1242

CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

3026	3053	3080	3107	3141	3179	3217	3249	3279	3289	3294	3327	3336	3342	3355
3359	3392	3401	3409	3418	3428	3457	3466	3471	3480	3485	3518	3554	3594	3605
3652	3650	3651	3679	3706	3714	3715	3742	3770	3778	3779	3809	3817	3818	3851
3894	3939	3947	3948	3956	3962	3970	3974	3982	3983	3991	3936	4004	4009	4017
4023	4031	4037	4081	4095	4104	4114	4123	4132	4141	4150	4159	4166	4175	4182
4191	4201	4232	4236	4271	4280	4286	4309	4318	4324	4347	4356	4361	4385	4394
4400	4435	4444	4451	4460	4466	4489	4498	4503	4512	4517	4526	4533	4566	4589
4598	4604	4627	4636	4642	4674	4683	4688	4697	4702	4711	4714	4723	4729	4752
4761	4767	4776	4782	4816	4825	4832	4841	4848	4871	4880	4886	4895	4902	4917
4916	4939	4973	4982	4988	4997	5003	5012	5018	5041	5050	5056	5065	5071	5080
5086	5121	5130	5136	5145	5150	5159	5166	5190	5199	5205	5214	5220	5229	5234
5258	5267	5273	5282	5287	5296	5304	5328	5337	5344	5353	5359	5368	5377	5448
5488														
424	487	502	511	520	531	540	543	559	564	573	576	578	583	592
595	597	602	611	614	616	621	629	632	658	663	672	675	677	682
690	693	695	700	708	711	713	718	726	729	731	736	746	749	751
756	757	767	771	776	786	789	791	796	806	809	811	816	826	829
831	836	846	849	867	872	881	884	886	891	899	902	904	909	917
920	922	927	935	938	955	960	970	973	975	980	989	992	999	1002
1024	1009	1019	1022	1024	1029	1039	1042	1044	1049	1059	1062	1078	1083	1093
1106	1110	1125	1130	1140	1153	1157	1172	1177	1187	1200	1204	1219	1224	1234
1247	1251	1288	1293	1300	1310	1319	1322	1334	1339	1347	1363	1367	1375	1380
1387	1398	1402	1412	1417	1425	1437	1441	1450	1455	1465	1478	1482	1494	1499
1510	1524	1528	1540	1545	1557	1572	1576	1588	1593	1605	1619	1623	1630	1635
1548	1664	1668	1684	1689	1700	1713	1717	1731	1736	1749	1766	1770	1785	1790
1802	1818	1822	1824	1829	1839	1854	1858	1875	1880	1890	1903	1907	1917	1922
1932	1943	1947	1958	1963	1973	1984	1988	1998	2003	2004	2011	2023	2027	2038
2043	2056	2060	2070	2075	2088	2092	2102	2107	2121	2125	2134	2139	2153	2170
2174	2183	2188	2202	2213	2216	2226	2231	2245	2255	2270	2281	2284	2293	2298
2314	2325	2328	2339	2344	2351	2376	2387	2390	2353	2404	2418	2428	2431	2440
2445	2461	2472	2475	2484	2489	2504	2514	2517	2526	2531	2546	2556	2559	2567
2572	2583	2586	2594	2599	2610	2613	2623	2628	2636	2649	2660	2675	2686	2689
2697	2702	2712	2721	2731	2743	2753	2756	2765	2770	2782	2785	2793	2798	2810
2813	2822	2827	2839	2842	2855	2860	2874	2884	2887	2896	2901	2914	2917	2926
2931	2942	2952	2955	2966	2971	2982	2991	2994	3002	3007	3020	3030	3033	3043
3048	3058	3061	3070	3075	3085	3088	3097	3102	3112	3115	3129	3134	3135	3146
3150	3167	3172	3173	3184	3188	3205	3210	3222	3225	3237	3242	3254	3257	3267
3272	3284	3299	3303	3314	3319	3332	3347	3364	3368	3377	3382	3397	3414	3433
3437	3446	3451	3462	3476	3490	3494	3504	3509	3523	3527	3539	3544	3545	3559
3563	3573	3578	3589	3593	3602	3607	3608	3619	3623	3634	3639	3647	3656	3659
3568	3673	3683	3686	3696	3701	3711	3720	3723	3733	3738	3747	3750	3760	3765
3775	3784	3787	3799	3804	3805	3814	3823	3827	3841	3846	3847	3856	3850	3874
3879	3880	3889	3893	3929	3934	3935	3944	3953	3967	3979	3988	4001	4014	4028
4042	4045	4064	4069	4070	4081	4100	4119	4137	4155	4171	4187	4206	4210	4220
4225	4232	4241	4244	4260	4265	4276	4291	4295	4297	4302	4314	4329	4333	4335
4340	4352	4366	4370	4372	4377	4390	4405	4409	4423	4428	4440	4456	4471	4475
4477	4452	4494	4508	4522	4538	4542	4554	4559	4571	4575	4577	4582	4594	4609
4613	4615	4620	4632	4647	4651	4663	4668	4679	4693	4707	4719	4734	4738	4740
4745	4757	4772	4787	4791	4803	4808	4821	4837	4853	4857	4859	4864	4876	4891
4907	4921	4925	4927	4932	4944	4948	4961	4966	4978	4993	5008	5023	5027	5028
5033	5046	5061	5076	5091	5095	5108	5113	5126	5141	5155	5171	5175	5177	5182
5195	5210	5225	5239	5243	5245	5250	5263	5278	5292	5309	5313	5315	5320	5333
5349	5364	5382	5386	5413	5418	5419	5448	5462	5467	5468	5488			
1	398	402	409	424	500	503	512	518	521	532	538	541	562	565
571	574	581	584	590	593	600	603	609	612	619	622	627	630	661
664	670	673	680	683	688	691	698	701	706	709	716	719	724	727

LIST

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DCKAAA.SEG/SOL/CRF PAGES: 100 DS:ERFD=9 SYMS: 00,00+AAA.P11
RUN-TIME: 61.84 12 SECONDS
RUN-TIME RATIO: 527/158=3.3
CORE USED: 33K (65 PAGES)