

IDENTIFICATION

PRODUCT CODE: AC-8041C-MC
PRODUCT NAME: CFKAACO 11/34 BSC INST TST
PRODUCT DATE: 30-OCT-78
MAINTAINER: DIAGNOSTIC ENGINEERING

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

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

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

*
* SUMMARY OF OPERATING INSTRUCTIONS *
*

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

OPERATING PROCEDURES:

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

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

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

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE

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

1.2.2 SOFTWARE

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

THE PROGRAM CAN ALSO BE RUN UNDER THE ACT 11 MONITOR

1.3 RELATED DOCUMENTS AND STANDARDS

PDP-11/34 MICROCODE LISTING

PDP-11/34 ELECTRICAL SCHEMATICS

DIAGNOSTIC ENGINEERING PROJECT PLANFOR 11/34

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

APT INTERFACE SPECIFICATION, REVISION 9.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

1.5 FAILURE ASSUMPTIONS

NONE

2.0 OPERATING INSTRUCTIONS2.1 LOADING AND STARTING PROCEDURES2.1.1 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
START2.1.3 SUBTEST START

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

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

2.2 SPECIAL ENVIRONMENTS

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

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

2.3 PROGRAM OPTIONS

SEQ 0005

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

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

2.4 EXECUTION TIMES

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

3.0 ERROR INFORMATION

3.1 ERROR TYPES

THERE ARE TWO BASIC TYPES OF ERRORS IN THE DIAGNOSTIC.

3.1.1 FUNCTIONAL ERRORS

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

3.1.2 SEQUENCE ERRORS

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

3.2 ERROR REPORTING PROCEDURES

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

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

3.3 ERROR DESCRIPTOR INFORMATION

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

3.2.1 \$MSGTYP

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

3.2.2 \$FATAL

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

3.2.3 \$PASS

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

3.2.4 \$TESTN

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

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.

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

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

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

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

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

5.1

CHECK THE MAILBOX

1. \$MSGTY THIS LOCATION SHOULD CONTAIN A 1. IF THE PROCESSOR HALTS AND THIS LOCATION IS ZERO, THEN THE PROCESSOR HAS COME TO AN UNEXPECTED HALT. FIRST SUSPECT A TRAP. CHECK THE PC AND IF A TRAP CHECK R6 AND THE STACK FOR THE LOCATION OF THE FAILING INSTRUCTION.
2. \$FATAL THIS LOCATION IS USED TO HOLD THE NUMBER OF THE ERROR WHICH HAS BEEN DETECTED. EACH ERROR BEING CHECKED BY THE DIAGNOSTIC IS ASSIGNED A UNIQUE NUMBER WHICH IS STORED IN \$FATAL WHEN THAT ERROR IS DETECTED.

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

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).

14	ACT11 HOOKS
25	APT MAILBOX-ETABLE
52	APT PARAMETER BLOCK
130	T1 CHECK BRANCHES ON Z BIT
177	DATA PATH TESTS
194	T3 TEST OF ZEROS IN THE DATA PATH
213	T3 TEST OF PATTERN 125252 IN DATA PATH
233	T4 TEST OF PATTERN 052525 IN DATA PATH
253	T5 TEST OF ALL ONES IN DATA PATH
270	B-REGISTER TEST
308	T6 SHIFT BIT 0 TO BIT 1
338	T7 SHIFT CARRY INTO BIT 0
363	T10 LEFT SHIFT FROM BIT 0 TO C-BIT
384	T11 SHIFT BIT 15 TO BIT 14
407	T12 RIGHT SHIFT FROM BIT 15 TO C-BIT
436	SCRATCH PAD TESTS
456	T13 TEST IF R0 CAN HOLD ALL ZEROS
476	T14 TEST IF R0 CAN HOLD ONES AND ZEROS
494	T15 TEST IF R0 CAN HOLD ZEROS AND ONES
513	T16 TEST IF R0 CAN HOLD ALL ONES
538	T17 TEST IF R1 CAN HOLD A ONE IN ALL BITS
563	T20 TEST IF R1 CAN HOLD A ZERO IN ALL BITS
588	T21 TEST IF R2 CAN HOLD A ONE IN ALL BITS
611	T22 TEST IF R2 CAN HOLD A ZERO IN ALL BITS
636	T23 TEST IF R3 CAN HOLD A ONE IN ALL BITS
662	T24 TEST IF R3 CAN HOLD A ZERO IN ALL BITS
687	T25 TEST IF R4 CAN HOLD A ONE IN ALL BITS
714	T26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
739	T27 TEST IF R5 CAN HOLD A ONE IN ALL BITS
765	T30 TEST IF R5 CAN HOLD A ZERO IN ALL BITS
790	T31 TEST IF R6 CAN HOLD A ONE IN ALL BITS
814	T32 TEST IF R6 CAN HOLD A ZERO IN ALL BITS
831	PSW TESTS
851	T33 TEST IF PSW WILL HOLD ZEROS
870	T34 TEST IF PSW WILL HOLD ONES AND ZEROS
889	T35 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROS AND ONES
904	T36 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
922	CONDITION CODE TEST
970	T37 TEST BRANCHES AROUND Z-BIT
1018	T40 TEST BRANCHES AROUND S-BIT
1066	T41 TEST BRANCHES AROUND V-BIT
1099	T42 TEST BRANCHES AROUND C-BIT
1136	MICROCODE TESTS
1184	T43 TEST MODE 0 USING SOP INST.
1227	T44 TEST REMAINDER OF SOP INSTS IN MODE 0
1255	T45 TEST MODE 0 EVEN BYTE USING SOP INST
1304	T46 TEST MODE 1 USING SOP INST
1350	T47 TEST MODE 1 EVEN BYTE USING SOP INST
1398	T50 TEST MODE 1 ODD BYTE USING SOP INST
1447	T51 TEST MODE 2 USING SOP INST.
1491	T52 TEST MODE 2 EVEN BYTE USING SOP INST.
1538	T53 TEST MODE 2 ODD BYTE USING SOP INST.
1595	T54 TEST MODE 0 USING NEGATE INSTRUCTION
1652	T55 TEST MODE 1 USING NEGATE INST.
1714	T56 TEST MODE 2 USING NEGATE INSTRUCTION
	T57 TEST MODE 3 USING SOP INST.

1762	T60 TEST MODE 3 EVEN BYTE USING SOP INST.
1817	T61 TEST MODE 3 ODD BYTE USING SOP INST.
1856	T62 TEST MODE 3 USING NEGATE INSTRUCTION
1933	T63 TEST MODE 4 USING SOP INSTS
2085	T64 TEST MODE 5 USING SOP INSTS
2228	T65 TEST MODE 6 USING SOP INSTS
2070	T66 TEST MODE 7 USING SOP INST
2104	T67 TEST MODE 4 WITH NEGATE INSTRUCTION
2146	T70 TEST MODE 5 WITH NEGATE INSTRUCTION
2193	T71 TEST MODE 6 WITH NEGATE
2229	T72 TEST MODE 7 W/ NEGATE
2245	T73 TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
2516	T74 TEST MODE 0 SOP NON-MODIFYING
2349	T75 TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING
2382	T76 TEST MODE 1 SOP NON-MODIFYING
2452	T77 TEST MODE 1 BYTE INST. NON-MODIFYING
2509	T100 TEST MODE 2 WITH SOP NON-MODIFYING
2577	T101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
2624	T102 TEST MODE 3 W/ SOP NON-MODIFYING INSTS
2685	T103 TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
2727	T104 TEST MODE 4 W/ SOP NON-MODIFYING INSTS
2772	T105 TEST MODE 5 W/ SOP NON-MODIFYING INSTS
2815	T106 TEST MODE 6 W/ SOP NON-MODIFYING INSTS
2857	T107 TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
2885	T110 TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.
2913	T111 MOV MODE 0 TO MODE 0
3029	T112 TEST SUB MODE 0,0
3071	T113 TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
3137	T114 TEST MODE 0 X DOUBLE-OPERAND INSTRUCTIONS
3181	T115 TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
3210	T116 TEST MODE 0, X DOUBLE-OPERAND NON-MODIFYING INSTS.
3240	T117 TEST MODE 1 W/ DOP INST.
3274	T120 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
3316	T121 TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
3347	T122 TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
3388	T123 TEST MODE 1-ODD BYTE W/ DOP INSTS.
3425	T124 TEST MODE 2 W/ DOP INSTS.
3466	T125 TEST MODE 2 - EVEN BYTE W/ DOP INST.
3493	T126 TEST MODE 2 - ODD BYTE W/ DOP INST.
3520	T127 TEST MODE 3 W/ DOP INSTS.
3541	T130 TEST MODE 3 - EVEN BYTE W/ DOP INSTS.
3575	T131 TEST MODE 3 - ODD BYTE W/ DOP INSTS.
3609	T132 TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
3653	T133 TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3721	T134 TEST DEST. MODE 2 W/ DOP NON-MODIFYING INST.
3783	T135 TEST DEST. MODE 2-BYTES W/DOP NON-MODIFYING INST.
3828	T136 TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
3893	T137 TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
4032	T140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
4072	T141 TEST DEST. MODE 5 W/DOP NON-MODIFYING INST.
4121	T142 TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
4188	T143 TEST DEST. MODE 7 W/DOP NON-MODIFYING INST.
	T144 TEST MOV DESTINATION MODE 1
	T145 TEST MOV DESTINATION MODE 2
	T146 TEST MOV-BYTE DESTINATION MODE 2
	T147 TEST MOV(B) DESTINATION MODE 3

4256	T150	TEST MOV DESTINATION MODE 4
4306	T151	TEST MOVB DESTINATION MODE 4
4376	T152	TEST MOV DESTINATION MODE 5
4446	T153	TEST MOV DESTINATION MODE 6
4515	T154	TEST MOV DESTINATION MODE 7
4585	T155	TEST MODE 4 W/ DOP INSTS.
4627	T156	TEST MODE 5 W/ DOP INSTS.
4665	T157	TEST MODE 6 W/ DOP INSTS.
4697	T160	TEST MODE 7 W/ DOP INSTS.
4727	T161	TEST ROTATE INSTRUCTIONS OF MODE 0
4776	T162	TEST ROTATE INSTRUCTIONS W/ MODE 1
4839	T163	TEST ROTATE INSTRUCTIONS W/ MODE 2
4909	T164	TEST ROTATE INSTRUCTIONS W/ MODE 3
4968	T165	TEST MODE 4 W/ ROTATE INSTRUCTIONS
5004	T166	TEST MODE 5 W/ ROTATE INSTRUCTIONS
5039	T167	TEST MODE 6 W/ ROTATE INSTRUCTIONS
5069	T170	TEST MODE 7 W/ ROTATE INSTRUCTIONS
5102	T171	TEST MODE 0 W/ SWAB INST.
5137	T172	TEST MODE 1 W/ SWAB INST.
5166	T173	TEST MODE 2 W/ SWAB INST.
5204	T174	TEST MODE 3 W/ SWAB INST.
5232	T175	TEST MODE 4 W/ SWAB INST.
5262	T176	TEST MODE 5 W/ SWAB INST.
5315	T177	TEST MODE 6 W/ SWAB INST.
5349	T200	TEST MODE 7 W/ SWAB INST.
5405	T201	TEST THE JMP INSTRUCTION IN ALL MODES
5541	T202	TEST JSR INSTRUCTION W/ ALL MODES
5598	T203	TEST RFS INSTRUCTION
5638	T204	TEST RPS INSTRUCTION
5775	T205	TEST BIT INSTRUCTION
5813	T206	TEST BIC INSTRUCTION
5850	T207	TEST BIS INSTRUCTION
5901	T210	TEST INC INSTRUCTION
5958	T211	TEST DEC INSTRUCTION
6034	T212	TEST CLR INSTRUCTION
6058	T213	TEST TST INSTRUCTION
6096	T214	TEST SWAB INSTRUCTION
6144	T215	TEST ADD INSTRUCTION
6222	T216	TEST ADC INSTRUCTION
6282	T217	TEST MOD INSTRUCTION
6343	T220	TEST CMP INSTRUCTION
6412	T221	TEST COM INSTRUCTION
6447	T222	TEST SUB INSTRUCTION
6505	T223	TEST SBC INSTRUCTION
6544	T224	TEST SOB INSTRUCTION
6564	T225	TEST ROR INSTRUCTION
6732	T226	TEST ASL INSTRUCTION
6802	T227	TEST ASR INSTRUCTION
6886	T230	TEST THE SXT INSTRUCTION
6940	T231	TEST THE XOR INSTRUCTION
6992	T232	TEST SOB INSTRUCTION
7037	T233	TEST MARK INSTRUCTION
7101	T234	TEST MTPS INSTRUCTION
7138	T235	TEST MTPS MODE 2
7169	T236	TEST MTPS MODE 3
7201	T237	TEST MTPS MODE 4

7232	T240	TEST MTPS MODE 5
7263	T241	TEST MTPS MODE 6
7294	T242	TEST MTPS MODE 7
7333	T243	TEST MFPS INSTRUCTION
7367	T244	TEST MFPS MODE 2
7410	T245	TEST MFPS MODE 3
7453	T246	TEST SOB MODE 4
7496	T247	TEST MFPS MODE 5
7539	T250	TEST MFPS MODE 6
7582	T251	TEST MFPS MODE 7
7633	T252	TEST THAT RESET DOES NOT CLEAR PSW
7667	T254	TEST USER MODE R6 CAN HOLD 1 ONE IN EVERY POSITION
7687	T254	TEST INDEPENDENCE USER AND KERNEL MODE R6'S
7728	T255	TEST MFPI WITH R6 IN MODE 0
7753	T256	TEST MFPI WITH R6 IN MODE 0
7808	T257	TEST THE BRANCH ROR
7866	T260	DUAL REGISTER ADDRESSING TEST
7917	T261	TEST BYTE INSTRUCTION ON PSW
7941	T262	TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES
7978	T263	TEST SET CC AND CLEAR CC INSTRUCTIONS
8030	T264	END OF PASS SEQUENCE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

000500

000240
000007
000006
177576
177564
140000
030000

000400
000046
026034
000052
000000
000400
000300

000300
000300
000302
000304
000306
000310
000312
000314
000316
000320
000321
000322
000324
000326

000330

```
.TITLE CFKAACO 11/34 BSC INST TST
.ENABLE ABS
STBOT=500
.NLIST CMD,MC,MD
.LIST ME
SCOPE=NOP
R7=37
R6=36
PS=177776
TS=177564
TPB=177566
USRM=140000
PUSRM=30000

.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
$SVPC=
;SAVE PC
;=46
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
;=52
$WORD 0 ;;2)SET LOC.52 TO ZERO
;=$SVPC
; RESTORE PC
;=300

.SBTTL APT MAILBOX-ETABLE
;*****
.EVEN
$MAIL: ;APT MAILBOX
$MSGTY: .WORD AMSGTY ;MESSAGE TYPE CODE
$FATAL: .WORD AFATAL ;FATAL ERROR NUMBER
$TESTN: .WORD ATESTN ;TEST NUMBER
$ASS: .WORD APASS ;PASS COUNT
$DEVCT: .WORD ADEVCT ;DEVICE COUNT
$UNIT: .WORD AUNIT ;I/O UNIT NUMBER
$MSGAD: .WORD AMSGAD ;MESSAGE ADDRESS
$MSGLG: .WORD AMGLG ;MESSAGE LENGTH
$TABLE: ;APT ENVIRONMENT TABLE
$ENV: .BYTE AENV ;ENVIRONMENT BYTE
$ENVM: .BYTE AENVM ;ENVIRONMENT MODE BITS
$SWREG: .WORD ASWREG ;APT SWITCH REGISTER
$USWR: .WORD AUSWR ;USER SWITCHES
$CPUOP: .WORD ACPUPOP ;CPU TYPE OPTIONS
;BITS 15-11=CPU TYPE
; 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
; 11/70=06,PDQ=07,Q=10
; BIT 10=REAL TIME CLOCK
; BIT 9=FLOATING POINT PROCESSOR
; BIT 8=MEMORY MANAGEMENT

$ETEND:
.NEXIT
.SBTTL APT PARAMETER BLOCK
;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
```

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112

000330
000024
000200
000044
000330
000330

000330
000330
000332
000334
000336
000340
000342

000004
000006
000010
000012
000014
000030
000032
000034
000036
000114
000116
000244
000246
000250
000252

000370
000376
000404

000500
000200
000204
000210

```
-$X=
;=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)
$STM: .WORD 10 ;RUN TIM OF LONGEST TEST
$PRSTH: .WORD 10 ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 10 ;ADDITIONAL RUN TIME (SECS) OF 1 PASS FOR EACH ADDITIONAL UNIT
;*****
;SOME POINTERS TO CPU TRAP HANDLERS
;*****
;=4
;T04
0
;T010
0
;T014
;=30
;T030
0
;T034
0
;=114
;T0114
0
;=244
;T0244
0
;T0250
0
;*****
;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
;*****
;=370
0,0,0,0,0,0
;=500
1,1,-1
;*****
;SET UP STARTING ADDRESS
-$X=
;=200
JMP START
NOV #STBOT,R6 ;SET STACK POINTER
NOV #$TESTN,R2 ;SET MAILBOX POINTER
```

```

113 000214 000137          JMP      @ (PC)+          ;JUMP TO SUBTEST
114 000216 000000          ;ADDR. OF SUBTEST GOES HERE
115
116          000500          ;=- $X
117          000302          $ERROR=$FATAL
118          000304          $STSNM=$TESTN
119 000500 012737 026310 000024  START:  MOV      #PWRDN @#24          ;SET UP FOR POWER FAIL
120 000506 012737 000000 000306  MOV      #0,@#PASS          ;CLEAR PASS COUNT
121 000514 012737 177777 026060  MOV      #1,@#PASSPT        ;SET PRINT COUNTER
122 000522 012706 000500          MOV      #ST60,R6          ;INITIALIZE STACK POINTER
123 000526 012702 000304          MOV      #STESTN,R2        ;SET UP POINTER TO MESSAGE TYPE
124 000532 012737 000000 000304  MOV      #0,@#STSNM        ;CLEAR TEST NUMBER
125 000540 012737 000000 000302  MOV      #0,@#ERROR        ;CLEAR ERROR NUMBER
126 000546 012737 000000 000300  MOV      #0,@#MSGTY        ;CLEAR MESSAGE TYPE(FOR APT)
  
```

```

127          ;*****
128          ;TEST 1 CHECK BRANCHES ON Z BIT
129          ;*****
130 000554 005212 000001          TST1:  INC      (R2)          ;UPDATE TEST NUMBER
131 000556 022712 000001          CMP      #1,(R2)          ;SEQUENCE ERROR?
132 000562 001024          BNE     TST2-10          ;BR TO ERROR HALT ON SEQ ERROR
133 000564 000257          CCC          ;CLEAR ALL CONDITION CODES
134 000566 001401          BEQ     BR1              ;SHOULD BRANCH
135 000570 000404          BR      BR2              ;BAD BRANCH OF Z-BIT
136          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
137          ; BRANCH INSTRUCTION AND <====
138          ; REPLACE THE MOVE INSTRUCTION <====
139          ; FOLLOWING W/ 774 <====
140          000572          BR1:  MOV      #1,(R2)          ;MOVE TO MAILBOX # ***** 1 *****
141 000576 012742 000001          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
142 000578 005242          HALT          ;SHOULD HAVE BRANCHED: Z=0
143 000600 000000          BR2:  BNE     BR3
144 000602 001004          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
145          ; BRANCH INSTRUCTION AND <====
146          ; REPLACE THE MOVE INSTRUCTION <====
147          ; WHICH FOLLOWS W/ 770 <====
148          000604 012742 000002          MOV      #2,(R2)          ;MOVE TO MAILBOX # ***** 2 *****
149 000610 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
150 000612 000000          BR3:  SEZ          ;
151 000614 000264          BNE     BR4              ;
152 000616 001001          BR      BR5              ;
153 000620 000404          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
154          ; BRANCH INSTRUCTION AND <====
155          ; REPLACE THE MOVE INSTRUCTION <====
156          ; FOLLOWING W/ 760 <====
157          000622          BR4:  MOV      #3,(R2)          ;MOVE TO MAILBOX # ***** 3 *****
158 000624 012742 000003          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
159 000626 005242          HALT          ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
160 000630 000000          BR5:  BEQ     TST2
161 000632 001404          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
162          ; BRANCH INSTRUCTION AND <====
163          ; REPLACE THE MOVE INSTRUCTION <====
164          ; WHICH FOLLOWS W/ 754 <====
165          000634 012742 000004          MOV      #4,(R2)          ;MOVE TO MAILBOX # ***** 4 *****
166 000640 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
167 000642 000000          HALT          ;SHOULD HAVE BRANCHED ON Z=1
168          ; OR SEQUENCE ERROR
169
170
171
172
173
  
```

```
174 ;*****  
175 ;SBTTL DATA PATH TESTS  
176 ;  
177 ; THE DATA PATH TESTS ARE USED TO VERIFY THAT VARIOUS  
178 ; DATA PATTERNS CAN BE SUCCESSFULLY MOVED THROUGH THE DATA PATHS  
179 ; MOVE AND COMPARE MODE 2,3 INSTRUCTIONS ARE USED TO PASS AND  
180 ; TEST VARIOUS DATA PATTERNS IN THE DATA PATHS.  
181 ; THE TEST EXERCISES THE INTERNAL DATA PATHS, THE UNIBUS  
182 ; DATA TRANSCIEVERS, AND AMUX CONTROL FOR ALU AND BUS INPUTS.  
183 ; IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)  
184 ; TO SEE WHICH BITS OF THE DATA PATH ARE FAILING. IF THIS PROVIDES  
185 ; INCONCLUSIVE DATA, TRY TO CHECK MODE 3 IR DECODE BY RUNNING  
186 ; JUST THE MICROCODE AND IR DECODE TESTS FOR THE MOVE AND COMPARE  
187 ; INSTRUCTIONS.  
188 ;*****  
189 ;TEST 2 TEST OF ZEROS IN THE DATA PATH  
190 ;*****  
191 ;  
192 000644 005212 ;UPDATE TEST NUMBER  
193 000646 022712 ;SEQUENCE ERROR?  
194 000652 001006 ;BR TO ERROR HALT ON SEQ ERROR  
195 000654 012737 000000 000000 ;MOVE ZEROES THRU ADDRESS LINES, DATA  
196 ; LINES AND INTERNAL PATHS  
197 000662 005737 000000 ;SUCCESSFUL?  
198 000666 001404 ;  
199 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
200 ; CONDITIONAL BRANCH INST. AND <====  
201 ; REPLACE THE MOVE INSTRUCTION <====  
202 ; WHICH FOLLOWS W/ 771 <====  
203 000670 012742 000005 ;  
204 000674 005242 ;MOVE TO MAILBOX # ***** 5 *****  
205 000676 000000 ;SET MSGTYP TO FATAL ERROR  
206 ; DATA INCORRECT  
207 ; OR SEQUENCE ERROR  
208 ;*****  
209 ;TEST 3 TEST OF PATTERN 125252 IN DATA PATH  
210 ;*****  
211 000700 005212 ;UPDATE TEST NUMBER  
212 000702 022712 ;SEQUENCE ERROR?  
213 000706 001007 ;BR TO ERROR HALT ON SEQ ERROR  
214 000710 012737 125252 000000 ;MOVE ALTERNATING ONES AND ZERES  
215 ; THRU DATA PATHS  
216 000716 022737 125252 000000 ;SUCCESSFUL  
217 000724 001404 ;  
218 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
219 ; CONDITIONAL BRANCH INST. AND <====  
220 ; REPLACE THE MOVE INSTRUCTION <====  
221 ; WHICH FOLLOWS W/ 771 <====  
222 000726 012742 000006 ;  
223 000732 005242 ;MOVE TO MAILBOX # ***** 6 *****  
224 000734 000000 ;SET MSGTYP TO FATAL ERROR  
225 ; DATA INCORRECT  
226 ; OR SEQUENCE ERROR
```

```
226 ;*****  
227 ;TEST 4 TEST OF PATTERN 052525 IN DATA PATH  
228 ;*****  
229 ;  
230 000736 005212 ;UPDATE TEST NUMBER  
231 000740 022712 ;SEQUENCE ERROR?  
232 000744 001007 ;BR TO ERROR HALT ON SEQ ERROR  
233 000746 012737 052525 000000 ;MOVE ALTERNATING ZEROES AND ONES  
234 ; THRU DATA PATH  
235 000754 022737 052525 000000 ;SUCCESSFUL?  
236 000762 001404 ;  
237 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
238 ; CONDITIONAL BRANCH INST. AND <====  
239 ; REPLACE THE MOVE INSTRUCTION <====  
240 ; WHICH FOLLOWS W/ 771 <====  
241 000764 012742 000007 ;  
242 000770 005242 ;MOVE TO MAILBOX # ***** 7 *****  
243 000772 000000 ;SET MSGTYP TO FATAL ERROR  
244 ; DATA INCORRECT  
245 ; OR SEQUENCE ERROR  
246 ;*****  
247 ;TEST 5 TEST OF ALL ONES IN DATA PATH  
248 ;*****  
249 ;  
250 000774 005212 ;UPDATE TEST NUMBER  
251 000776 022712 ;SEQUENCE ERROR?  
252 001002 001007 ;BR TO ERROR HALT ON SEQ ERROR  
253 001004 012737 177777 000000 ;MOVE ONES THRU DATA PATH  
254 001012 022737 177777 000000 ;SUCCESSFUL  
255 001020 001404 ;  
256 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
257 ; CONDITIONAL BRANCH INST. AND <====  
258 ; REPLACE THE MOVE INSTRUCTION <====  
259 ; WHICH FOLLOWS W/ 771 <====  
260 001022 012742 000010 ;  
261 001026 005242 ;MOVE TO MAILBOX # ***** 10 *****  
262 001030 000000 ;SET MSGTYP TO FATAL ERROR  
263 ; DATA INCORRECT  
264 ; OR SEQUENCE ERROR
```

```
263 ;*****  
264 ;SBTTL B-REGISTER TEST  
265 ;  
266 ; THE B-REGISTER SHIFTING LOGIC TESTS ARE USED TO TEST THAT THE  
267 ; B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT THE ASSOCIATED  
268 ; LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE B-REGISTER AND C-BIT.  
269 ; A ONE IS SHIFED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN  
270 ; BOTH DIRECTIONS.  
271 ; THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS  
272 ; A SHIFT REGISTER. DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU,  
273 ; IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE  
274 ; WHICH BITS OF THE B-REGISTER MAY BE FAILING. IF THIS PROVIDES  
275 ; INCONCLUSIVE DATA TRY TO CHECK THE MODE 3 IR DECODE BY RUNNING JUST  
276 ; THE MICROCODE AND IR DECODE TESTS FOR THE PARTICULAR INSTRUCTIONS.  
277 ;*****  
278 ;TEST 6 SHIFT BIT 0 TO BIT 1  
279 ;*****  
280 TST6: INC (R2) ;UPDATE TEST NUMBER  
281 CMP #0,(R2) ;SEQUENCE ERROR?  
282 BNE TST7-10 ;BR TO ERROR HALT ON SEQ ERROR  
283 CLC ;CLEAR CARRY BIT  
284 MOV #1,@#0 ;LOAD A 1  
285 ROL @#0 ;SHIFT LEFT  
286 CMP #0,@#0 ;SUCCESSFUL  
287 BEQ TST7 ;  
288 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
289 ; CONDITIONAL BRANCH INST. AND <=====  
290 ; REPLACE THE MOVE INSTRUCTION <=====  
291 ; WHICH FOLLOWS W/ 766 <=====  
292 ;*****  
293 001032 005212 000006 MOV #11,-(R2) ;MOVE TO MAILBOX # ***** 11 *****  
294 001034 022712 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
295 001040 010112 HALT ;BIT 1 NOT SET  
296 001042 000241 ;OR SEQUENCE ERROR  
297 001044 012737 000001 000000  
298 001052 006137 000000 000000  
299 001056 022737 000002 000000  
300 001064 001404  
301 ;*****  
302 ;TEST 7 SHIFT CARRY INTO BIT 0  
303 ;*****  
304 TST7: INC (R2) ;UPDATE TEST NUMBER  
305 CMP #0,(R2) ;SEQUENCE ERROR?  
306 BNE TST10-10 ;BR TO ERROR HALT ON SEQ ERROR  
307 MOV #0,@#0 ;CLEAR LOCATION  
308 CLC ;CLEAR CARRY  
309 ROL @#0 ;ROTATE CARRY BIT TO BIT 0  
310 BCC TST10 ;  
311 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
312 ; CONDITIONAL BRANCH INST. AND <=====  
313 ; REPLACE THE MOVE INSTRUCTION <=====  
314 ; WHICH FOLLOWS W/ 771 <=====  
315 ;*****  
316 001124 012742 000012 MOV #12,-(R2) ;MOVE TO MAILBOX # ***** 12 *****  
317 001130 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
318 001132 000000 HALT ;CARRY CLEAR  
319 ;OR SEQUENCE ERROR  
320 ;BIT 0 SET
```

```
319 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
320 ; CONDITIONAL BRANCH INST. AND <=====  
321 ; REPLACE THE MOVE INSTRUCTION <=====  
322 ; WHICH FOLLOWS W/ 761 <=====  
323 ;*****  
324 001144 012742 000013 MOV #13,-(R2) ;MOVE TO MAILBOX # ***** 13 *****  
325 001150 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
326 001152 000000 HALT ;BIT 0 NOT SET  
327 ;OR SEQUENCE ERROR  
328 ;*****  
329 ;TEST 10 LEFT SHIFT FROM BIT 0 TO C-BIT  
330 ;*****  
331 TST10: INC (R2) ;UPDATE TEST NUMBER  
332 CMP #10,(R2) ;SEQUENCE ERROR?  
333 BNE TST11-10 ;BR TO ERROR HALT ON SEQ ERROR  
334 MOV #1,@#0 ;SET BIT 0  
335 MOV #21,RO ;SET BIT COUNTER  
336 CLC ;CLEAR C-BIT  
337 SHL: INC RO ;INCREMENT BIT COUNTER  
338 BEQ SHLE ;BR TO ERROR HALT IF BIT IS LOST  
339 ROL @#0 ;SHIFT LEFT ONE POSITION  
340 SHL ;BRANCH IF C-BIT NOT SET  
341 BEQ TST11 ;  
342 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
343 ; CONDITIONAL BRANCH INST. AND <=====  
344 ; REPLACE THE MOVE INSTRUCTION <=====  
345 ; WHICH FOLLOWS W/ 764 <=====  
346 ;*****  
347 001214 012742 000014 SHLE: MOV #14,-(R2) ;MOVE TO MAILBOX # ***** 14 *****  
348 001220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
349 001222 000000 HALT ;LEFT SHIFTING LOGIC FAILED  
350 ;OR SEQUENCE ERROR  
351 ;*****  
352 ;TEST 11 SHIFT BIT 15 TO BIT 14  
353 ;*****  
354 TST11: INC (R2) ;UPDATE TEST NUMBER  
355 CMP #11,(R2) ;SEQUENCE ERROR?  
356 BNE TST12-10 ;BR TO ERROR HALT ON SEQ ERROR  
357 MOV #100000,@#0 ;SET BIT 15  
358 CLC ;CLEAR CARRY  
359 ROR @#0 ;SHIFT BIT 15 TO BIT 14  
360 CMP #100000,@#0 ;SUCCESSFUL  
361 BEQ TST12 ;  
362 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
363 ; CONDITIONAL BRANCH INST. AND <=====  
364 ; REPLACE THE MOVE INSTRUCTION <=====  
365 ; WHICH FOLLOWS W/ 766 <=====  
366 ;*****  
367 001260 012742 000015 MOV #15,-(R2) ;MOVE TO MAILBOX # ***** 15 *****  
368 001264 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
369 001266 000000 HALT ;BIT 14 NOT SET  
370 ;OR SEQUENCE ERROR  
371 ;*****  
372 ;TEST 12 RIGHT SHIFT FROM BIT 15 TO C-BIT  
373 ;*****  
374
```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 10
CFKAAC.P11 18-OCT-78 11:01 T12 RIGHT SHIFT FROM BIT 15 TO C-BIT SEQ 0022

375 001270 005212 TST12: INC (R2) ;UPDATE TEST NUMBER
376 001272 022712 000012 CMP #12,(R2) ;SEQUENCE ERROR?
377 001276 001014 BNE TST13-10 ;BR TO ERROR HALT ON SEQ ERROR
378 001300 012737 100000 MOV #100000,@#0 ;SET BIT 15
379 001306 012700 177757 MOV # -21,R0 ;SET BIT COUNTER
380 001312 000241 CLC ;CLEAR C-BIT
381 001314 005200 SHR: INC R0 ;INCREMENT BIT COUNTER
382 001316 001404 BEQ SHRE ;BR TO ERROR HALT IF BIT IS LOST
383 001320 006037 000000 ROR #0 ;ROTATE RIGHT ONE POSITION
384 001324 103373 BCC SHR ;BRANCH IF C-BIT CLEAR
385 001326 001404 BEQ TST13
386 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
387 ; CONDITIONAL BRANCH INST. AND <===
388 ; REPLACE THE MOVE INSTRUCTION <===
389 ; WHICH FOLLOWS W/ 764 <===
390 SHRE: MOV #16,-(R2) ;MOVE TO MAILBOX # ***** 16 *****
391 INC -(R2) ;SET MSGTYP TO FATAL ERROR
392 HALT ;RIGHT SHIFT LOGIC FAILED
393 ; OR SEQUENCE ERROR
394

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 11
CFKAAC.P11 18-OCT-78 11:01 T12 RIGHT SHIFT FROM BIT 15 TO C-BIT SEQ 0023

395 ;*****
396 ;SBTTL SCRATCH PAD TESTS
397 ;
398 ;
399 ; THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
400 ; DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
401 ; CIRCUITRY. MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
402 ; R0 CAN HOLD VARIOUS DATA PATTERNS. EACH DATA PATTERN IS
403 ; MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING. THE
404 ; SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
405 ; TO THE SCRATCH PAD ITSELF.
406 ; THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
407 ; A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
408 ; BIT AT A TIME INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT
409 ; NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
410 ; CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
411 ; ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
412 ; THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROS WERE PICKED.
413 ; AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
414 ; AS WELL AS REGISTER 11. REGISTERS 10 AND 12 HAVE BEEN ACCESSED BY
415 ; THE INSTRUCTIONS. REGISTERS 13,14, AND 17 WILL BE TESTED LATER IN THE
416 ; MICROCODE TESTS.
417 ; IF THE PATTERN TESTS WITH REGISTER 0 FAIL CHECK THE RESULTANT
418 ; DATA FOR A CLUE TO A FAULT IN THE EXTERNAL CIRCUITRY. IF THE
419 ; PATTERN TESTS WITH R0 ARE SUCCESSFUL BUT THE TESTS WITH THE OTHER
420 ; REGISTERS FAIL, SUSPECT THE REGISTER SELECT LINES AND THEN THE SCRATCH
421 ; PAD ITSELF.
422 ;*****
423 ;
424 ;TEST 13 TEST IF R0 CAN HOLD ALL ZEROES
425 ;*****
426 001340 005212 TST13: INC (R2) ;UPDATE TEST NUMBER
427 001342 022712 000013 CMP #13,(R2) ;SEQUENCE ERROR?
428 001346 001004 BNE TST14-10 ;BR TO ERROR HALT ON SEQ ERROR
429
430 001350 012700 000000 MOV #0,R0 ;MOVE ZEROES TO R0
431 001354 005700 TST R0 ;SUCCESSFUL?
432 001356 001404 BEQ TST14
433 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
434 ; CONDITIONAL BRANCH INST. AND <===
435 ; REPLACE THE MOVE INSTRUCTION <===
436 ; WHICH FOLLOWS W/ 774 ***** <===
437 MOV #17,-(R2) ;MOVE TO MAILBOX # ***** 17 *****
438 INC -(R2) ;SET MSGTYP TO FATAL ERROR
439 HALT ;R0 NOT 0
440 ; OR SEQUENCE ERROR
441 ;*****
442 ;
443 ;TEST 14 TEST IF R0 CAN HOLD ONES AND ZEROES
444 ;*****
445 001370 005212 TST14: INC (R2) ;UPDATE TEST NUMBER
446 001372 022712 000014 CMP #14,(R2) ;SEQUENCE ERROR?
447 001376 001005 BNE TST15-10 ;BR TO ERROR HALT ON SEQ ERROR
448 001400 012700 125252 MOV #125252,R0 ;MOVE ALTERNATING ONES AND ZEROES TO R0
449 001404 020027 125252 CMP R0,#125252 ;SUCCESSFUL?
450 001410 001404 BEQ TST15

```



```
451 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
452 ; CONDITIONAL BRANCH INST. AND <====  
453 ; REPLACE THE MOVE INSTRUCTION <====  
454 ; WHICH FOLLOWS W/ 773 <====  
455 001412 012742 000020 MOV #20,(R2) ;MOVE TO MAILBOX # ***** 20 *****  
456 001416 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
457 001420 000000 HALT ;RO NOT 12525  
458 ; OR SEQUENCE ERROR  
459  
460 ;*****  
461 ;TEST 15 TEST IF R0 CAN HOLD ZEROES AND ONES  
462 ;*****  
463 001422 005212 TST15: INC (R2) ;UPDATE TEST NUMBER  
464 001424 022712 CMP #15,(R2) ;SEQUENCE ERROR?  
465 001430 001005 BNE #16-10 ;BR TO ERROR HALT ON SEQ ERROR  
466 001432 012700 MOV #052525,R0 ;MOVE ALTERNATING ZEROES AND ONES TO R0  
467 001436 020027 RO #052525 ;SUCCESSFUL?  
468 001442 001404 BEQ TST16 ;  
469 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
470 ; CONDITIONAL BRANCH INST. AND <====  
471 ; REPLACE THE MOVE INSTRUCTION <====  
472 ; WHICH FOLLOWS W/ 773 <====  
473 001444 012742 000021 MOV #21,(R2) ;MOVE TO MAILBOX # ***** 21 *****  
474 001450 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
475 001452 000000 HALT ;RO NOT 525  
476 ; OR SEQUENCE ERROR  
477  
478 ;*****  
479 ;TEST 16 TEST IF R0 CAN HOLD ALL ONES  
480 ;*****  
481 001454 005212 TST16: INC (R2) ;UPDATE TEST NUMBER  
482 001456 022712 CMP #16,(R2) ;SEQUENCE ERROR?  
483 001462 001005 BNE #17-10 ;BR TO ERROR HALT ON SEQ ERROR  
484 001464 012700 MOV #177777,R0 ;MOVE ALL ONES TO R0  
485 001470 020027 RO #177777 ;SUCCESSFUL?  
486 001474 001404 BEQ TST17 ;  
487 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
488 ; CONDITIONAL BRANCH INST. AND <====  
489 ; REPLACE THE MOVE INSTRUCTION <====  
490 ; WHICH FOLLOWS W/ 773 <====  
491 001476 012742 000022 MOV #22,(R2) ;MOVE TO MAILBOX # ***** 22 *****  
492 001502 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
493 001504 000000 HALT ;RO NOT 17777  
494 ; OR SEQUENCE ERROR  
495  
496 ;*****  
497 ;TEST 17 TEST IF R1 CAN HOLD A ONE IN ALL BITS  
498 ;*****  
499 001506 005212 TST17: INC (R2) ;UPDATE TEST NUMBER  
500 001510 022712 CMP #17,(R2) ;SEQUENCE ERROR?  
501 001514 001012 BNE #20-10 ;BR TO ERROR HALT ON SEQ ERROR  
502 001516 012701 MOV #1,R1 ;SET BIT 0  
503 001522 012700 MOV #21,R0 ;SET BIT COUNTER  
504 001526 000241 CLC ;CLEAR C-BIT  
505 001530 005200 REG1: INC R0 ;INCREMENT BIT COUNTER  
506 001532 001403 BEQ REG1E ;BR TO ERROR HALT IF BIT IS LOST
```

```
507 001534 006101 ROL R1 ;ROTATE 1 POSITION  
508 001536 013774 BCC REG1 ;ALL DONE  
509 001540 001404 BEQ TST20 ;  
510 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
511 ; CONDITIONAL BRANCH INST. AND <====  
512 ; REPLACE THE MOVE INSTRUCTION <====  
513 ; WHICH FOLLOWS W/ 766 <====  
514 001542 012742 000023 REG1E: MOV #23,(R2) ;MOVE TO MAILBOX # ***** 23 *****  
515 001544 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
516 001546 000000 HALT ;FAILURE WITH R1  
517 ; OR SEQUENCE ERROR  
518  
519 ;*****  
520 ;TEST 20 TEST IF R1 CAN HOLD A ZERO IN ALL BITS  
521 ;*****  
522 001552 005212 TST20: INC (R2) ;UPDATE TEST NUMBER  
523 001554 022712 CMP #20,(R2) ;SEQUENCE ERROR?  
524 001560 001014 BNE #21-10 ;BR TO ERROR HALT ON SEQ ERROR  
525 001562 012701 MOV #2,R1 ;SET ALL ONES IN R1 EXCEPT FOR BIT 0  
526 001566 012700 MOV #21,R0 ;SET BIT COUNTER  
527 001572 000261 SEC ;SET C-BIT  
528 001574 005200 REG1A: INC R0 ;INCREMENT COUNTER  
529 001576 001405 BEQ R1ERR ;BR TO ERROR HALT IF COUNTER=0  
530 001600 006101 ROL R1 ;ROTATE 1 POSITION  
531 001602 103774 BCS REG1A ;CONTINUE UNTIL C-BIT IS CLEAR  
532 001604 022701 CMP #1,R1 ;CHECK DATA IN R1  
533 001610 001404 BEQ TST21 ;  
534 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
535 ; CONDITIONAL BRANCH INST. AND <====  
536 ; REPLACE THE MOVE INSTRUCTION <====  
537 ; WHICH FOLLOWS W/ 764 <====  
538 001612 012742 000024 R1ERR: MOV #24,(R2) ;MOVE TO MAILBOX # ***** 24 *****  
539 001614 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
540 001620 000000 HALT ;FAILURE WITH R1  
541 ; OR SEQUENCE ERROR  
542 ;*****  
543 ;TEST 21 TEST IF R2 CAN HOLD A ONE IN ALL BITS  
544 ;*****  
545 001622 005212 TST21: INC (R2) ;UPDATE TEST NUMBER  
546 001624 022712 CMP #21,(R2) ;SEQUENCE ERROR?  
547 001630 001012 BNE REG2A-14 ;BR TO ERROR HALT ON SEQ ERROR  
548 001632 012702 MOV #1,R2 ;SET BIT 0  
549 001636 012700 MOV #21,R0 ;SET BIT COUNTER  
550 001642 000241 CLC ;CLEAR C-BIT  
551 001644 005200 REG2: RO REG2A-14 ;INCREMENT BIT COUNTER  
552 001646 001403 BEQ REG2 ;BR TO ERROR HALT IF BIT IS LOST  
553 001650 006102 ROL R2 ;ROTATE 1 POSITION  
554 001652 103374 BCC REG2 ;ALL DONE  
555 001654 001406 BEQ REG2A ;  
556 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
557 ; BRANCH INSTRUCTION AND <====  
558 ; REPLACE THE MOVE INSTRUCTION <====  
559 ; FOLLOWING W/ 771 <====  
560 001656 012702 000304 MOV #TESTN,R2 ;RESTORE POINTER
```

```

CPKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 14
CPKAAC-P11 18-OCT-78 11:01 T21 TEST IF R2 CAN HOLD A ONE IN ALL BITS SEQ 0026

563 001662 012742 000025 MOV #25,-(R2) ;MOVE TO MAILBOX # ***** 25 *****
564 001666 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
565 001670 000000 HALT ;FAILURE WITH R2
566 001672 012702 000304 REG2A: MOV #STESTN,R2 ;RESTORE POINTER
*****
;TEST 22 TEST IF R2 CAN HOLD A ZERO IN ALL BITS
*****
TST22: INC (R2) ;UPDATE TEST NUMBER
CMP #22,(R2) ;SEQUENCE ERROR?
BNE TST23-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #2,R2 ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
MOV #21,R0 ;SET BIT COUNTER
SEC ;SET C-BIT
REG2B: INC R0 ;INCREMENT BIT COUNTER
BEQ R2ERR ;BR TO ERROR HALT IF COUNTER=0
ROL R2 ;ROTATE 1 POSITION
BCS REG2B ;CONTINUE UNTIL C-BIT IS CLEAR
BEQ REG2C ;CHECK DATA IN R2
MOV #STESTN,R2 ;RESTORE POINTER
R2ERR: MOV #26,-(R2) ;MOVE TO MAILBOX # ***** 26 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R2
REG2C: MOV #STESTN,R2 ;RESTORE POINTER
*****
;TEST 23 TEST IF R3 CAN HOLD A ONE IN ALL BITS
*****
TST23: INC (R2) ;UPDATE TEST NUMBER
CMP #23,(R2) ;SEQUENCE ERROR?
BNE TST24-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #1,R3 ;SET BIT 0
MOV #21,R0 ;SET BIT COUNTER
CLC ;CLEAR C-BIT
REG3: INC R0 ;INCREMENT BIT COUNTER
BEQ REG3E ;BR TO ERROR HALT IF BIT IS LOST
ROL R3 ;ROTATE 1 POSITION
BCS REG3 ;CHECK DATA
BEQ TST24 ;ALL DONE
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
REG3E: MOV #27,-(R2) ;MOVE TO MAILBOX # ***** 27 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R3
; OR SEQUENCE ERROR
*****
;TEST 24 TEST IF R3 CAN HOLD A ZERO IN ALL BITS
*****
TST24: INC (R2) ;UPDATE TEST NUMBER
CMP #24,(R2) ;SEQUENCE ERROR?

```

```

CPKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 15
CPKAAC-P11 18-OCT-78 11:01 T24 TEST IF R3 CAN HOLD A ZERO IN ALL BITS SEQ 0027

619 002030 001014 BNE TST25-10 ;BR TO ERROR HALT ON SEQ ERROR
620 002032 012703 177776 MOV #21,R0 ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
621 002036 012700 177757 MOV #21,R0 ;SET BIT COUNTER
622 002040 000261 SEC ;SET C-BIT
623 002044 005200 REG3A: INC R0 ;INCREMENT BIT COUNTER
624 002046 001405 BEQ R3ERR ;BR TO ERROR HALT IF COUNTER=0
625 002050 006103 ROL R3 ;ROTATE 1 POSITION
626 002052 103774 BCS REG3A ;CONTINUE UNTIL C-BIT IS CLEAR
627 002054 022703 177777 CMP #R3 ;CHECK DATA
628 002060 001404 BEQ TST25 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
R3ERR: MOV #30,-(R2) ;MOVE TO MAILBOX # ***** 30 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R3
; OR SEQUENCE ERROR
*****
;TEST 25 TEST IF R4 CAN HOLD A ONE IN ALL BITS
*****
TST25: INC (R2) ;UPDATE TEST NUMBER
CMP #25,(R2) ;SEQUENCE ERROR?
BNE TST26-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #1,R4 ;SET BIT 0
MOV #21,R0 ;SET BIT COUNTER
CLC ;CLEAR C-BIT
REG4: INC R0 ;INCREMENT BIT COUNTER
BEQ REG4E ;BR TO ERROR HALT IF BIT IS LOST
ROL R4 ;ROTATE 1 POSITION
BCS REG4 ;CHECK DATA
BEQ TST26 ;ALL DONE
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
REG4E: MOV #31,-(R2) ;MOVE TO MAILBOX # ***** 31 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R4
; OR SEQUENCE ERROR
*****
;TEST 26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
*****
TST26: INC (R2) ;UPDATE TEST NUMBER
CMP #26,(R2) ;SEQUENCE ERROR?
BNE TST27-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #2,R4 ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
MOV #21,R0 ;SET BIT COUNTER
SEC ;SET C-BIT
REG4A: INC R0 ;INCREMENT BIT COUNTER
BEQ R4ERR ;BR TO ERROR HALT IF COUNTER=0
ROL R4 ;ROTATE 1 POSITION

```

```

CPKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 16
CPKAAC.P11 18-OCT-78 11:01 T26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS SEQ 0028

675 002166 193774 177777
676 002176 012742 000032
677 002174 061404
678
679
680
681
682
683 002176 012742 000032
684 002176 012742 000032
685 002202 005242
686 002204 000000
687
688
689
690
691
692 002206 005212 000027
693 002210 022712
694 002214 001012
695 002216 012705 000001
696 002222 012700 177757
697 002226 000241
698 002230 005200
699 002234 006105
700 002236 103374
701 002238 001404
702
703
704
705
706
707 002242
708 002242 012742 000033
709 002246 005242
710 002250 000000
711
712
713
714
715
716 002252 005212 000030
717 002254 022712
718 002260 001014
719 002262 012705 177776
720 002266 012700 177757
721
722 002274 005200
723 002276 001405
724 002300 006105
725 002302 103374
726 002304 022705 177777
727 002310 001404
728
729
730

;*****
;TEST 27 TEST IF R5 CAN HOLD A ONE IN ALL BITS
;*****
TST27: INC (R2) ;UPDATE TEST NUMBER
CMP #27,(R2) ;SEQUENCE ERROR?
BNE #TST30-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #1,R5 ;SET BIT 0
MOV #-21,R0 ;SET BIT COUNTER
CLC ;CLEAR C-BIT
REG5: INC REG5E ;INCREMENT BIT COUNTER
BEQ REG5E ;BR TO ERROR HALT IF BIT IS LOST
ROL R5 ;ROTATE 1 POSITION
BCC REG5 ;ALL DONE
BEQ TST30

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

REG5E: MOV #33,-(R2) ;MOVE TO MAILBOX # ***** 33 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R5
; OR SEQUENCE ERROR

;*****
;TEST 30 TEST IF R5 CAN HOLD A ZERO IN ALL BITS
;*****
TST30: INC (R2) ;UPDATE TEST NUMBER
CMP #30,(R2) ;SEQUENCE ERROR?
BNE #TST31-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #2,R5 ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
MOV #-21,R0 ;SET BIT COUNTER
SEC ;SET C-BIT
REG5A: INC R0 ;INCREMENT BIT COUNT
BEQ R5ERR ;BR TO ERROR HALT IF COUNTER=0
ROL R5 ;ROTATE 1 POSITION
BCS REG5A ;CONTINUE UNTIL C-BIT IS CLEAR
CMP #1,R5 ;CHECK DATA
BEQ #TST31

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

```

```

CPKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 17
CPKAAC.P11 18-OCT-78 11:01 T30 TEST IF R5 CAN HOLD A ZERO IN ALL BITS SEQ 0029

731
732
733 002312 012742 000034
734 002316 005242
735 002320 000000
736
737
738
739
740
741 002322 005212 000031
742 002324 022712
743 002330 001012
744 002332 012706 000001
745 002336 012700 177757
746 002342 000241
747 002344 005200
748 002346 001403
749 002350 006106
750 002352 103374
751 002354 001404
752
753
754
755
756 002356 012742 000035
757 002356 005242
758 002362 000000
759
760
761
762
763
764
765 002366 005212 000032
766 002370 022712
767 002374 001014 177776
768 002376 012706 177757
769 002402 012700
770 002406 000261
771 002410 005200
772 002412 001405
773 002414 006106
774 002416 103374 177777
775 002420 022706
776 002424 001404
777
778
779
780
781 002426 012742 000036
782 002426 005242
783 002432 000000
784 002434
785

;*****
;TEST 29 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
;*****
R5ERR: MOV #34,-(R2) ;MOVE TO MAILBOX # ***** 34 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R5
; OR SEQUENCE ERROR

;*****
;TEST 31 TEST IF R6 CAN HOLD A ONE IN ALL BITS
;*****
TST31: INC (R2) ;UPDATE TEST NUMBER
CMP #31,(R2) ;SEQUENCE ERROR?
BNE #TST32-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #1,R6 ;SET BIT 0
MOV #-21,R0 ;SET BIT COUNTER
CLC ;CLEAR C-BIT
REG6: INC REG6E ;INCREMENT BIT COUNTER
BEQ REG6E ;BR TO ERROR HALT IF BIT IS LOST
ROL R6 ;ROTATE 1 POSITION
BCC REG6 ;ALL DONE
BEQ TST32

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

REG6E: MOV #35,-(R2) ;MOVE TO MAILBOX # ***** 35 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R6
; OR SEQUENCE ERROR

;*****
;TEST 32 TEST IF R6 CAN HOLD A ZERO IN ALL BITS
;*****
TST32: INC (R2) ;UPDATE TEST NUMBER
CMP #32,(R2) ;SEQUENCE ERROR?
BNE #TST33-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #2,R6 ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
MOV #-21,R0 ;SET BIT COUNTER
SEC ;SET C-BIT
REG6A: INC R0 ;INCREMENT BIT COUNT
BEQ R6ERR ;BR TO ERROR HALT IF COUNTER=0
ROL R6 ;ROTATE 1 POSITION
BCS REG6A ;CONTINUE UNTIL C-BIT IS CLEAR
CMP #1,R6 ;CHECK DATA
BEQ #TST33

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

R6ERR: MOV #36,-(R2) ;MOVE TO MAILBOX # ***** 36 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;FAILURE WITH R6
; OR SEQUENCE ERROR

```

```
786 ;*****  
787 ;SBTTL PSW TESTS  
788 ;  
789 ; THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA  
790 ; PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE  
791 ; PSW ADDRESSING LOGIC IS FUNCTIONING. MOVE AND COMPARE INSTRUCTIONS  
792 ; ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.  
793 ; EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR  
794 ; SCOPING.  
795 ; THE PSW REGISTER ITSELF IS TESTED AS WELL AS THE ADDRESS  
796 ; SELECT CIRCUITRY. THE AMUX INPUTS TO THE PSW MUX ARE TESTED. THE  
797 ; CC INPUTS ARE TESTED LATER IN THE MICROCODE TESTS. SETTING OF  
798 ; THE T-BIT BY THE TEST PATTERNS IS PURPOSELY AVOIDED; TESTING OF THE  
799 ; T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.  
800 ;*****  
801 ;TEST 33 TEST IF PSW WILL HOLD ZEROES  
802 ;*****  
803 TST33: INC (R2) ;UPDATE TEST NUMBER  
804 CMP #33,(R2) ;SEQUENCE ERROR?  
805 BNE TST34-10 ;BR TO ERROR HALT ON SEQ ERROR  
806 MOV #STBOT,R6 ;  
807 MOV #0,#PS ;SET PSW TO ZERO  
808 TST #0 ;SUCCESSFUL  
809 BEQ TST34 ;  
810 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
811 ; CONDITIONAL BRANCH INST. AND <====  
812 ; REPLACE THE MOVE INSTRUCTION <====  
813 ; WHICH FOLLOWS W/ 770 <====  
814 ;  
815 MOV #37,(R2) ;MOVE TO MAILBOX # ***** 37 *****  
816 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
817 HALT ;PSW NOT 0  
818 ; OR SEQUENCE ERROR  
819 ;*****  
820 ;TEST 34 TEST IF PSW WILL HOLD ONES AND ZEROES  
821 ;*****  
822 TST34: INC (R2) ;UPDATE TEST NUMBER  
823 CMP #34,(R2) ;SEQUENCE ERROR?  
824 BNE TST35-10 ;BR TO ERROR HALT ON SEQ ERROR  
825 MOV #105,#PS ;MOVE ALT. ONES AND ZEROES TO PSW  
826 CMP #105,#PS ;SUCCESSFUL?  
827 BEQ TST35 ;  
828 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
829 ; CONDITIONAL BRANCH INST. AND <====  
830 ; REPLACE THE MOVE INSTRUCTION <====  
831 ; WHICH FOLLOWS W/ 771 <====  
832 ;  
833 MOV #40,(R2) ;MOVE TO MAILBOX # ***** 40 *****  
834 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
835 HALT ;PSW NOT 252  
836 ; OR SEQUENCE ERROR  
837
```

```
838 ;*****  
839 ;TEST 35 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES  
840 ;*****  
841 TST35: INC (R2) ;UPDATE TEST NUMBER  
842 CMP #35,(R2) ;SEQUENCE ERROR?  
843 BNE TST36-10 ;BR TO ERROR HALT ON SEQ ERROR  
844 MOV #105,#PS ;MOVE ALT. ONES AND ZEROES TO PSW  
845 CMP #105,#PS ;SUCCESSFUL?  
846 BEQ TST36 ;  
847 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
848 ; CONDITIONAL BRANCH INST. AND <====  
849 ; REPLACE THE MOVE INSTRUCTION <====  
850 ; WHICH FOLLOWS W/ 771 <====  
851 ;  
852 MOV #41,(R2) ;MOVE TO MAILBOX # ***** 41 *****  
853 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
854 HALT ;PSW NOT 105  
855 ; OR SEQUENCE ERROR  
856 ;*****  
857 ;TEST 36 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES  
858 ;*****  
859 TST36: INC (R2) ;UPDATE TEST NUMBER  
860 CMP #36,(R2) ;SEQUENCE ERROR?  
861 BNE TST37-10 ;BR TO ERROR HALT ON SEQ ERROR  
862 MOV #357,#PS ;MOVE ONES TO PSW  
863 CMP #357,#PS ;SUCCESSFUL  
864 BEQ TST37 ;  
865 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
866 ; CONDITIONAL BRANCH INST. AND <====  
867 ; REPLACE THE MOVE INSTRUCTION <====  
868 ; WHICH FOLLOWS W/ 771 <====  
869 ;  
870 MOV #42,(R2) ;MOVE TO MAILBOX # ***** 42 *****  
871 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
872 HALT ;PSW NOT 357  
873 ; OR SEQUENCE ERROR
```

```
.SBTTL CONDITION CODE TEST
;*****
; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.
; THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
; BEQ AND BNE ARE TESTED FOR PROPER EXECUTION. THEN THE Z-BIT IS
; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
; AGAIN FOR PROPER OPERATION.
; THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
; CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
; BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
; LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
; USED IN THE TEST ARE VERIFIED HERE.
;*****
;TEST 37 TEST BRANCHES AROUND Z-BIT
;*****
TST37: INC (R2) ;UPDATE TEST NUMBER
        CMP #37,(R2) ;SEQUENCE ERROR?
        BNE TST40-10 ;BR TO ERROR HALT ON SEQ ERROR
        ;FIRST WITH Z-BIT ON
        CCC ;CC=0100: JUST Z-BIT
        SEZ ;CHECK OPPOSITE CONDITION
        BNE BRZ1
        BEQ BRZ2
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ; CONDITIONAL BRANCH INST. AND <====
        ; REPLACE THE MOVE INSTRUCTION <====
        ; WHICH FOLLOWS W/ 774 <====
BRZ1: MOV #43,-(R2) ;MOVE TO MAILBOX # ***** 43 *****
        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 TST40
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ; CONDITIONAL BRANCH INST. AND <====
        ; REPLACE THE MOVE INSTRUCTION <====
        ; WHICH FOLLOWS W/ 764 <====
BRZ3: MOV #44,-(R2) ;MOVE TO MAILBOX # ***** 44 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;IMPROPER BR W/ Z=0
        ; OR SEQUENCE ERROR
```

```
;*****
; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
; THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
; BMI AND BPL ARE TESTED FOR PROPER EXECUTION. THEN THE N-BIT IS
; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
; AGAIN FOR PROPER OPERATION.
; THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
; CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
; BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
; LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
; USED IN THE TEST ARE VERIFIED HERE.
;*****
;TEST 40 TEST BRANCHES AROUND N-BIT
;*****
TST40: INC (R2) ;UPDATE TEST NUMBER
        CMP #40,(R2) ;SEQUENCE ERROR?
        BNE TST41-10 ;BR TO ERROR HALT ON SEQ ERROR
        ;FIRST WITH N-BIT ON
        CCC ;CC=1000: JUST N-BIT
        SEN ;CHECK OPPOSITE CONDITION
        BPL BRN1
        BMI BRN2
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ; CONDITIONAL BRANCH INST. AND <====
        ; REPLACE THE MOVE INSTRUCTION <====
        ; WHICH FOLLOWS W/ 774 <====
BRN1: MOV #45,-(R2) ;MOVE TO MAILBOX # ***** 45 *****
        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
        BPL TST41
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ; CONDITIONAL BRANCH INST. AND <====
        ; REPLACE THE MOVE INSTRUCTION <====
        ; WHICH FOLLOWS W/ 764 <====
BRN3: MOV #46,-(R2) ;MOVE TO MAILBOX # ***** 46 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;IMPROPER BR W/ N=0
        ; OR SEQUENCE ERROR
```

```
969 ;*****  
970 ;  
971 ; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.  
972 ; THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS  
973 ; BVS AND BVC ARE TESTED FOR PROPER EXECUTION. THEN THE V-BIT IS  
974 ; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED  
975 ; AGAIN FOR PROPER OPERATION.  
976 ; THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION  
977 ; CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL  
978 ; BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR  
979 ; LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY  
980 ; USED IN THE TEST ARE VERIFIED HERE.  
981 ;  
982 ;*****  
983 ;  
984 ; TEST 41 TEST BRANCHES AROUND V-BIT  
985 ;*****  
986 TST41: INC (R2) ;UPDATE TEST NUMBER  
987 CMP #41,(R2) ;SEQUENCE ERROR?  
988 BNE TST42-10 ;BR TO ERROR HALT ON SEQ ERROR  
989 ;FIRST WITH V-BIT ON  
990 CCC ;CC=0010: JUST V-BIT  
991 SEC ;  
992 BVC BRV1 ;CHECK OPPOSITE CONDITION  
993 BVS BRV2 ;  
994 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
995 ; CONDITIONAL BRANCH INST. AND <====  
996 ; REPLACE THE MOVE INSTRUCTION <====  
997 ; WHICH FOLLOWS W/ 774 <====  
998 ;  
999 BRV1: MOV #47,-(R2) ;MOVE TO MAILBOX # ***** 47 *****  
1000 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1001 HALT ;IMPROPER BR W/ V=1  
1002 ;CHECK WITH V-BIT OFF  
1003 BRV2: SCC ;CC=1101: ALL BVT V-BIT  
1004 CLV ;  
1005 BVC BRV3 ;CHECK OPPOSITE CONDITION  
1006 BVS TST42 ;  
1007 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1008 ; CONDITIONAL BRANCH INST. AND <====  
1009 ; REPLACE THE MOVE INSTRUCTION <====  
1010 ; WHICH FOLLOWS W/ 764 <====  
1011 ;  
1012 BRV3: MOV #50,-(R2) ;MOVE TO MAILBOX # ***** 50 *****  
1013 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1014 HALT ;IMPROPER BR W/ V=0  
1015 ; OR SEQUENCE ERROR
```

```
1016 ;*****  
1017 ;  
1018 ; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.  
1019 ; THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS  
1020 ; BCS AND BCC ARE TESTED FOR PROPER EXECUTION. THEN THE C-BIT IS  
1021 ; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED  
1022 ; AGAIN FOR PROPER OPERATION.  
1023 ; THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION  
1024 ; CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL  
1025 ; BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR  
1026 ; LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY  
1027 ; USED IN THE TEST ARE VERIFIED HERE.  
1028 ;  
1029 ;*****  
1030 ;  
1031 ; TEST 42 TEST BRANCHES AROUND C-BIT  
1032 ;*****  
1033 TST42: INC (R2) ;UPDATE TEST NUMBER  
1034 CMP #42,(R2) ;SEQUENCE ERROR?  
1035 BNE TST43-10 ;BR TO ERROR HALT ON SEQ ERROR  
1036 ;FIRST WITH C-BIT ON  
1037 CCC ;CC=0001: JUST C-BIT  
1038 SEC ;  
1039 BCC BRC1 ;CHECK OPPOSITE CONDITION  
1040 BCS BRC2 ;  
1041 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1042 ; CONDITIONAL BRANCH INST. AND <====  
1043 ; REPLACE THE MOVE INSTRUCTION <====  
1044 ; WHICH FOLLOWS W/ 774 <====  
1045 ;  
1046 BRC1: MOV #51,-(R2) ;MOVE TO MAILBOX # ***** 51 *****  
1047 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1048 HALT ;IMPROPER BR W/ C=1  
1049 ;CHECK WITH C-BIT OFF  
1050 BRC2: SCC ;CC=1110  
1051 CLC ;  
1052 BCI BRC3 ;CHECK OPPOSITE CONDITION  
1053 BMI TST43 ;  
1054 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1055 ; CONDITIONAL BRANCH INST. AND <====  
1056 ; REPLACE THE MOVE INSTRUCTION <====  
1057 ; WHICH FOLLOWS W/ 764 <====  
1058 ;  
1059 BRC3: MOV #52,-(R2) ;MOVE TO MAILBOX # ***** 52 *****  
1060 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1061 HALT ;IMPROPER BR W/ C=0  
1062 ; OR SEQUENCE ERROR
```

```

1063 ;*****
1064 ;SBTTL MICROCODE TESTS
1065 ;
1066 ; THE MICROCODE TESTS ARE USED TO VERIFY THE MICROPROGRAMM
1067 ; FLOW. THE GOAL OF THESE TESTS IS TO EXERCISE EVERY POSSIBLE
1068 ; BRANCH IN THE MICROPROGRAM FLOW.
1069 ; THE TEST EXERCISES EVERY BRANCH IN THE MICROCODE BY
1070 ; TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
1071 ; ALL POSSIBLE MODES. FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
1072 ; AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
1073 ; ADDRESSING MODES. BYTE MODES ARE ALSO TESTED, AS EACH NEW
1074 ; MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
1075 ; A SMALL LOOP CONVENIENT FOR SCOPING. THE TEST IS SET UP USING
1076 ; ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
1077 ; VERIFIED.
1078 ; IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
1079 ; FAULT.
1080 ;*****
1081
1082 ;*****
1083 ;
1084 ; THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
1085 ; MODE WITH THE SINGLE OPERAND INSTRUCTION. FOLLOWING THE SEQUENCE CHECK,
1086 ; THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
1087 ; CHECKS THAT THE Z-BIT WAS PROPERLY SET. THIS SHALL TEST IS SELF-SUFFICIENT
1088 ; AND CAN BE SCOPED TO TROUBLE SHOOT ALL OF THE IR DECODE LOGIC AND
1089 ; MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0. FOLLOWING THIS TEST
1090 ; SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0. THESE
1091 ; INSTRUCTIONS MAINPOLATE DATA AND SERVE TO CHECK THE DATA RESULTS
1092 ; OF THE SOP INSTRUCTIONS IN THIS TEST. THE DATA IN THIS TEST IS
1093 ; OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
1094 ;*****
1095 ;*****
1096 ;TEST 43 TEST MODE 0 USING SOP INST.
1097 ;*****
1098 ;*****
1099 ;*****
1100 ;*****
1101 ;*****
1102 003070 005212 000043 ;TST43: INC (R2) ;UPDATE TEST NUMBER
1103 003072 022712 ;CMP #43,(R2) ;SEQUENCE ERROR?
1104 003074 001020 ;BNE TST44-10 ;BR TO ERROR HALT ON SEQ ERROR
1105 003100 005000 ;CLR RO ;INITIALIZE
1106 003102 001404 ;BEQ SOPOA ;TRY THE CLEAR INST.
1107 ;
1108 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1109 ; CONDITIONAL BRANCH INST. AND <====
1110 ; REPLACE THE MOVE INSTRUCTION <====
1111 ; WHICH FOLLOWS W/ 716 <====
1112 003104 012742 000053 ;MOV #53,-(R2) ;MOVE TO MAILBOX # ***** 53 *****
1113 003110 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
1114 003112 000000 ;HALT ;CLR DID NOT SET Z-BIT
1115 003114 005200 ;SOP0A: INC RO ;TRY THE INCREMENT INST.
1116 003116 005100 ;COM RO ;TRY COMPLEMENT
1117 003118 005200 ;DEC RO
1118 003122 100404 ;BMI SOP0B ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

```

1119 ;
1120 ;
1121 ;
1122 003124 012742 000054 ;MOV #54,-(R2) ;MOVE TO MAILBOX # ***** 54 *****
1123 003130 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
1124 003132 000000 ;HALT ;NEGATE DID NOT SET N-BIT
1125 003134 005100 ;SOP0B: COM RO ;TRY COMPLEMENT INST.
1126 003136 001404 ;BEQ TST44 ;
1127 ;
1128 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1129 ; CONDITIONAL BRANCH INST. AND <====
1130 ; REPLACE THE MOVE INSTRUCTION <====
1131 ; WHICH FOLLOWS W/ 716 <====
1132 003140 012742 000055 ;MOV #55,-(R2) ;MOVE TO MAILBOX # ***** 55 *****
1133 003144 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
1134 003146 000000 ;HALT ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
1135 ;
1136 ;
1137 ;
1138 ;
1139 ;
1140 ; THIS TEST INTRODUCES THE REMAINING SOP INSTRUCTIONS AND TESTS
1141 ; THEM IN MODE 0. THE PURPOSE IS TO PROVIDE A BASELINE OF
1142 ; INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS. SINCE THE MICROCODE FOR
1143 ; THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
1144 ; SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
1145 ; FUNCTIONING.
1146 ;*****
1147 ;*****
1148 ;*****
1149 ;TEST 44 TEST REMAINDER OF SOP INSTS IN MODE 0
1150 ;*****
1151 ;*****
1152 ;*****
1153 ;*****
1154 ;*****
1155 ;*****
1156 ;*****
1157 ;*****
1158 ;*****
1159 ;*****
1160 003150 005212 000044 ;TST44: INC (R2) ;UPDATE TEST NUMBER
1161 003152 022712 ;CMP #44,(R2) ;SEQUENCE ERROR?
1162 003156 001021 ;BNE TST45-10 ;BR TO ERROR HALT ON SEQ ERROR
1163 003160 005000 ;CLR RO ;INITIALIZE
1164 003162 005300 ;DEC RO ;TRY DECREMENT INST.
1165 003164 100404 ;BMI SOP0C ;
1166 ;
1167 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1168 ; CONDITIONAL BRANCH INST. AND <====
1169 ; REPLACE THE MOVE INSTRUCTION <====
1170 ; WHICH FOLLOWS W/ 716 <====
1171 003166 012742 000056 ;MOV #56,-(R2) ;MOVE TO MAILBOX # ***** 56 *****
1172 003172 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
1173 003174 000000 ;HALT ;N-BIT NOT SET ON DEC
1174 003176 000261 ;SOP0C: SEC ;INITIALIZE CARRY
1175 003200 005500 ;ADC RO ;TRY ADD CARRY INST
1176 003202 001007 ;BNE SOP0D ;
1177 003204 000261 ;SBC RO ;INITIALIZE CARRY
1178 003206 005600 ;BPL SOP0D ;TRY SUBTRACT-CARRY INST
1179 003210 100004 ;COM RO
1180 003212 005100 ;INC RO
1181 003214 005200 ;DEC RO
1182 003218 005300 ;BEQ TST45 ;
1183 003220 001404 ;
1184 ;
1185 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1186 ; CONDITIONAL BRANCH INST. AND <====
1187 ; REPLACE THE MOVE INSTRUCTION <====

```

```
1175                                     )          WHICH FOLLOWS W/ 757          <====  
1176 003222                                     )  
1177 003222 012742 000057  SOP0D:  MOV #57,-(R2) ;MOVE TO MAILBOX # ***** 57 *****  
1178 003226 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1179 003230 000000          HALT ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F  
1180                                     ; OR SEQUENCE ERROR
```

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



```
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228 003276 005212  
1229 003300 022712 000046  
1230 003304 001014  
1231 003306 005000  
1232 003310 005010  
1233 003312 001404  
1234  
1235  
1236  
1237  
1238 003314 012742 000062  
1239 003320 005242  
1240 003322 000000  
1241 003324 005310  
1242 003326 100003  
1243 003330 000261  
1244 003332 005510  
1245 003334 001404  
1246  
1247  
1248  
1249  
1250 003336  
1251 003336 012742 000063  
1252 003342 005242  
1253 003344 000000  
1254  
1255
```

```
*****  
; THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST  
; SINGLE OPERAND MODE 1 INSTRUCTIONS- AGAIN, THE CLR INSTRUCTION  
; IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER  
; CONDITION CODES ARE SET. OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE  
; COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.  
*****  
; TEST 46 TEST MODE 1 USING SOP INST.  
; *****  
TST46: INC (R2) ;UPDATE TEST NUMBER  
; CMP #46,(R2) ;SEQUENCE ERROR? <=====  
; BNE TST47-10 ;BR TO ERROR HALT ON SEQ ERROR <=====  
; CLR R0 ;INITIALIZE R0 <=====  
; CLR (R0) ;TRY CLEAR INST W/MODE 1 <=====  
; BEQ SOP1A  
; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; ; CONDITIONAL BRANCH INST. AND <=====  
; ; REPLACE THE MOVE INSTRUCTION <=====  
; ; WHICH FOLLOWS W/ 775 ***** <=====  
; ; MOVE TO MAILBOX # ***** 62 *****  
SOP1A: MOV #62,-(R2) ;MOVE TO MAILBOX # ***** 62 *****  
; INC -(R2) ;SET MSGTYP TO FATAL ERROR  
; HALT ;CLR DID NOT SET Z-BIT  
; DEC (R0) ;TRY DECREMENT INST W/MODE 1  
; BPL SOP1B  
; SEC ;INITIALIZE CARRY  
; ADC (R0) ;TRY ADD-CARRY W/MODE 1  
; BEQ TST47  
; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; ; CONDITIONAL BRANCH INST. AND <=====  
; ; REPLACE THE MOVE INSTRUCTION <=====  
; ; WHICH FOLLOWS W/ 764 <=====  
SOP1B: MOV #63,-(R2) ;MOVE TO MAILBOX # ***** 63 *****  
; INC -(R2) ;SET MSGTYP TO FATAL ERROR  
; HALT ;TEST CUMULATIVE RESULT OF ABOVE INST  
; ; OR SEQUENCE ERROR
```

```
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266 003346 005212  
1267 003350 022712 000047  
1268 003354 001020  
1269 003356 005000  
1270 003360 005010  
1271 003362 005110  
1272 003364 105010  
1273 003366 001404  
1274  
1275  
1276  
1277  
1278 003370 012742 000064  
1279 003374 005242  
1280 003376 000000  
1281 003400 005210  
1282 003402 100005  
1283 003404 105110  
1284 003406 105210  
1285 003410 100002  
1286 003412 105210  
1287 003414 001404  
1288  
1289  
1290  
1291  
1292 003416  
1293 003416 012742 000065  
1294 003422 005242  
1295 003424 000000  
1296  
1297
```

```
*****  
; THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1  
; SINGLE OPERAND INSTRUCTIONS.  
; THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED  
; AND VERIFIED.  
*****  
; TEST 47 TEST MODE 1 EVEN BYTE USING SOP INST  
; *****  
TST47: INC (R2) ;UPDATE TEST NUMBER  
; CMP #47,(R2) ;SEQUENCE ERROR? <=====  
; BNE TST50-10 ;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 ***** <=====  
; ; MOVE TO MAILBOX # ***** 64 *****  
SOPB1A: MOV #64,-(R2) ;MOVE TO MAILBOX # ***** 64 *****  
; INC -(R2) ;SET MSGTYP TO FATAL ERROR  
; HALT ;CLRB DID NOT SET Z-BIT  
; DEC (R0) ;INCREMENT TO TEST WORD  
; BPL SOPB1B  
; COMB (R0) ;COMPLEMENT: ODD BYTE = 376  
; COMB (R0) ;INC: ODD BYTE = 377  
; INCB (R0) ;INCREMENT ODD BYTE=0  
; BPL SOPB1B  
; INCB (R0)  
; BEQ TST50  
; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; ; CONDITIONAL BRANCH INST. AND <=====  
; ; REPLACE THE MOVE INSTRUCTION <=====  
; ; WHICH FOLLOWS W/ 760 <=====  
SOPB1B: MOV #65,-(R2) ;MOVE TO MAILBOX # ***** 65 *****  
; INC -(R2) ;SET MSGTYP TO FATAL ERROR  
; HALT ;CHECK CUMULATIVE RESULT OF ABOVE INST  
; ; OR SEQUENCE ERROR
```

```
1298 ;*****\ *****  
1299 ;  
1300 ; THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL  
1301 ; FUNCTION CORRECTLY FOR ODD BYTES.  
1302 ; THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN  
1303 ; EXERCISED. CHECKS ARE MADE THAT THE PROPER BYTE IS MODIFIED AND  
1304 ; THE CONDITION CODES ARE CHECKED. IT IS ALSO VERIFIED THAT THE UNADDRESSED  
1305 ; BYTE IS NOT ALTERED BY THE INSTRUCTION.  
1306 ;*****\ *****  
1307 ;  
1308 ;*****\ *****  
1309 ; TEST 50 TEST MODE 1 ODD BYTE USING SOP INST  
1310 ;*****\ *****  
1311 003426 005212 000050 TST50: INC (R2) ;UPDATE TEST NUMBER  
1312 003430 022712 ;CMP #50(R2) ;SEQUENCE ERROR?  
1313 003434 001022 ;BNE TST51-10 ;BR TO ERROR HALT ON SEQ ERROR  
1314 003436 005000 ;CLR R0 ;INITIALIZE R0  
1315 003440 005010 ;CLR (R0) ;INITIALIZE LOC. 0  
1316 003442 005110 ;COM (R0)  
1317 003444 005200 ;INC R0 ;R0=ODD BYTE  
1318 003446 105010 ;CLR (R0) ;TRY TO CLEAR BYTE 1  
1319 003450 001404 ;BEQ SOPB1C  
1320 ;  
1321 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1322 ; CONDITIONAL BRANCH INST. AND <=====  
1323 ; REPLACE THE MOVE INSTRUCTION <=====  
1324 ; WHICH FOLLOWS W/ 72 <=====  
1325 003452 012742 000066 ;MOV #66-(R2) ;MOVE TO MAILBOX # ***** 66 *****  
1326 003456 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1327 003460 000000 ;HALT ;CLR DID NOT SET Z-BIT  
1328 003462 005300 ;SOPB1C: DEC R0 ;R0=WORD ADDR.  
1329 003464 005210 ;INC (R0) ;INCREMENT TO TEST WORD  
1330 003466 005200 ;INC R0 ;R0=ODD BYTE  
1331 003470 105110 ;COMB (R0) ;TRY TO COMPLEMENT BYTE 1  
1332 003472 105210 ;INCB (R0)  
1333 003474 100002 ;BPL SOPB1D  
1334 003476 105210 ;INCB (R0)  
1335 003500 001404 ;BEQ TST51 ;TRY TO INCREMENT BYTE 1  
1336 ;  
1337 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1338 ; CONDITIONAL BRANCH INST. AND <=====  
1339 ; REPLACE THE MOVE INSTRUCTION <=====  
1340 ; WHICH FOLLOWS W/ 756 <=====  
1341 003502 012742 000067 ;SOPB1D: MOV #67-(R2) ;MOVE TO MAILBOX # ***** 67 *****  
1342 003506 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1343 003510 000000 ;HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.  
; OR SEQUENCE ERROR
```

```
1344 ;*****\ *****  
1345 ;  
1346 ; THIS TEST VERIFIES MODE 2 SINGLE-OPERAND INSTRUCTIONS. PREVIOUSLY  
1347 ; TESTED INSTRUCTIONS ARE USED TO SET A POINTER IN R0 TO LOC. 400.  
1348 ; LOC. 400 IS INITIALIZED TO -1 BEFORE A CLR MODE 2 IS EXECUTED.  
1349 ; THEN R0 IS DECREMENTED BY TWO TO AGAIN POINT TO 400 BEFORE EACH  
1350 ; OF SEVERAL MODE 2 INSTRUCTIONS ARE USED TO VERIFY THE DATA RESULTS OF  
1351 ; THE TEST. THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE  
1352 ; REGISTER.  
1353 ;*****\ *****  
1354 ;  
1355 ;*****\ *****  
1356 ; TEST 51 TEST MODE 2 USING SOP INST  
1357 ;*****\ *****  
1358 003512 005212 000051 TST51: INC (R2) ;UPDATE TEST NUMBER  
1359 003514 022712 ;CMP #51(R2) ;SEQUENCE ERROR?  
1360 003520 001022 ;BNE TST52-10 ;BR TO ERROR HALT ON SEQ ERROR  
1361 003522 005000 ;CLR R0 ;SET R0=400  
1362 003524 105100 ;COMB R0  
1363 003526 005200 ;INC R0  
1364 003530 005010 ;CLR (R0)  
1365 003532 005110 ;COM (R0)  
1366 003534 005020 ;COM (R0)  
1367 003536 001404 ;CLR (R0)+ ;CLEAR 400  
1368 ;BEQ SOPZA ;INITIALIZE: 400=-1  
1369 ; ;TRY CLEARING WITH MODE 2  
1370 ;  
1371 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1372 ; CONDITIONAL BRANCH INST. AND <=====  
1373 ; REPLACE THE MOVE INSTRUCTION <=====  
1374 ; WHICH FOLLOWS W/ 771 <=====  
1375 003540 012742 000070 ;SOPZA: MOV #70-(R2) ;MOVE TO MAILBOX # ***** 70 *****  
1376 003544 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1377 003546 000000 ;HALT ;CLR INST DID NOT SET Z-BIT  
1378 003550 005300 ;SOPZA: DEC R0 ;RESET R0  
1379 003552 005120 ;DEC R0  
1380 003554 100004 ;COM (R0)+ ;TRY COMPLEMENTING WITH MODE 2  
1381 003556 100004 ;COM (R0)+  
1382 003560 005300 ;BPL SOPZB ;RESET R0  
1383 003562 005300 ;DEC R0  
1384 003564 005220 ;DEC R0  
1385 003566 001404 ;INC (R0)+ ;TRY INCREMENTING WITH MODE 2  
1386 ;BEQ TST52  
1387 ;  
1388 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1389 ; CONDITIONAL BRANCH INST. AND <=====  
1390 ; REPLACE THE MOVE INSTRUCTION <=====  
1391 ; WHICH FOLLOWS W/ 755 <=====  
1392 003570 012742 000071 ;SOPZB: MOV #71-(R2) ;MOVE TO MAILBOX # ***** 71 *****  
1393 003574 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1394 003576 000000 ;HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST.  
; OR SEQUENCE ERROR
```

```
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406 003600 005212  
1407 003602 022712 000052  
1408 003606 001023  
1409 003610 005000  
1410 003612 105100  
1411 003614 005200  
1412 003616 005010  
1413 003620 005110  
1414 003622 105020  
1415 003624 001404  
1416  
1417  
1418  
1419  
1420 003626 012742 000072  
1421 003630 005242  
1422 003634 000000  
1423 003636 005300  
1424 003640 005210  
1425 003642 105110  
1426 003644 105220  
1427 003646 100003  
1428 003650 005300  
1429 003652 105220  
1430 003654 001404  
1431  
1432  
1433  
1434  
1435 003656 012742 000073  
1436 003656 005242  
1437 003662 000000  
1438 003664 000000  
1439
```

THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH ADDRESS EVEN BYTES. R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION 400 TO -1. CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH MODE 2. R0 IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST. THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE REGISTER.

TEST 52 TEST MODE 2 EVEN BYTE USING SOP INST.

TST52: INC (R2) ;UPDATE TEST NUMBER
CMP #52,(R2) ;SEQUENCE ERROR?
BNE TST53-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400
CMB R0
INC R0
CLR (R0) ;CLEAR 400
COM (R0) ;INITIALIZE: 400=-1
CLRB (R0)+ ;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 ***** <=====
; MOVE TO MAILBOX # ***** 72 *****
; SET MSGTYP TO FATAL ERROR
; CLRB DID NOT SET Z-BIT
SOPB2A: DEC R0 ;RESULT R0=400
INC (R0) ;INC 400 TO TEST WORD
CMB (R0)
INCB (R0)+
BPL SOPB2B ;TRY TO INC EVEN BYTE
R0 ;RESET R0=400
INCB (R0)+ ;TRY INCREMENT OF EVEN BYTE
BEQ TST53
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 755 <=====
SOPB2B: MOV #73,(R2) ;MOVE TO MAILBOX # ***** 73 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.
; OR SEQUENCE ERROR

```
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449 003666 005212  
1450 003670 022712 000053  
1451 003674 001026  
1452 003676 005000  
1453 003700 105100  
1454 003702 005200  
1455 003704 005010  
1456 003706 005110  
1457 003710 005200  
1458 003712 105020  
1459 003714 001404  
1460  
1461  
1462  
1463  
1464 003716 012742 000074  
1465 003722 005242  
1466 003724 000000  
1467 003726 005300  
1468 003730 005300  
1469 003732 005220  
1470 003734 005300  
1471 003736 105110  
1472 003740 105220  
1473 003742 100003  
1474 003744 005300  
1475 003746 105220  
1476 003750 001404  
1477  
1478  
1479  
1480  
1481 003752 012742 000075  
1482 003752 005242  
1483 003756 000000  
1484 003760 000000  
1486
```

THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS TEST. HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.

TEST 53 TEST MODE 2 ODD BYTE USING SOP INST.

TST53: INC (R2) ;UPDATE TEST NUMBER
CMP #53,(R2) ;SEQUENCE ERROR?
BNE TST54-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400
CMB R0
INC R0
CLR (R0) ;CLEAR LOC 400
COM (R0) ;INITIALIZE: 400=-1
CLRB (R0)+ ;R0=ODD BYTE
BEQ SOPB2C ;TRY TO CLEAR ODD BYTE
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 770 ***** <=====
; MOVE TO MAILBOX # ***** 74 *****
; SET MSGTYP TO FATAL ERROR
; CLRB DID NOT SET Z-BIT
; R0=WORD ADDR.
SOPB2C: DEC R0 ;INCREMENT WORD
INC (R0)+ ;POINT TO ODD BYTE
DEC R0 ;COMPLEMENT ODD BYTE
CMB (R0) ;TRY TO INCREMENT ODD BYTE
INCB (R0)+
BPL SOPB2D ;RESET R0 TO ODD BYTE
R0 ;TRY TO INCREMENT ODD BYTE
INCB (R0)+
BEQ TST54
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 752 <=====
SOPB2D: MOV #75,(R2) ;MOVE TO MAILBOX # ***** 75 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.
; OR SEQUENCE ERROR

```

*****
;
; THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES. PREVIOUSLY
; TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
;
;*****
;TEST 54 TEST MODE 0 USING NEGATE INSTRUCTION
;*****
TST54: INC (R2) ;UPDATE TEST NUMBER
        CMP #54,(R2) ;SEQUENCE ERROR?
        BNE TST55-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR R0 ;SET R0=0
        INC R0 ; R0=1
        RO ;TRY NEGATE MODE 0: R0=-1
        BPL NEG00 ;CC=1001?
        BEQ NEG00
        BVS NEG00
        BCS NEG01

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

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

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

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

004024 NEG02: MOV #77,-(R2) ;MOVE TO MAILBOX # ***** 77 *****
004030 INC -(R2) ;SET MSGTYP TO FATAL ERROR
004032 HALT ;DATA RESULT OF NEGATE INCORRECT

004034 NEG02: COMB R0 ;R0=377
004034 NEGB R0 ;R=1
004040 BMT NEG03 ;CC=0001?
004042 BEQ NEG03
004044 BVS NEG03
004046 BCS NEG04

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

004050 NEG03: MOV #100,-(R2) ;MOVE TO MAILBOX # ***** 100 *****
004050 INC -(R2) ;SET MSGTYP TO FATAL ERROR
004056 HALT ;NEGB DID NOT SET CC'S CORRECTLY
004060 BEQ NEG04 ;TEST DATA RESULT
004062

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

```

```

*****
;
; WHICH FOLLOWS W/ 743 <====
; MOVE TO MAILBOX # ***** 101 ***** <====
; SET MSGTYP TO FATAL ERROR <====
; DATA RESULT OF NEGB INCORRECT <====
; OR SEQUENCE ERROR <====
;*****
;TEST 55 TEST MODE 1 USING NEGATE INST.
;*****
TST55: INC (R2) ;UPDATE TEST NUMBER
        CMP #55,(R2) ;SEQUENCE ERROR?
        BNE TST56-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR R0 ;POINT TO LOC. 0
        INC R0 ;CLEAR LOC. 0
        RO ;LOC. 0=1
        BPL NEG10 ;TRY NEG. LOC. 0=-1
        BEQ NEG10 ;CC=1001?
        BVS NEG10
        BCS NEG11

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

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

004134 NEG11: INC R0 ;TEST DATA RESULT
004140 BEQ NEG12

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

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

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

004166 NEG13: MOV #104,-(R2) ;MOVE TO MAILBOX # ***** 104 *****
004172 INC -(R2) ;SET MSGTYP TO FATAL ERROR
004174 HALT ;NEGB DID NOT SET CC'S CORRECTLY
004176 INC R0 ;TEST DATA RESULT
004202 BEQ TST56

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

```



```
1650 ;*****  
1651 ; THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS. IT  
1652 ; USES LOCATION 0 AS ITS TARGET DATA TABLE LOCATED AT LOC. 400  
1653 ; THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE  
1654 ; INSTRUCTIONS UNDER TEST.  
1655 ; R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR  
1656 ; INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0. THEN R0  
1657 ; IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON  
1658 ; LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST. THE PROPER INCREMENTING  
1659 ; OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.  
1660 ; IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE  
1661 ; (LOC. 400-402) HAS THE PROPER VALUES (0).  
1662 ;*****  
1663 ;*****  
1664 ;*****  
1665 ;*****  
1666 ;*****  
1667 ;*****  
1668 ;*****  
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 ;*****
```

```
1700 ;*****  
1701 ; THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS  
1702 ; WHICH ADDRESS EVEN BYTES. AGAIN THE TARGET LOCATION 0 IS USED  
1703 ; AND THE SAME TABLE AT 400 IS EMPLOYED.  
1704 ; AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION  
1705 ; 0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.  
1706 ; SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE  
1707 ; TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.  
1708 ; IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS  
1709 ; THE PROPER VALUES (0).  
1710 ;*****  
1711 ;*****  
1712 ;*****  
1713 ;*****  
1714 ;*****  
1715 ;*****  
1716 ;*****  
1717 ;*****  
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 ; THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS  
1755 ; WHICH ADDRESS ODD BYTES. THE TARGET IS BYTE 1. A TABLE AT  
1756 ; LOC. 400-406 IS USED. R0 SERVES AS THE TABLE POINTER.  
1757 ; R0 IS INITIALIZED TO 400. LOC. 0 IS SET TO -1 USING THE  
1758 ; FIRST TWO TABLE ENTRIES. A CLR MODE 3 IS EXECUTED ON BYTE 1 USING  
1759 ; TABLE ADDRESS AT 404. R0 IS DECREMENTED TO 403 AND SEVERAL SOP  
1760 ; MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER  
1761 ; REGISTER INCREMENTING.  
1762 ; THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND  
1763 ; AFTER THE TEST IS RUN.  
1764 ;*****  
1765 ;  
1766 ; TEST 61 TEST MODE 3 ODD BYTE USING SOP INST.  
1767 ;*****  
1768 ;  
1769 004474 005212 000061 TST61: INC (R2) ;UPDATE TEST NUMBER  
1770 004476 022712 CMP #61,(R2) ;SEQUENCE ERROR?  
1771 004502 001074 BNE TST62-10 ;BR TO ERROR HALT ON SEQ ERROR  
1772 004504 005000 CLR R0 ;R0=400  
1773 004506 105100 COMB R0 ;SET R0=400  
1774 004510 005200 INC R0  
1775 004512 005030 CLR @(R0)+ ;INITIALIZE  
1776 004514 005130 CDW @(R0)+ ;LOC 0=-1 R0=404  
1777 004516 005030 CLR @(R0)+ ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406  
1778 004520 001404 BEQ SOPB3C ;  
1779 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1780 ; CONDITIONAL BRANCH INST. AND <====  
1781 ; REPLACE THE MOVE INSTRUCTION <====  
1782 ; WHICH FOLLOWS W/ 115 ***** <====  
1783 004522 012742 000115 MOV #115,-(R2) ;MOVE TO MAILBOX # *****/ 115 *****  
1784 004526 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1785 004530 000000 HALT ;CLR DID NOT SET Z-BIT  
1786 004532 005300 SOPB3C: R0 ;RESET R0=402  
1787 004534 005300 DEC R0 ;  
1788 004536 005300 DEC R0 ;POINT TO EVEN BYTE ADDR.  
1789 004540 005300 DEC R0 ;  
1790 004542 005230 INC @(R0)+ ;INCREMENT WORD LOC. 0=400 R0=404  
1791 004544 105430 NEGB @(R0)+ ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406  
1792 004546 100002 BPL SOPB3D ;  
1793 004550 105230 CLRB @(R0)+ ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410  
1794 004552 001404 BEQ TST62 ;  
1795 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1796 ; CONDITIONAL BRANCH INST. AND <====  
1797 ; REPLACE THE MOVE INSTRUCTION <====  
1798 ; WHICH FOLLOWS W/ 754 ***** <====  
1799 004554 012742 000116 SOPB3D: MOV #116,-(R2) ;MOVE TO MAILBOX # ***** 116 *****  
1800 004554 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1801 004560 000000 HALT ;CUMULATIVE RESULT OF ABOVE INSTS FAILED  
1802 004562 000000 ; OR SEQUENCE ERROR  
1803
```

```
1804 ;*****  
1805 ;  
1806 ; TEST 62 TEST MODE 3 USING NEGATE INSTRUCTION  
1807 ;*****  
1808 ;  
1809 004564 005212 000062 TST62: INC (R2) ;UPDATE TEST NUMBER  
1810 004566 022712 CMP #62,(R2) ;SEQUENCE ERROR?  
1811 004572 001054 BNE TST63-10 ;BR TO ERROR HALT ON SEQ ERROR  
1812 004574 005000 CLR R0 ;R0=400  
1813 004576 105200 COMB R0  
1814 004600 005200 INC R0  
1815 004602 005010 CLR (R0) ;LOC. 400=0  
1816 004604 005004 CLR R4 ;R4=0  
1817 004606 005014 CLR (R4) ;LOC. 0=0  
1818 004610 005230 INC @(R0)+ ;LOC. 0=-1 R0=402  
1819 004612 100003 NEG @(R0)+ ;TRY NEGATE LOC. 0=-1 R0=402  
1820 004614 100003 BPL NEG30 ;CC=1001?  
1821 004616 001402 BEQ NEG30  
1822 004620 102401 BVS NEG30  
1823 004622 103404 BCS NEG31  
1824 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1825 ; CONDITIONAL BRANCH INST. AND <====  
1826 ; REPLACE THE MOVE INSTRUCTION <====  
1827 ; WHICH FOLLOWS W/ 764 ***** <====  
1828 004624 012742 000117 NEG30: MOV #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****  
1829 004626 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1830 004630 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY  
1831 004632 005214 NEG31: INC (R4) ;LOC. 0=0  
1832 004634 001404 BEQ NEG32 ;  
1833 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1834 ; CONDITIONAL BRANCH INST. AND <====  
1835 ; REPLACE THE MOVE INSTRUCTION <====  
1836 ; WHICH FOLLOWS W/ 756 ***** <====  
1837 004640 012742 000120 MOV #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****  
1838 004642 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1839 004644 000000 HALT ;DATA RESULT OF NEG INCORRECT  
1840 004646 005137 000001 COMB #1 ;LOC. 0=177400  
1841 004648 005237 000000 INC #40 ;LOC. 0=177400  
1842 004650 105430 NEGB @(R0)+ ;TRY NEGB LOC. 0=177777 R0=404  
1843 004652 100404 BMI NEG33 ;  
1844 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1845 ; CONDITIONAL BRANCH INST. AND <====  
1846 ; REPLACE THE MOVE INSTRUCTION <====  
1847 ; WHICH FOLLOWS W/ 744 ***** <====  
1848 004664 012742 000121 MOV #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****  
1849 004666 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1850 004668 000000 HALT ;NEGB FAILED WITH EVEN BYTE  
1851 004670 105430 NEGB @(R0)+ ;TRY NEGB LOC.0=777 R0=406  
1852 004672 100004 BPL NEG34 ;  
1853 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1854 ; CONDITIONAL BRANCH INST. AND <====  
1855 ; REPLACE THE MOVE INSTRUCTION <====  
1856 ; WHICH FOLLOWS W/ 736 ***** <====  
1857 004700 012742 000122 MOV #122,-(R2) ;MOVE TO MAILBOX # *****/ 122 *****  
1858 004702 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1859 004704 000000 HALT ;NEGB FAILED WITH ODD BYTE  
1860 004710 105137 000001 NEG34: COMB #1 ;LOC. 0=177377
```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 42
CFKAAC.P11 18-OCT-78 11:01 T62 TEST MODE 3 USING NEGATE INSTRUCTION SEQ 0054
1860 004714 105237 000001 INCB #1 ;LOC. 0=17777
1861 004720 005214 INC (R4) ;LOC. 0=0
1862 004722 001404 BEQ TST63
1863
1864 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1865 ; CONDITIONAL BRANCH INST. AND <====
1866 ; REPLACE THE MOVE INSTRUCTION <====
1867 ; WHICH FOLLOWS W/ 724 <====
1868 004774 012742 000123 MOV #123,-(R2) ;MOVE TO MAILBOX # ***** 123 *****
1869 004732 000000 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1870 HALT ;DATA RESULT OF NEGATE'S INCORRECT
; OR SEQUENCE ERROR

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 43
CFKAAC.P11 18-OCT-78 11:01 T62 TEST MODE 3 USING NEGATE INSTRUCTION SEQ 0055
1871
1872 ;*****
1873 ;
1874 ; THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
1875 ;RO IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
1876 ;LOC. 376. RO IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
1877 ;COMPLEMENTS LOC. 376.
1878 ;TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
1879 ;TO COMPLETE THE TEST.
1880 ;*****
1881 ;TEST 63 TEST MODE 4 USING SOP INSTS
1882 ;*****
1883 004734 005212 000063 TST63: INC (R2) ;UPDATE TEST NUMBER
1884 004736 022712 CMP #63,(R2) ;SEQUENCE ERROR?
1885 004742 001021 BNE TST64-10 ;BR TO ERROR HALT ON SEQ ERROR
1886 004744 005000 CLR RO ;SET RO=400
1887 004746 105100 COMB RO
1888 004750 005200 INC RO
1889 004752 005040 CLR -(RO) ;TRY TO CLEAR USING MODE 4
1890 004754 001404 BEQ SOP4A
1891
1892 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1893 ; CONDITIONAL BRANCH INST. AND <====
1894 ; REPLACE THE MOVE INSTRUCTION <====
1895 ; WHICH FOLLOWS W/ 773 <====
1896 004756 012742 000124 MOV #124,-(R2) ;MOVE TO MAILBOX # ***** 124 *****
1897 004764 000000 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1898 004766 005200 HALT ;CLR DID NOT SET Z-BIT
1899 004770 005200 SOP4A: RO ;RESET RO
1900 004772 005140 COM -(RO) ;TRY TO COMPLEMENT USING MODE 4
1901 004774 100004 BPL SOP4B
1902 004776 005200 INC RO ;MOVE POINTER
1903 005000 005200 INC RO
1904 005002 005240 INC -(RO)
1905 005004 001404 BEQ TST64
1906
1907 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1908 ; CONDITIONAL BRANCH INST. AND <====
1909 ; REPLACE THE MOVE INSTRUCTION <====
1910 ; WHICH FOLLOWS W/ 757 <====
1911 005006 012742 000125 SOP4B: MOV #125,-(R2) ;MOVE TO MAILBOX # ***** 125 *****
1912 005012 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1913 005014 000000 HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
1914 ; OR SEQUENCE ERROR

```



```
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934 005016 005212  
1935 005020 022712 000064  
1936 005024 061017  
1937 005026 005000  
1938 005030 005020  
1939 005032 105400  
1940 005034 005050  
1941 005036 001404  
1942  
1943  
1944  
1945  
1946 005040 012742 000126  
1947 005044 005242  
1948 005046 000000  
1949 005050 005200  
1950 005052 005200  
1951 005054 005150  
1952 005056 100002  
1953 005060 005250  
1954 005062 001404  
1955  
1956  
1957  
1958  
1959 005064  
1960 005064 012742 000127  
1961 005070 005242  
1962 005072 000000  
1963
```

```
*****  
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 64 TEST MODE 5 USING SOP INSTS  
*****  
TST64: INC (R2) ;UPDATE TEST NUMBER  
CMP #64(R2) ;SEQUENCE ERROR?  
BNE TST65-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;SET RO=376  
CLR (R0)+  
NEGB RO  
CLR 8-(R0) ;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 #126-(R2) ;MOVE TO MAILBOX # ***** 126 *****  
INC -(R2) ;SET MSGTVP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOP5A: INC RO ;RESET RO  
INC RO  
COM 8-(R0) ;TRY TO COMPLEMENT LOC. 0 W/MODE 5  
BPL 8-(R0) ;TRY TO INCREMENT LOC. 0 W/MODE 5  
INC 8-(R0)  
BEQ TST65  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 761 <====  
SOP5B: MOV #127-(R2) ;MOVE TO MAILBOX # ***** 127 *****  
INC -(R2) ;SET MSGTVP TO FATAL ERROR  
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS  
; OR SEQUENCE ERROR
```

```
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976 005074 005212  
1977 005076 022712 000065  
1978 005102 001020  
1979 005104 005000  
1980 005106 105100  
1981 005110 005200  
1982 005112 005060 177400  
1983 005116 001404  
1984  
1985  
1986  
1987  
1988 005120 012742 000130  
1989 005124 005242  
1990 005126 000000  
1991 005130 005160 177400  
1992 005134 100003  
1993 005136 005260 177400  
1994 005142 001404  
1995  
1996  
1997  
1998  
1999 005144  
2000 005144 012742 000131  
2001 005150 005242  
2002 005152 000000  
2003
```

```
*****  
THIS TEST VERIFIES MODE 6 SINGLE OPERAND INSTRUCTIONS. IT  
USES LOCATION 0 AS ITS TARGET DATA. RO IS SET TO 400 USING  
PREVIOUSLY TESTED INSTRUCTIONS AND A MODE 6 CLR INSTRUCTION IS  
EXECUTED ON LOC. 0 USING RO AND A -400 OFFSET. COM AND INC  
INSTRUCTIONS ARE THEN USED TO VERIFY THE DATA.  
*****  
TEST 65 TEST MODE 6 USING SOP INSTS  
*****  
TST65: INC (R2) ;UPDATE TEST NUMBER  
CMP #65(R2) ;SEQUENCE ERROR?  
BNE TST66-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;SET RO=400  
COMB RO  
INC RO  
CLR -400(R0) ;TRY TO CLEAR LOCATION 0 W/MODE 6  
BEQ SOP6A  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 772 <====  
MOV #130-(R2) ;MOVE TO MAILBOX # ***** 130 *****  
INC -(R2) ;SET MSGTVP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOP6A: COM -400(R0) ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6  
BPL SOP6B  
INC -400(R0) ;TRY TO INCREMENT LOCATION 0 W/MODE 6  
BEQ TST66  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 760 <====  
SOP6B: MOV #131-(R2) ;MOVE TO MAILBOX # ***** 131 *****  
INC -(R2) ;SET MSGTVP TO FATAL ERROR  
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS  
; OR SEQUENCE ERROR
```

```
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017 005154 005212  
2018 005156 022712 000066  
2019 005162 001021  
2020 005164 005000  
2021 005166 105100  
2022 005170 005200  
2023 005172 005210  
2024 005174 005070 000002  
2025 005200 001404  
2026  
2027  
2028  
2029  
2030 005202 012742 000132  
2031 005206 005242  
2032 005210 000000  
2033 005212 005170 000002  
2034 005216 100003  
2035 005220 005270 000002  
2036 005224 001404  
2037  
2038  
2039  
2040  
2041 005226  
2042 005226 012742 000133  
2043 005232 005242  
2044 005234 000000  
2045  
2046
```

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 66 TEST MODE 7 USING SOP INST.

TST66: INC (R2) ;UPDATE TEST NUMBER
CMP #66(R2) ;SEQUENCE ERROR?
BNE #5767-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC (R0)
CLR (R0) ;RO=1
BEQ #2(R0) ;TRY TO CLEAR LOC. 0 W/MODE 7
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 ***** <====
MOV #132,-(R2) ;MOVE TO MAILBOX # ***** 132 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP7A: COM #2(R0) ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
BPL #2(R0)
INC #2(R0) ;TRY TO INCREMENT LOC. 0 W/MODE 7
BEQ #5767
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 ***** <====
SOP7B: MOV #133,-(R2) ;MOVE TO MAILBOX # ***** 133 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
; OR SEQUENCE ERROR

```
2047  
2048  
2049  
2050 005236 005212  
2051 005240 022712 000067  
2052 005244 001024  
2053 005246 005000  
2054 005250 005010  
2055 005252 005120  
2056 005254 005440  
2057 005256 100403  
2058 005260 001402  
2059 005262 102401  
2060 005264 103404  
2061  
2062  
2063  
2064  
2065 005266  
2066 005266 012742 000134  
2067 005272 005242  
2068 005274 000000  
2069 005276 005400  
2070 005300 001404  
2071  
2072  
2073  
2074  
2075 005302 012742 000135  
2076 005306 005242  
2077 005310 000000  
2078 005312 005310  
2079 005314 001404  
2080  
2081  
2082  
2083  
2084 005316 012742 000136  
2085 005322 005242  
2086 005324 000000  
2087
```

TEST 67 TEST MODE 4 WITH NEGATE INSTRUCTION

TST67: INC (R2) ;UPDATE TEST NUMBER
CMP #67(R2) ;SEQUENCE ERROR?
BNE #5770-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO
CLR (R0)
COMB (R0)+ ;LOC. 0=177777, RO=2
NEG -(R0) ;TRY NEGATE, LOC. 0=1
BWI NEG40 ;CC=0001?
BEQ NEG40
BVS NEG40
BCS NEG41
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 ***** <====
NEG40: MOV #134,-(R2) ;MOVE TO MAILBOX # ***** 134 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEG DID NOT SET CC'S CORRECTLY
NEG41: NEG RO ;TST RO WITH A NEG.
BEQ NEG42
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 ***** <====
NEG42: MOV #135,-(R2) ;MOVE TO MAILBOX # ***** 135 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RO NOT DECREMENTED PROPERLY
DEC (R0) ;TEST DTA RESULT OF NEG
BEQ #5770
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 ***** <====
MOV #136,-(R2) ;MOVE TO MAILBOX # ***** 136 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF NEG INCORRECT
; OR SEQUENCE ERROR

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 48
CFKAAC.P11 18-OCT-78 11:01 T67 TEST MODE 4 WITH NEGATE INSTRUCTION SEQ 0060
;*****
;TEST 70 TEST MODE 5 WITH NEGATE INSTRUCTION
;*****
2088 ;*****
2089 ;*****
2090 ;*****
2091 005326 005212 000070 TST70: INC (R2) ;UPDATE TEST NUMBER
2092 005330 022712 000070 CMP #70,(R2) ;SEQUENCE ERROR?
2093 005334 001031 000070 BNE #T71-10 ;BR TO ERROR HALT ON SEQ ERROR
2094 005336 005000 000070 CLR R0 ;R0=0
2095 005340 005010 000070 CLR R0 ;R0=377
2096 005342 105100 000070 COMB R0 ;R0=400
2097 005344 005200 000070 INC R0 ;SET 400 = 0
2098 005346 005010 000070 CLR (R0) ;R4=0
2099 005350 005004 000070 CLR R4 ;LOC 0=177777
2100 005352 005314 000070 DEC (R4) ;LOC 0=177777
2101 005354 005450 000070 NEG (R0) ;TRY NEGATE: LOC. 0=1
2102 005356 100403 000070 BMI NEG50 ;CC=0001?
2103 005360 001402 000070 BEQ NEG50
2104 005362 102401 000070 BVS NEG50
2105 005364 103404 000070 BCS NEG51
2106 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2107 ; CONDITIONAL BRANCH INST. AND <====
2108 ; REPLACE THE MOVE INSTRUCTION <====
2109 ; WHICH FOLLOWS W/ 764 <====
2110 005366 012742 000137 NEG50: MOV #137,-(R2) ;MOVE TO MAILBOX # ***** 137 *****
2111 005368 005242 000137 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2112 005374 000000 000137 HALT ;NEG DID NOT SET CC'S CORRECTLY
2113 005376 005314 000137 NEG51: DEC (R4)
2114 005378 001404 000137 BEQ NEG52
2115 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2116 ; CONDITIONAL BRANCH INST. AND <====
2117 ; REPLACE THE MOVE INSTRUCTION <====
2118 ; WHICH FOLLOWS W/ 756 <====
2119 005402 012742 000140 NEG52: MOV #140,-(R2) ;MOVE TO MAILBOX # ***** 140 *****
2120 005406 005242 000140 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2121 005410 000000 000140 HALT ;DATA RESULT OF NEG INCORRECT
2122 005412 105100 000140 COMB R0
2123 005414 001404 000140 DEC R0
2124 005416 001404 000140 BEQ TST71
2125 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2126 ; CONDITIONAL BRANCH INST. AND <====
2127 ; REPLACE THE MOVE INSTRUCTION <====
2128 ; WHICH FOLLOWS W/ 747 <====
2129 005420 012742 000141 MOV #141,-(R2) ;MOVE TO MAILBOX # ***** 141 *****
2130 005424 005242 000141 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2131 005426 000000 000141 HALT ;REGISTER NOT DECREMENTED PROPERLY
2132 ; OR SEQUENCE ERROR
2133 ;*****

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 49
CFKAAC.P11 18-OCT-78 11:01 T70 TEST MODE 5 WITH NEGATE INSTRUCTION SEQ 0061
;*****
;TEST 71 TEST MODE 6 WITH NEGATE
;*****
2134 ;*****
2135 ;*****
2136 ;*****
2137 005430 005212 000071 TST71: INC (R2) ;UPDATE TEST NUMBER
2138 005432 022712 000071 CMP #71,(R2) ;SEQUENCE ERROR?
2139 005436 001022 000071 BNE #T72-10 ;BR TO ERROR HALT ON SEQ ERROR
2140 005440 005000 000071 CLR R0 ;R0=0
2141 005442 005004 000071 CLR R4 ;R4=0
2142 005444 105100 000071 COMB R0 ;R0=377
2143 005446 005014 000071 CLR (R4) ;LOC 0=0
2144 005450 105024 000071 CLRB (R4)+ ;LOC 0=177777, R4=1
2145 005452 105114 000071 COMB (R4) ;LOC 0=177400
2146 005454 005460 000071 NEG -377(R0) ;LOC 0=400
2147 005460 100403 000071 BMI NEG60 ;CC=0001
2148 005462 001402 000071 BEQ NEG60
2149 005464 102401 000071 BVS NEG60
2150 005466 103404 000071 BCS NEG61
2151 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2152 ; CONDITIONAL BRANCH INST. AND <====
2153 ; REPLACE THE MOVE INSTRUCTION <====
2154 ; WHICH FOLLOWS W/ 764 <====
2155 005470 012742 000142 NEG60: MOV #142,-(R2) ;MOVE TO MAILBOX # ***** 142 *****
2156 005474 005242 000142 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2157 005476 000000 000142 HALT ;NEG DID NOT SET CC'S CORRECTLY
2158 005500 105314 000142 NEG61: DEC (R4)
2159 005502 001404 000142 BEQ TST72
2160 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2161 ; CONDITIONAL BRANCH INST. AND <====
2162 ; REPLACE THE MOVE INSTRUCTION <====
2163 ; WHICH FOLLOWS W/ 756 <====
2164 005504 012742 000143 MOV #143,-(R2) ;MOVE TO MAILBOX # ***** 143 *****
2165 005510 005242 000143 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2166 005512 000000 000143 HALT ;DATA RESULT OF NEG INCORRECT
2167 ; OR SEQUENCE ERROR
2168 ;*****

```

```

;*****
;TEST 72 TEST MODE 7 W/ NEGATE
;*****
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
        CLR (R0) ;LOC=0=0
        COM (R0) ;LOC=177777
        RO ;R0=377
        COMB RO ;R0+5=404, 404=1, LOC. 0=777
        NEG B #5(R0) ;CC=0001?
        BMI NEG70
        BEQ NEG70
        BVS NEG70
        BCS NEG71

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

NEG70: MOV #144,-(R2) ;MOVE TO MAILBOX # ***** 144 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;NEG DID NOT SET CC'S CORRECTLY
        COMB R0 ;R0=0
        DECB (R0) ;LOC=0=400, R0=1
        NEG 0 ;LOC=0=0
        BEQ TST73 ;USE NEG MODE 67 TO TST FOR ZERO

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

NEG71: MOV #145,-(R2) ;MOVE TO MAILBOX # ***** 145 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;DATA RESULT OF NEG WAS INCORRECT
        ; OR SEQUENCE ERROR
    
```

```

;*****
; THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
; INSTRUCTIONS. CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
; INSTRUCTION (SOPX). THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
; 77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
; OF THESE INSTRUCTIONS.
;*****
;TEST 73 TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
;*****
TST73: INC (R2) ;UPDATE TEST NUMBER
        CMP #73,(R2) ;SEQUENCE ERROR?
        BNE SOPB ;BR TO ERROR HALT ON SEQ ERROR
        CLR (R7)+ ;CLEAR NEXT LOCATION: (SOPX)
        BEQ SOPA ;USE MODE 27

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 775 <====

SOPA: MOV #146,-(R2) ;MOVE TO MAILBOX # ***** 146 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CLR DID NOT SET 2-BIT
        INC @SOPX ;INC SOPX W/MODE 37
        NEG SOPB ;NEGATE SOPX W/MODE 67
        BPL @SOPXAD ;INC SOPX W/MODE 77
        INC @SOPXAD
        BEQ TST74

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

SOPB: MOV #147,-(R2) ;MOVE TO MAILBOX # ***** 147 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;INC DID NOT SET 2-BIT
        ; OR SEQUENCE ERROR
        SOPXAD: SOPX ;INDIRECT ADDRESS OF SOPX
    
```

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

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

2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321 005764 005212
2322 005766 022712 000076
2323 005772 001011
2324 005774 005000
2325 005776 005010
2326 006000 000277
2327 006002 000244
2328 006004 005710
2329 006006 102403
2330 006010 103402
2331 006012 100401
2332 006014 001404
2333
2334
2335
2336
2337 006016
2338 006016 012742 000152
2339 006022 005242
2340 006024 000000
2341

```
*****  
; 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 76 TEST MODE 1 SOP NON-MODIFYING  
*****  
TST76: INC (R2) ;UPDATE TEST NUMBER  
CMP #76,(R2) ;SEQUENCE ERROR?  
BNE TST77-10 ;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 TST77  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====  
  
SNM1A: MOV #152,-(R2) ;MOVE TO MAILBOX # ***** 152 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT SET PROPERLY  
; OR SEQUENCE ERROR
```

2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353 006026 005212
2354 006030 022712 000077
2355 006034 001026
2356 006036 005000
2357 006040 005010
2358 006042 105110
2359 006044 000277
2360 006046 000250
2361 006050 105710
2362 006052 102402
2363 006054 101401
2364 006056 100404
2365
2366
2367
2368
2369 006060
2370 006060 012742 000153
2371 006064 005242
2372 006066 000000
2373 006070 005000
2374 006072 005200
2375 006074 000277
2376 006076 000244
2377 006100 105710
2378 006102 102403
2379 006104 103402
2380 006106 100401
2381 006110 001404
2382
2383
2384
2385
2386 006112
2387 006112 012742 000154
2388 006116 005242
2389 006120 000000
2390

```
*****  
; THIS TEST SETS LOCATION 0 TO 377 AND THEN USES RO TO TEST  
; THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.  
; AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE  
; PROPER CONDITION CODE BITS.  
*****  
;TEST 77 TEST MODE 1 BYTE INST. NON-MODIFYING  
*****  
TST77: INC (R2) ;UPDATE TEST NUMBER  
CMP #77,(R2) ;SEQUENCE ERROR?  
BNE TST100-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;POINT TO LOC 0  
CLR (RO) ;CLEAR LOC 0  
COMB (RO) ;COMPLEMENT BYTE 0  
SCC ;SET CC=0111  
CLZ  
TSTB (RO) ;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 #153,-(R2) ;MOVE TO MAILBOX # ***** 153 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT CORRECT  
  
SNMB1B: CLR RO  
INC RO  
SCC ;SET CC=1011  
CLZ  
TSTB (RO) ;TRY TO TST AN ODD BYTE  
BVS SNMB1C  
BCS SNMB1C  
BMI SNMB1C  
BEQ TST100  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 752 <====  
  
SNMB1C: MOV #154,-(R2) ;MOVE TO MAILBOX # ***** 154 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT CORRECT  
; OR SEQUENCE ERROR
```

```
2391 ;*****  
2392 ;  
2393 ; THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS  
2394 ; USING MODE 2. IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE  
2395 ; MODE 1 TESTS. ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT  
2396 ; IT IS INCREMENTED PROPERLY.  
2397 ;  
2398 ;*****  
2399 ;  
2400 ;TEST 100 TEST MODE 2 WITH SOP NON-MODIFYING  
2401 ;*****  
2402 006122 005212 000100 TST100: INC (R2) ;UPDATE TEST NUMBER  
2403 006124 022712 ;SEQUENCE ERROR?  
2404 006130 001020 ;BNE TST101-10 ;BR TO ERROR HALT ON SEQ ERROR  
2405 006132 005000 ;CLR RO ;INITIALIZE RO=0  
2406 006134 005010 ;CLR (R0) ;CLEAR LOC 0  
2407 006136 000277 ;SCC ;SET CC=1011  
2408 006140 000244 ;CLZ  
2409 006142 005720 ;TST (R0)+ ;TRY TST W/ MODE 2  
2410 006144 102403 ;BVS SNM2A ;CHECK CC=0100  
2411 006146 103402 ;BCS SNM2A  
2412 006150 100401 ;BMI SNM2A  
2413 006152 001404 ;BEQ SNM2B  
2414 ;  
2415 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2416 ; CONDITIONAL BRANCH INST. AND <====  
2417 ; REPLACE THE MOVE INSTRUCTION <====  
2418 ; WHICH FOLLOWS W/ 767 <====  
2419 006154 012742 000155 SNM2A: MOV #155,-(R2) ;MOVE TO MAILBOX # ***** 155 *****  
2420 006160 005242 ;INC ;SET MSGTYP TO FATAL ERROR  
2421 006162 000000 ;HALT ;CC'S NOT CORRECT  
2422 006164 005300 ;SNM2B: RO ;RESET RO  
2423 006166 005300 ;DEC  
2424 006170 001404 ;BEQ TST101  
2425 ;  
2426 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2427 ; CONDITIONAL BRANCH INST. AND <====  
2428 ; REPLACE THE MOVE INSTRUCTION <====  
2429 ; WHICH FOLLOWS W/ 760 <====  
2430 006172 012742 000156 MOV #156,-(R2) ;MOVE TO MAILBOX # ***** 156 *****  
2431 006176 005242 ;INC ;SET MSGTYP TO FATAL ERROR  
2432 006200 000000 ;HALT ;MODE 2 DID NOT INC REG CORRECTLY  
 ; OR SEQUENCE ERROR
```

```
2433 ;*****  
2434 ;  
2435 ; THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE  
2436 ; INSTRUCTIONS IT USES RO TO POINT TO LOC. 0. WITH LOCATION 0  
2437 ; SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS  
2438 ; TO VERIFY THE CORRECT CC ARE SET. THE REGISTER IS CHECKED FOR  
2439 ; PROPER INCREMENTING.  
2440 ;  
2441 ;*****  
2442 ;TEST 101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING  
2443 ;*****  
2444 006202 005212 000101 TST101: INC (R2) ;UPDATE TEST NUMBER  
2445 006204 022712 ;CMP #101,(R2) ;SEQUENCE ERROR?  
2446 006210 001042 ;BNE TST102-10 ;BR TO ERROR HALT ON SEQ ERROR  
2447 006212 005000 ;CLR RO ;CLEAR RO  
2448 006214 005110 ;CLR (R0) ;CLEAR LOC 0  
2449 006216 105110 ;COMB (R0) ;SET LOC 0=377  
2450 006220 000277 ;SCC ;SET CC=0111  
2451 006222 000250 ;CLN  
2452 006224 005720 ;TSTB (R0)+ ;TRY TST OF EVEN BYTE  
2453 006226 102403 ;BVS SNMB2A  
2454 006228 103402 ;BLOS SNMB2A  
2455 006230 101401 ;BMI SNMB2B  
2456 006232 100404 ;BEQ SNMB2B  
2457 ;  
2458 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2459 ; CONDITIONAL BRANCH INST. AND <====  
2460 ; REPLACE THE MOVE INSTRUCTION <====  
2461 ; WHICH FOLLOWS W/ 767 <====  
2462 006234 012742 000157 SNMB2A: MOV #157,-(R2) ;MOVE TO MAILBOX # ***** 157 *****  
2463 006240 005242 ;INC ;SET MSGTYP TO FATAL ERROR  
2464 006242 000000 ;HALT ;CC'S NOT SET CORRECTLY  
2465 006244 005300 ;SNMB2B: DEC RO ;DECREMENT RO  
2466 006246 001404 ;BEQ SNMB2C  
2467 ;  
2468 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2469 ; CONDITIONAL BRANCH INST. AND <====  
2470 ; REPLACE THE MOVE INSTRUCTION <====  
2471 ; WHICH FOLLOWS W/ 761 <====  
2472 006250 012742 000160 SNMB2C: MOV #160,-(R2) ;MOVE TO MAILBOX # ***** 160 *****  
2473 006254 005242 ;INC ;SET MSGTYP TO FATAL ERROR  
2474 006256 000000 ;HALT ;MODE 2 DID NOT INC REG CORRECTLY  
2475 006260 000277 ;SNMB2C: INC RO ;POINT TO ODD BYTE  
2476 006264 000244 ;SCC ;SET CC=1011  
2477 006266 005720 ;CLZ  
2478 006270 102403 ;TSTB (R0)+ ;TRY TST OF ODD BYTE  
2479 006272 103402 ;BVS SNMB2D ;CHECK CC'S=0100  
2480 006274 100401 ;BCS SNMB2D  
2481 006276 001404 ;BMI SNMB2D  
2482 ;BEQ SNMB2E  
2483 ;  
2484 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2485 ; CONDITIONAL BRANCH INST. AND <====  
2486 ; REPLACE THE MOVE INSTRUCTION <====  
2487 ; WHICH FOLLOWS W/ 745 <====  
2488 006300 012742 000161 SNMB2D: MOV #161,-(R2) ;MOVE TO MAILBOX # ***** 161 *****  
2489 006304 005242 ;INC ;SET MSGTYP TO FATAL ERROR
```

2489 006306 000000
2490 006310 005300
2491 006312 005300
2492 006314 001404
2493
2494
2495
2496
2497
2498 006316 012742 000162
2499 006322 005242
2500 006324 000000

SNMB2E: HALT ;CC'S NOT CORRECT
DEC RO
DEC RO
BEQ TST102

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 736 <====
; MOVE TO MAILBOX # ***** 162 *****
; SET MSGTYP TO FATAL ERROR
; RO DID NOT INCREMENT PROPERLY
; OR SEQUENCE ERROR

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

2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512 006326 005212
2513 006330 027742 000102
2514 006334 001022
2515 006336 005000
2516 006340 005010
2517 006342 105100
2518 006344 005300
2519 006346 002777
2520 006350 002744
2521 006352 005730
2522 006354 102403
2523 006356 103402
2524 006360 100401
2525 006362 001404

; 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 102 TEST MODE 3 W/ SOP NON-MODIFYING INSTS

TST102: INC (R2) ;UPDATE TEST NUMBER
CMP #102, (R2) ;SEQUENCE ERROR?
BNE TST103-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;RO=0
CLR (R0) ;CLEAR LOC 0
COMB RO ;RO=376
DEC RO ;SET CC=1011
SCC
CLZ ;TRY TST W/ MODE 3
TST @(R0)+ ;CHECK CC=0100
BVS SNM3A
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 #163, -(R2) ;MOVE TO MAILBOX # ***** 163 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
SNM3B: DEC RO ;RO=377
RO ;RO=0
BEQ TST103

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

2530 006364
2531 006370 012742 000163
2532 006370 005242
2533 006372 000000
2534 006374 005300
2535 006376 105100
2536 006400 001404
2537
2538
2539
2540
2541 006402 012742 000164
2542 006406 005242
2543 006410 000000
2544

MOV #164, -(R2)
INC -(R2)
HALT

; MOVE TO MAILBOX # ***** 164 *****
; SET MSGTYP TO FATAL ERROR
; MODE 3 DID NOT INC REG CORRECTLY
; OR SEQUENCE ERROR


```

2545 ;*****
2546 ;
2547 ;
2548 ; THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3
2549 ; LOC. 0 IS SET TO 377. TABLE AT LOC. 402-404 IS USED TO TEST
2550 ; BYTE 0 AND BYTE 1. THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND
2551 ; THE CC'S ARE VERIFIED.
2552 ; THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND
2553 ; AFTER THE TEST IS RUN.
2554 ;*****
2555 ;***** TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS *****
2556 ;*****
2557 ;
2558 006412 005212 000103 TST103: INC (R2) ;UPDATE TEST NUMBER
2559 006414 022712 ;SEQUENCE ERROR?
2560 006420 001036 BNE TST104-10 ;BR TO ERROR HALT ON SEQ ERROR
2561 006422 005000 CLR RO ;RO=0
2562 006424 005010 CLR (RO) ;CLEAR LOC 0
2563 006426 105110 COMB (RO) ;LOC. 0 =377
2564 006430 105100 COMB RO
2565 006432 005200 INC RO
2566 006434 005720 TST (RO)+ ;RO=402
2567 006436 000277 SCC ;CC=0111
2568 006440 002550 CLN
2569 006442 105730 TSTB @ (RO)+ ;TRY TST OF EVEN BYTE
2570 006444 102402 BVS SNMB3A ;CHECK CC=1000
2571 006446 101401 BLOS SNMB3A
2572 006450 100404 BMI SNMB3B
2573 ;
2574 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2575 ; CONDITIONAL BRANCH INST. AND <====
2576 ; REPLACE THE MOVE INSTRUCTION <====
2577 ; WHICH FOLLOWS W/ 764 <====
2578 006452 012742 000165 SNMB3A: MOV #165,-(R2) ;MOVE TO MAILBOX # ***** 165 *****
2579 006456 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2580 006460 000000 HALT ;CC'S NOT CORRECT
2581 006462 000277 SNMB3B: SCC ;SET CC=1011
2582 006464 000244 CLZ
2583 006466 105730 TSTB @ (RO)+ ;TRY TST OF ODD BYTE
2584 006470 102403 BVS SNMB3C ;CHECK CC=0100
2585 006472 103402 BCS SNMB3C
2586 006474 100401 BMI SNMB3C
2587 006476 001404 BEQ SNMB3D
2588 ;
2589 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2590 ; CONDITIONAL BRANCH INST. AND <====
2591 ; REPLACE THE MOVE INSTRUCTION <====
2592 ; WHICH FOLLOWS W/ 751 <====
2593 006500 012742 000166 SNMB3C: MOV #166,-(R2) ;MOVE TO MAILBOX # ***** 166 *****
2594 006504 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2595 006506 000000 HALT ;CC'S NOT CORRECT
2596 006510 005720 TST (RO)+ ;RO=410
2597 006514 100404 BMI TST104
2598 ;
2599 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2600 ; CONDITIONAL BRANCH INST. AND <====

```

```

2601 ;
2602 ; REPLACE THE MOVE INSTRUCTION <====
2603 ; WHICH FOLLOWS W/ 764 <====
2604 006516 012742 000167 MOV #167,-(R2) ;MOVE TO MAILBOX # ***** 167 *****
2605 006522 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2606 006524 000000 HALT ;TST DID NOT INCREMENT RO CORRECTLY
2607 ; OR SEQUENCE ERROR
2608 ;*****
2609 ;***** TEST MODE 4 W/ SOP NON-MODIFYING INSTS *****
2610 ;*****
2611 ;
2612 ; THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
2613 ; LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
2614 ; EXPECTED RESULTS. RO AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
2615 ; THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
2616 ; IS CHECKED FOR PROPER DECREMENTING.
2617 ;*****
2618 ;***** TEST MODE 4 W/ SOP NON-MODIFYING INSTS *****
2619 ;*****
2620 ;
2621 006526 005212 000104 TST104: INC (R2) ;UPDATE TEST NUMBER
2622 006530 022712 CMP #104,(R2) ;SEQUENCE ERROR?
2623 006534 001017 BNE TST105-10 ;BR TO ERROR HALT ON SEQ ERROR
2624 006536 005000 CLR RO ;RO=0
2625 006540 005010 CLR (RO) ;LOC 0=0
2626 006542 005120 COMB (RO)+ ;LOC 0=-1
2627 006544 000277 SCC ;SET CC=1011
2628 006546 000244 CLZ
2629 006550 005740 TST -(RO) ;TRY TST W/ MODE 4
2630 006552 102402 BVS SNM4A ;CHECK CC=0100
2631 006554 101401 BLOS SNM4A
2632 006556 100404 BMI SNM4B
2633 ;
2634 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2635 ; CONDITIONAL BRANCH INST. AND <====
2636 ; REPLACE THE MOVE INSTRUCTION <====
2637 ; WHICH FOLLOWS W/ 767 <====
2638 006560 012742 000170 SNM4A: MOV #170,-(R2) ;MOVE TO MAILBOX # ***** 170 *****
2639 006564 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2640 006566 000000 HALT ;CC'S NOT CORRECT
2641 006570 005700 TST RO
2642 006572 001404 BEQ TST105
2643 ;
2644 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2645 ; CONDITIONAL BRANCH INST. AND <====
2646 ; REPLACE THE MOVE INSTRUCTION <====
2647 ; WHICH FOLLOWS W/ 761 <====
2648 006574 012742 000171 SNM4B: MOV #171,-(R2) ;MOVE TO MAILBOX # ***** 171 *****
2649 006600 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2650 006602 000000 HALT ;TST MODE 4 DID NOT DEC RO CORRECTLY
2651 ; OR SEQUENCE ERROR

```

```
2648 ;*****  
2649 ; THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.  
2650 ; IT USES A POINTER AT LOC 376 TO TEST LOC 0. RO IS SET  
2651 ; TO 400. A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.  
2652 ; RO IS CHECKED TO INSURE PROPER DECREMENTING.  
2653 ;*****  
2654 ; TEST 105 TEST MODE 5 W/ SOP NON-MODIFYING INSTS  
2655 ;*****  
2656 TST105: INC (R2) ;UPDATE TEST NUMBER  
2657 CMP #105,(R2) ;SEQUENCE ERROR?  
2658 BNE TST106-10 ;BR TO ERROR HALT ON SEQ ERROR  
2659 CLR RO ;RO=0  
2660 LCL 0=0 ;LOC 0=0  
2661 CLR (R0) ;LOC 0=-1  
2662 COM (R0) ;RO=377  
2663 COMB RO ;RO=400  
2664 INC RO ;SET CC=0111  
2665 SCC ;  
2666 CLN ;  
2667 TST 0-(R0) ;TRY TST W/ MODE 5  
2668 BVS SNM5A ;CHECK CC=1000  
2669 BLOS SNM5A  
2670 BMI SNM5B  
2671 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2672 ; CONDITIONAL BRANCH INST. AND <====  
2673 ; REPLACE THE MOVE INSTRUCTION <====  
2674 ; WHICH FOLLOWS W/ 765 <====  
2675 ;*****  
2676 SNM5A: MOV #172,-(R2) ;MOVE TO MAILBOX # ***** 172 *****  
2677 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2678 HALT ;CC'S NOT SET PROPERLY  
2679 SNM5B: RO ;RO=377  
2680 RO ;RO=0  
2681 BEQ TST106  
2682 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2683 ; CONDITIONAL BRANCH INST. AND <====  
2684 ; REPLACE THE MOVE INSTRUCTION <====  
2685 ; WHICH FOLLOWS W/ 765 <====  
2686 ;*****  
2687 SNM5B: MOV #173,-(R2) ;MOVE TO MAILBOX # ***** 173 *****  
2688 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2689 HALT ;MODE 5 DID NOT DEC RO CORRECTLY  
2690 ; OR SEQUENCE ERROR  
2691 ;*****
```

```
2692 ;*****  
2693 ; THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.  
2694 ; RO IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED  
2695 ; USING RO AND AN OFFSET OF -377. THE CC'S ARE CHECKED AS WELL  
2696 ; AS RO TO INSURE IT WAS NOT ALTERED.  
2697 ;*****  
2698 ; TEST 106 TEST MODE 6 W/ SOP NON-MODIFYING INSTS  
2699 ;*****  
2700 TST106: INC (R2) ;UPDATE TEST NUMBER  
2701 CMP #106,(R2) ;SEQUENCE ERROR?  
2702 BNE TST107-10 ;BR TO ERROR HALT ON SEQ ERROR  
2703 CLR RO ;RO=0  
2704 LCL 0=0 ;LOC 0=0  
2705 CLR (R0) ;LOC 0=-1  
2706 COM (R0) ;RO=377  
2707 COMB RO ;SET CC=0111  
2708 SCC ;  
2709 CLN ;  
2710 TST -377(R0) ;TRY TST W/ MODE 6  
2711 BVS SNM6A ;CHECK CC=1000  
2712 BLOS SNM6A  
2713 BMI SNM6B  
2714 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2715 ; CONDITIONAL BRANCH INST. AND <====  
2716 ; REPLACE THE MOVE INSTRUCTION <====  
2717 ; WHICH FOLLOWS W/ 765 <====  
2718 ;*****  
2719 SNM6A: MOV #174,-(R2) ;MOVE TO MAILBOX # ***** 174 *****  
2720 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2721 HALT ;CC'S INCORRECT  
2722 SNM6B: RO ;RO=0  
2723 COMB RO  
2724 BEQ TST107  
2725 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2726 ; CONDITIONAL BRANCH INST. AND <====  
2727 ; REPLACE THE MOVE INSTRUCTION <====  
2728 ; WHICH FOLLOWS W/ 765 <====  
2729 ;*****  
2730 SNM6B: MOV #175,-(R2) ;MOVE TO MAILBOX # ***** 175 *****  
2731 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2732 HALT ;TST MODE 6 INCORRECTLY CHANGED RO  
2733 ; OR SEQUENCE ERROR
```

```
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746 006752 005212  
2747 006754 027712 000107  
2748 006760 001021  
2749 006762 005000  
2750 006764 005010  
2751 006766 005110  
2752 006770 105100  
2753 006772 000277  
2754 006774 000250  
2755 006776 005770 000001  
2756 007002 102402  
2757 007004 101401  
2758 007006 100404  
2759  
2760  
2761  
2762  
2763  
2764 007010  
2765 007014 012742 000176  
2766 007016 000000  
2767 007020 105100  
2768 007022 001404  
2769  
2770  
2771  
2772 007024 012742 000177  
2773 007030 005242  
2774 007032 000000  
2775
```

```
*****  
THIS TEST VERIFIES MODE 7 SOP NON-MODIFYING INSTRUCTIONS.  
IT USES A POINTER TO LOC. 0 STORED AT LOC. 400 TO TEST LOC. 0  
R0 IS SET TO 377 AND LOC. 0 IS TESTED THRU THE POINTER AT 400 USING  
R0 AND AN OFFSET OF 1.  
*****  
TEST 107 TEST MODE 7 W/ SOP NON-MODIFYING INSTS.  
*****  
TST107: INC (R2) ;UPDATE TEST NUMBER  
CMP #107,(R2) ;SEQUENCE ERROR?  
BNE TST110-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC 0=0  
COM (R0) ;LOC 0=-1  
COMB R0 ;R0=377  
SCC ;CC=0111  
CLW  
TST #1(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 #176,-(R2) ;MOVE TO MAILBOX # ***** 176 *****  
;SET MSGTYP TO FATAL ERROR  
SNM7B: COMB R0 ;R0 NOT CORRECT  
BEQ R0 ;R0=0  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====  
  
MOV #177,-(R2) ;MOVE TO MAILBOX # ***** 177 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TST MODE 7 INCORRECTLY CHANGED R0  
; OR SEQUENCE ERROR
```

```
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786 007034 005212  
2787 007036 027712 000110  
2788 007042 001006  
2789 007044 005000  
2790 007046 005100  
2791 007050 005004  
2792 007052 060004  
2793 007054 005204  
2794 007056 001404  
2795  
2796  
2797  
2798  
2799 007060 012742 000200  
2800 007064 005242  
2801 007066 000000  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813 007070 005212  
2814 007072 027712 000111  
2815 007076 001006  
2816 007100 005000  
2817 007102 005004  
2818 007104 005100  
2819 007106 010004  
2820 007110 005204  
2821 007112 001404  
2822  
2823  
2824  
2825  
2826 007114 012742 000201  
2827 007120 005242  
2828 007122 000000  
2829  
2830  
2831
```

```
*****  
THIS TEST VERIFIES MODE 0 DOUBLE OPERAND INSTRUCTIONS. IT SETS  
DATA IN R0 AND R4 AND USES THE ADD INSTRUCTION TO TEST THE DOP  
MICROCODE.  
*****  
TEST 110 TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.  
*****  
TST110: INC (R2) ;UPDATE TEST NUMBER  
CMP #110,(R2) ;SEQUENCE ERROR?  
BNE TST111-10 ;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 TST111  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 772 <====  
  
MOV #200,-(R2) ;MOVE TO MAILBOX # ***** 200 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ADD INST. FAILED W/ MODE 0  
; OR SEQUENCE ERROR  
  
*****  
THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.  
THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES UNIQUE  
MICROCODE.  
*****  
TEST 111 MOV MODE 0 TO MODE 0  
*****  
TST111: INC (R2) ;UPDATE TEST NUMBER  
CMP #111,(R2) ;SEQUENCE ERROR?  
BNE TST112-10 ;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  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 772 <====  
  
MOV #201,-(R2) ;MOVE TO MAILBOX # ***** 201 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MOVE FAILED MODE 0 TO MODE 0  
; OR SEQUENCE ERROR  
  
*****
```

```
2832 ;
2833 ; THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
2834 ; THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES SOME
2835 ; UNIQUE MICROCODE.
2836 ;
2837 ;*****
2838 ;TEST 112 TEST SUB MODE 0,0
2839 ;*****
2840 007124 005212 000112 TST112: INC (R2) ;UPDATE TEST NUMBER
2841 007126 022712 ;SEQUENCE ERROR?
2842 007132 001016 BNE TST113-10 ;BR TO ERROR HALT ON SEQ ERROR
2843 007134 005000 CLR R0 ;R0=0
2844 007136 005004 CLR R4 ;R4=0
2845 007140 005204 INC R4 ;R4=1
2846 007142 104000 SBL SUB R0 ;TRY SUB 0,0 R0=-1
2847 007144 104003 BPL SUB0 ;CC=1001
2848 007146 001407 BEQ SUB0
2849 007150 102401 BVS SUB0
2850 007152 103404 BCS SUB0A
2851 ;
2852 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2853 ; CONDITIONAL BRANCH INST. AND <====
2854 ; REPLACE THE MOVE INSTRUCTION <====
2855 ; WHICH FOLLOWS W/ 770 <====
2856 007154 012742 000202 SUB0: MOV #202-(R2) ;MOVE TO MAILBOX # ***** 202 *****
2857 007160 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2858 007162 000000 HALT ;CONDITION CODE FAILED ON SUB
2859 007164 005200 SUB0A: RO
2860 007166 001404 BEQ TST113
2861 ;
2862 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2863 ; CONDITIONAL BRANCH INST. AND <====
2864 ; REPLACE THE MOVE INSTRUCTION <====
2865 ; WHICH FOLLOWS W/ 762 <====
2866 007170 012742 000203 MOV #203-(R2) ;MOVE TO MAILBOX # ***** 203 *****
2867 007174 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2868 007176 000000 HALT ;DATA RESULT OF SUB FAILED
; OR SEQUENCE ERROR
```

```
2869 ;*****
2870 ; THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
2871 ; WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS
2872 ; SINGLE OPERAND INSTRUCTIONS ARE USED TO SET DATA IN R0 AND R4
2873 ; BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
2874 ; VERIFIED.
2875 ;*****
2876 ;TEST 113 TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
2877 ;*****
2878 ;*****
2879 ;*****
2880 ;*****
2881 007200 005212 000113 TST113: INC (R2) ;UPDATE TEST NUMBER
2882 007202 022712 CMP #113,(R2) ;SEQUENCE ERROR?
2883 007206 001051 BNE TST114-10 ;BR TO ERROR HALT ON SEQ ERROR
2884 007210 005000 CLR R0 ;R0=0
2885 007212 013004 MOV R0,R4 ;TRY MOVE MODE 0,0
2886 007214 001404 BEQ DOP0A
2887 ;
2888 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2889 ; CONDITIONAL BRANCH INST. AND <====
2890 ; REPLACE THE MOVE INSTRUCTION <====
2891 ; WHICH FOLLOWS W/ 775 <====
2892 007216 012742 000204 DOP0A: MOV #204-(R2) ;MOVE TO MAILBOX # ***** 204 *****
2893 007222 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2894 007224 000000 HALT ;Z-BIT NOT SET
2895 007226 005200 DOP0A: RO ;R0=1
2896 007230 005100 COM R4 ;R0=177776
2897 007232 005104 COM R4 ;R4=177777
2898 007234 040004 BIC R0,R4 ;TRY BIC: R4=1
2899 007236 005304 DEC R4 ;R4=0
2900 007240 001404 BEQ DOP0B
2901 ;
2902 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2903 ; CONDITIONAL BRANCH INST. AND <====
2904 ; REPLACE THE MOVE INSTRUCTION <====
2905 ; WHICH FOLLOWS W/ 763 <====
2906 007242 012742 000205 DOP0B: MOV #205-(R2) ;MOVE TO MAILBOX # ***** 205 *****
2907 007246 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2908 007250 000000 HALT ;BIC CLEAR RESULT INCORRECT
2909 007252 005004 BIC R0,R4 ;TRY BIC: R4=177777
2910 007254 005204 INC R4
2911 007256 005204 INC R4 ;R4=0
2912 007260 001404 BEQ DOP0C
2913 ;
2914 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2915 ; CONDITIONAL BRANCH INST. AND <====
2916 ; REPLACE THE MOVE INSTRUCTION <====
2917 ; WHICH FOLLOWS W/ 753 <====
2918 007262 012742 000206 DOP0C: MOV #206-(R2) ;MOVE TO MAILBOX # ***** 206 *****
2919 007266 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2920 007270 000000 HALT ;RESULT OF BIS INCORRECT
2921 007272 005000 CLR R0 ;R0=0
2922 007274 105100 CONB R0 ;R0=377
2923 007276 005004 CLR R4 ;R4=0
2924 007300 005104 COM R4 ;R4=177777
2925 007302 040004 BIC R0,R4 ;R4=177400
2926 007304 060004 ADD R0,R4 ;TRY ADD: R4=177777
2927 007306 005204 INC R4 ;R4=0
```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 68
CFKAAC.P11 18-OCT-78 11:01 T113 TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0 SEQ 0080
2925 007310 001404 BEQ DOP0D ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2926 ; ; ; ; ; <====
2927 ; ; ; ; ; <====
2928 ; ; ; ; ; <====
2929 ; ; ; ; ; <====
2930 007312 012742 000207 MOV #207,-(R2) ;MOVE TO MAILBOX # ***** 207 *****
2931 007316 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2932 007320 000000 HALT ;RESULT OF ADD INCORRECT
2933 007322 160004 DOP0D: SUB R0,R4 ;17401=R4
2934 007324 105404 NEGB R4 ;R4=177777
2935 007326 005204 INC R4 ;RD=0
2936 007330 001404 BEQ TST114 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2937 ; ; ; ; ; <====
2938 ; ; ; ; ; <====
2939 ; ; ; ; ; <====
2940 ; ; ; ; ; <====
2941 007332 012742 000210 MOV #210,-(R2) ;MOVE TO MAILBOX # ***** 210 *****
2942 007336 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2943 007340 000000 HALT ;RESULT OF SUB INCORRECT
2944 ; ; ; ; ; <====
2945 ; ; ; ; ; <====
; OR SEQUENCE ERROR

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 69
CFKAAC.P11 18-OCT-78 11:01 T113 TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0 SEQ 0081
2946 ;*****
2947 ; THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS. IT SETS
2948 ; DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
2949 ;*****
2950 ;TEST 114 TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
2951 ;*****
2952 TST114: INC (R2) ;UPDATE TEST NUMBER
2953 CMP #114,(R2) ;SEQUENCE ERROR?
2954 BNE TST115-10 ;BR TO ERROR HALT ON SEQ ERROR
2955 CLR R0 ;R0=0
2956 007342 005212 000114 CLR (R0) ;LOC. 0=0
2957 007344 022712 BNE (R0) ;LOC. 0=377
2958 007350 001024 COMB (R0)+ ;LOC. 0=400 R0=2
2959 007352 005000 INC R0 ;R0=-2
2960 007354 005010 NEG R0 ;TRY ADD 0,3; LOC. 0=376
2961 007356 105110 ADD R0,#0 ;CC=0001?
2962 007362 060037 000000 BMI DOP03A
2963 007370 100403 BEQ DOP03A
2964 007372 001402 BVS DOP03A
2965 007374 102401 BVS DOP03A
2966 007376 103404 BCS DOP03B
2967 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2968 ; ; ; ; ; <====
2969 ; ; ; ; ; <====
2970 ; ; ; ; ; <====
2971 007400 DOP03A: MOV #211,-(R2) ;MOVE TO MAILBOX # ***** 211 *****
2972 007400 012742 000211 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2973 007404 005242 HALT ;CC'S NOT SET CORRECTLY
2974 007406 000000 DOP03B: COMB #0 ;LOC. 0=1
2975 007410 105137 000000 BEQ #0 ;LOC. 0=0
2976 007414 005337 000000 DEC #0
2977 007420 001404 BEQ TST115 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2978 ; ; ; ; ; <====
2979 ; ; ; ; ; <====
2980 ; ; ; ; ; <====
2981 ; ; ; ; ; <====
2982 007422 012742 000212 MOV #212,-(R2) ;MOVE TO MAILBOX # ***** 212 *****
2983 007426 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2984 007430 000000 HALT ;DATA RESULT INCORRECT
2985 ; ; ; ; ; <====
; OR SEQUENCE ERROR

```

```

2986 *****
2987 ;
2988 ; THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
2989 ; R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY. COMPARE INSTRUCTIONS ARE
2990 ; THEN EXECUTED AND CHECKED. FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
2991 *****
2992 ;
2993 ;
2994 ;
2995 ;
2996 007432 005212 000115 TST115: INC (R2) ;UPDATE TEST NUMBER
2997 007434 022712 ;SEQUENCE ERROR?
2998 007440 001042 ;BR TO ERROR HALT ON SEQ ERROR
2999 007442 005000 ;R0=0
3000 007444 005004 ;R4=0
3001 007446 005204 ;R4=1
3002 007452 003004 ;TRY COMPARE R4 TO R0
3003 ;
3004 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3005 ; CONDITIONAL BRANCH INST. AND <====
3006 ; REPLACE THE MOVE INSTRUCTION <====
3007 ; WHICH FOLLOWS W/ 773 ***** <====
3008 007454 012742 000213 MOV #213,-(R2) ;MOVE TO MAILBOX # ***** 213 *****
3009 007460 005242 ;SET MSGTYP TO FATAL ERROR
3010 007462 000000 ;CC'S NOT CORRECT FOR CMP
3011 007464 020004 DNM1: RO,R4 ;TRY COMPARE R0 TO R4
3012 007466 002404 HALT DNM2
3013 ;
3014 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3015 ; CONDITIONAL BRANCH INST. AND <====
3016 ; REPLACE THE MOVE INSTRUCTION <====
3017 ; WHICH FOLLOWS W/ 765 ***** <====
3018 007470 012742 000214 MOV #214,-(R2) ;MOVE TO MAILBOX # ***** 214 *****
3019 007472 005242 ;SET MSGTYP TO FATAL ERROR
3020 007500 005200 ;CC'S NOT CORRECT FOR CMP
3021 007502 020400 DNM2: RO ;R0=1
3022 007504 001404 CMP R4,R0 ;TRY COMPARE R4=1 TO R0=1
3023 BEQ DNM3
3024 ;
3025 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3026 ; CONDITIONAL BRANCH INST. AND <====
3027 ; REPLACE THE MOVE INSTRUCTION <====
3028 ; WHICH FOLLOWS W/ 756 ***** <====
3029 007506 012742 000215 MOV #215,-(R2) ;MOVE TO MAILBOX # ***** 215 *****
3030 007512 005242 ;SET MSGTYP TO FATAL ERROR
3031 007514 000000 ;CC'S NOT CORRECT (Z=1) FOR CMP
3032 007516 005000 DNM3: RO ;R0=0
3033 007520 005100 COM ;R0=177777
3034 007522 005004 CLR R4 ;R4=0
3035 007524 030004 BIT RO,R4 ;TRY BIT R0 TO R4
3036 BEQ DNM4
3037 ;
3038 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3039 ; CONDITIONAL BRANCH INST. AND <====
3040 ; REPLACE THE MOVE INSTRUCTION <====
3041 ; WHICH FOLLOWS W/ 745 ***** <====
3042 007530 012742 000216 MOV #216,-(R2) ;MOVE TO MAILBOX # ***** 216 *****
3043 007532 005242 ;SET MSGTYP TO FATAL ERROR
3044 007534 000000 HALT DNM4: DEC R4 ;CC'S NOT CORRECT FOR BIT
3045 007540 005304 ;R4=177777
    
```

```

3042 007542 030004 BIT RO,R4 ;TRY BIT AGAIN
3043 007544 100404 BMI TST116
3044 ;
3045 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3046 ; CONDITIONAL BRANCH INST. AND <====
3047 ; REPLACE THE MOVE INSTRUCTION <====
3048 ; WHICH FOLLOWS W/ 736 ***** <====
3049 007546 012742 000217 MOV #217,-(R2) ;MOVE TO MAILBOX # ***** 217 *****
3050 007552 005242 ;SET MSGTYP TO FATAL ERROR
3051 007554 000000 HALT ;CC'S NOT CORRECT FOR BIT
3052 ; OR SEQUENCE ERROR
3053 *****
3054 ;
3055 ; THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
3056 ; IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
3057 *****
3058 ;
3059 ;
3060 ;
3061 ;
3062 ;
3063 ;
3064 ;
3065 ;
3066 ;
3067 ;
3068 ;
3069 ;
3070 ;
3071 ;
3072 ;
3073 ;
3074 ;
3075 ;
3076 007612 012742 000220 DNM03A: MOV #220,-(R2) ;MOVE TO MAILBOX # ***** 220 *****
3077 007614 005242 ;SET MSGTYP TO FATAL ERROR
3078 007620 000000 HALT ;CC'S NOT SET CORRECTLY
3079 007622 005300 DNM03B: DEC R0
3080 007624 001002 BNE DNM03C
3081 007626 005210 INC (R0)
3082 007630 001404 BEQ TST117
3083 ;
3084 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3085 ; CONDITIONAL BRANCH INST. AND <====
3086 ; REPLACE THE MOVE INSTRUCTION <====
3087 ; WHICH FOLLOWS W/ 756 ***** <====
3088 007632 012742 000221 DNM03C: MOV #221,-(R2) ;MOVE TO MAILBOX # ***** 221 *****
3089 007634 005242 ;SET MSGTYP TO FATAL ERROR
3090 007636 000000 HALT ;DATA INCORRECTLY MODIFIED BY CMP
3091 007640 000000 ; OR SEQUENCE ERROR
3092
    
```

```
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104 007642 005212  
3105 007644 022712 000117  
3106 007650 001007  
3107 007652 005060  
3108 007654 005100  
3109 007656 005004  
3110 007660 005214  
3111 007662 005214  
3112 007664 061400  
3113 007666 001404  
3114  
3115  
3116  
3117 007670 012742 000222  
3118 007674 005242  
3119 007676 000000  
3120
```

```
*****  
; THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS. R0 IS SET TO -1  
; AND LOC 0 TO 1. R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.  
; IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE  
; RESULTS VERIFIED.  
*****  
; TEST 117 TEST MODE 1 W/ DOP INST  
; *****  
TST117: INC (R2) ; UPDATE TEST NUMBER  
CMP #117,(R2) ; SEQUENCE ERROR?  
BNE TST120-10 ; BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ; R0=0  
COM R0 ; R0=-177777  
CLR R4 ; R4=0  
CLR (R4) ; LOC 0=0  
INC (R4) ; LOC 0=1  
ADD (R4),R0 ; TRY ADD SOURCE MODE 1  
BEQ TST120  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 771 ***** <====
```

```
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131 007700 005212  
3132 007702 022712 000120  
3133 007706 001007  
3134 007710 005000  
3135 007712 005010  
3136 007714 005110  
3137 007716 005004  
3138 007720 151004  
3139 007722 105104  
3140 007724 001404  
3141  
3142  
3143  
3144  
3145 007726 012742 000223  
3146 007732 005242  
3147 007734 000000  
3148
```

```
*****  
; THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS  
; EVEN BYTES. LOC. 0 IS SET TO -1 AND R4 IS CLEARED. THEN R4 IS  
; SET TO -1 USING A BISB THRU R0 WITH MODE 1.  
*****  
; TEST 120 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.  
; *****  
TST120: INC (R2) ; UPDATE TEST NUMBER  
CMP #120,(R2) ; SEQUENCE ERROR?  
BNE TST121-10 ; BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ; R0=0  
CLR (R0) ; LOC 0=0  
COM (R0) ; LOC 0=177777  
CLR R4 ; R4=0  
BISB (R0),R4 ; TRY MODE 1- EVEN BYTE W/ DOP  
COMB R4 ; R4=0  
BEQ TST121  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 771 ***** <====
```

```
3149 ;*****
3150 ;
3151 ; THIS TEST VERIFIES MODE 1 DOP NON-MODIFYING INSTRUCTIONS
3152 ; WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO -1 AND R0 IS CLEARED
3153 ; AND USED AS THE ADDRESSING REGISTER. R4 IS SET TO 377 AND A
3154 ; MODE 1,0 CMPB INSTRUCTION IS USED THE RESULTS VERIFIED.
3155 ;*****
3156 ;
3157 ;
3158 ; TEST 121 TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
3159 ;*****
3160 007736 005212
3161 007740 022712 000121
3162 007744 001007
3163 007746 005000
3164 007750 005010
3165 007752 005110
3166 007754 005004
3167 007756 105104
3168 007760 121004
3169 007762 001404
3170
3171
3172
3173
3174 007764 012742 000224
3175 007770 005242
3176 007772 000000
3177
```

```

;*****
;
; UPDATE TEST NUMBER
; SEQUENCE ERROR?
; BR TO ERROR HALT ON SEQ ERROR
; R0=0
; LOC 0=0
; LOC 0=17777
; R4=0
; R4=377
; TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 771
; *****
; OR SEQUENCE ERROR

```

```
3178 ;*****
3179 ;
3180 ; THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
3181 ; WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
3182 ; R4 IS SET TO -1. MOVBS ARE USED TO MOVE BYTE 0 TO R4. THIS
3183 ; VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
3184 ; FUNCTION WITH MODE 0.
3185 ; THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
3186 ; THE LOGIC FOR COMPLEMENTARY DATA.
3187 ; THIS TEST EXERCISES UNIQUE MICROCODE.
3188 ;*****
3189 ;
3190 ; TEST 122 TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
3191 ;*****
3192 007774 005212
3193 007776 022712 000122
3194 010002 001020
3195 010004 005000
3196 010006 005010
3197 010010 105110
3198 010012 005110
3199 010014 005004
3200 010016 005104
3201 010020 111004
3202 010022 005704
3203 010024 001404
3204
3205
3206
3207
3208
3209 010026 012742 000225
3210 010032 005242
3211 010034 000000
3212 010036 005110
3213 010040 111004
3214 010042 100404
3215
3216
3217
3218
3219 010044 012742 000226
3220 010050 005242
3221 010052 000000
3222
```

```

;*****
;
; UPDATE TEST NUMBER
; SEQUENCE ERROR?
; BR TO ERROR HALT ON SEQ ERROR
; R0=0
; LOC 0=0
; LOC 0=177400
; R4=0
; R4=17777
; CHECK SIGN OF WORD
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 767
; *****
; OR SEQUENCE ERROR

```

```

;*****
;
; UPDATE TEST NUMBER
; SEQUENCE ERROR?
; BR TO ERROR HALT ON SEQ ERROR
; R0=0
; LOC 0=17777
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 760
; *****
; OR SEQUENCE ERROR

```



```
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231  
3232  
3233  
3234 010054 005212  
3235 010056 022712 000123  
3236 010062 001010  
3237 010064 005000  
3238 010066 005010  
3239 010070 005004  
3240 010072 005204  
3241 010074 105114  
3242 010076 151410  
3243 010100 005210  
3244 010102 001404  
3245  
3246  
3247  
3248  
3249 010104 012742 000227  
3250 010110 005242  
3251 010112 000000  
3252
```

```
*****  
; 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 BISSB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.  
; THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.  
*****  
; TEST 123 TEST MODE 1-ODD BYTE W/ DOP INSTS.  
*****  
TST123: INC (R2) ;UPDATE TEST NUMBER  
CMP #123,(R2) ;SEQUENCE ERROR?  
BNE TST124-10 ;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  
BISSB (R4),(R0) ;TRY TO BISS LOW ORDER BITS W/ MODE 1  
INC (R0) ;CHECK RESULT  
BEQ TST124  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 770 <====
```

```
3253  
3254  
3255  
3256  
3257  
3258  
3259  
3260  
3261  
3262  
3263  
3264 010114 005212  
3265 010116 022712 000124  
3266 010122 001010  
3267 010124 005000  
3268 010126 005010  
3269 010130 005110  
3270 010132 012004  
3271 010134 005204  
3272 010136 001404  
3273  
3274  
3275  
3276  
3277 010140 012742 000230  
3278 010144 005242  
3279 010146 000000  
3280 010150 005300  
3281 010152 005300  
3282 010154 001404  
3283  
3284  
3285  
3286 010156 012742 000231  
3287 010162 005242  
3288 010164 000000  
3290
```

```
*****  
; THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS. LOC. 0 IS SET TO -1.  
; R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0  
; TO R7. THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER  
; IS CHECKED.  
*****  
; TEST 124 TEST MODE 2 W/ DOP INSTS.  
*****  
TST124: INC (R2) ;UPDATE TEST NUMBER  
CMP #124,(R2) ;SEQUENCE ERROR?  
BNE TST125-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
COM (R0) ;LOC. 0=177777  
MOV (R0)+,R4 ;TRY MOVE MODE 2,0  
INC R4 ;CHECK R4  
BEQ DOP2  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 772 <====  
; MOVE TO MAILBOX # ***** 230 *****  
; SET MSGTYP TO FATAL ERROR  
; RESULT OF MOV INST INCORRECT  
DOP2: DEC R0 ;TEST R0 AFTER MODE 2  
DEC R0  
BEQ TST125  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====  
; MOVE TO MAILBOX # ***** 231 *****  
; SET MSGTYP TO FATAL ERROR  
; REGISTER NOT INCREMENTED IN MODE 2  
; OR SEQUENCE ERROR
```

```
3291 ;*****  
3292 ; THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS  
3293 ; EVEN BYTES. LOC. 0 IS SET TO -1. RO IS CLEARED AND USED AS THE  
3294 ; ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING  
3295 ; BYTE 0 DATA AND A BICB. UNIQUE IN THIS TEST IS USE OF THE  
3296 ; SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION. THE SOURCE AND  
3297 ; DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.  
3298 ;*****  
3299 ;*****  
3300 ;*****  
3301 ;*****  
3302 ;*****  
3303 ;*****  
3304 010166 005212 000125 TST125: INC (R2) ;UPDATE TEST NUMBER  
3305 010170 022712 ;SEQUENCE ERROR?  
3306 010174 001016 BNE #125,(R2) ;BR TO ERROR HALT ON SEQ ERROR  
3307 010176 005000 CLR R0 ;RO=0  
3308 010200 010010 MOV R0,(R0) ;LOC. 0=0  
3309 010202 005110 COM (R0) ;LOC. 0=177777  
3310 010204 142010 BICB (R0)+,(R0) ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB  
3311 010206 105737 TSTB #1 ;CHECK RESULT  
3312 010212 001404 BEQ DOPB2A ;  
3313 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
3314 ; CONDITIONAL BRANCH INST. AND <=====  
3315 ; REPLACE THE MOVE INSTRUCTION <=====  
3316 ; WHICH FOLLOWS W/ 771 <=====  
3317 010214 012742 000232 MOV #232,-(R2) ;MOVE TO MAILBOX # ***** 232 *****  
3318 010220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3319 010222 000000 HALT ;BICB DESTINATION INCORRECT  
3320 010224 105137 000000 DOPB2A: COMB #0 ;CHECK BICB SOURCE  
3321 010230 001404 BEQ TST126 ;  
3322 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
3323 ; CONDITIONAL BRANCH INST. AND <=====  
3324 ; REPLACE THE MOVE INSTRUCTION <=====  
3325 ; WHICH FOLLOWS W/ 762 <=====  
3326 010232 012742 000233 MOV #233,-(R2) ;MOVE TO MAILBOX # ***** 233 *****  
3327 010236 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3328 010240 000000 HALT ;BICB SOURCE INCORRECTLY CHANGED  
3329 ; OR SEQUENCE ERROR
```

```
3330 ;*****  
3331 ; THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE  
3332 ; ODD BYTES. RO IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.  
3333 ; A MODE 2 MOVB USES RO TO MOVE BYTE 1 TO R4. AN INCREMENT  
3334 ; IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.  
3335 ;*****  
3336 ;*****  
3337 ;*****  
3338 ;*****  
3339 ;*****  
3340 010242 005212 000126 TST126: INC (R2) ;UPDATE TEST NUMBER  
3341 010244 022712 ;SEQUENCE ERROR?  
3342 010250 001017 BNE #126,(R2) ;BR TO ERROR HALT ON SEQ ERROR  
3343 010252 005000 CLR R0 ;RO=0  
3344 010254 005004 CLR R4 ;R4=0  
3345 010256 005010 MOV R0,(R0) ;LOC. 0=0  
3346 010260 005110 COM (R0) ;LOC. 0=177777  
3347 010262 105120 COMB (R0)+,R4 ;LOC 0=177400; RO=1  
3348 010264 112004 MOVB R4 ;TRY DOP MODE 2 W/ ODD BYTE  
3349 010266 005204 INC R4 ;CHECK RESULT OF MOVB  
3350 010270 001404 BEQ DOPB2B ;  
3351 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
3352 ; CONDITIONAL BRANCH INST. AND <=====  
3353 ; REPLACE THE MOVE INSTRUCTION <=====  
3354 ; WHICH FOLLOWS W/ 770 <=====  
3355 010272 012742 000234 MOV #234,-(R2) ;MOVE TO MAILBOX # ***** 234 *****  
3356 010276 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3357 010300 000000 HALT ;RESULT OF MOVB INCORRECT  
3358 010302 005740 DOPB2B: TST -(R0) ;BUMP RO DOWN BY 2  
3359 010304 005700 TST R0 ;CHECK RO  
3360 010306 001404 BEQ TST127 ;  
3361 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
3362 ; CONDITIONAL BRANCH INST. AND <=====  
3363 ; REPLACE THE MOVE INSTRUCTION <=====  
3364 ; WHICH FOLLOWS W/ 761 <=====  
3365 010310 012742 000235 MOV #235,-(R2) ;MOVE TO MAILBOX # ***** 235 *****  
3366 010314 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3367 010316 000000 HALT ;MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY  
3368 ; OR SEQUENCE ERROR
```

```
*****
THIS TEST VERIFIES MODE 3 DOUBLE-OPERAND INSTRUCTIONS.
LOC. 0 IS LOADED WITH ALTERNATING ZEROES AND ONES; AND R0 IS LOADED
WITH ALTERNATING ONES AND ZEROES. A MODE 3 BIS IS USED TO SET R0
TO -1 BY USING LOC. 0 AS THE SOURCE TO BIS THE ZEROES IN R0. THE
RESULT IS TESTED BY INCREMENTING R0 AND CHECKING FOR ZERO.
*****
TEST 127 TEST MODE 3 W/ DOP INST.
*****
TST127: INC (R2) ;UPDATE TEST NUMBER
          CMP #127,(R2) ;SEQUENCE ERROR?
          BNE TST130-10 ;BR TO ERROR HALT ON SEQ ERROR
          MOV #052525,@#0 ;MOVE 52525 TO LOC. 0
          MOV #125252,R0 ;SET ALT. ONE AND ZERO IN R0
          BIS @#0,R0 ;TRY TO SET ALL OTHER BITS W/ MODE 3
          INC R0 ;TEST RESULT
          BEQ TST130 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 767 <====
          MOV #236,-(R2) ;MOVE TO MAILBOX # ***** 236 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;BIS W/ MODE 3 INCORRECT RESULT
          ; OR SEQUENCE ERROR
*****
THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS WHICH
ADDRESS EVEN BYTES. BYTE 0 IS SET TO ALTERNATING 1'S AND 0'S; BYTE 1,
ALTERNATING 0'S AND 1'S. R0 IS CLEARED AND A BISB IS USED TO
SET THE LOW BYTE OF R0 TO 252.
*****
TEST 130 TEST MODE 3 - EVEN BYTE W/ DOP INST.
*****
TST130: INC (R2) ;UPDATE TEST NUMBER
          CMP #130,(R2) ;SEQUENCE ERROR?
          BNE TST131-10 ;BR TO ERROR HALT ON SEQ ERROR
          MOV #52652,@#0 ;MOVE 1'S AND 0'S PATTERN TO LOC. 0
          CLR R0 ;R0=0
          BISB @#0,R0 ;TRY R0=252 W/ MODE 3 - EVEN BYTE
          CMP #252,R0 ;BISB W/ EVEN BYTE SUCCESSFUL?
          BEQ TST131 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 767 <====
          MOV #237,-(R2) ;MOVE TO MAILBOX # ***** 237 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;BISB W/ MODE 3 - EVEN BYTE FAILED
          ; OR SEQUENCE ERROR
*****
```

```
*****
THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
WHICH ADDRESS ODD BYTES. THE SAME PROCEDURE USED IN PREVIOUS
TEST IS USED HERE THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
THE EXPECTED RESULT IS: R0 = 125.
*****
TEST 131 TEST MODE 3 - ODD BYTE W/ DOP INST.
*****
TST131: INC (R2) ;UPDATE TEST NUMBER
          CMP #131,(R2) ;SEQUENCE ERROR?
          BNE TST132-10 ;BR TO ERROR HALT ON SEQ ERROR
          MOV #52652,@#0 ;MOVE 1'S AND 0'S PATTERN TO LOC 0
          CLR R0 ;R0=0
          BISB @#1,R0 ;TRY R0=152 W/ MODE 3 - ODD BYTE
          CMP #125,R0 ;R0=125?
          BEQ TST132 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 767 <====
          MOV #240,-(R2) ;MOVE TO MAILBOX # ***** 240 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;BISB W/ MODE 3 - ODD BYTE FAILED
          ; OR SEQUENCE ERROR
*****
TEST 132 TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
*****
TST132: INC (R2) ;UPDATE TEST NUMBER
          CMP #132,(R2) ;SEQUENCE ERROR?
          BNE TST133-10 ;BR TO ERROR HALT ON SEQ ERROR
          CLR R0 ;R0=0
          COMB R0 ;R0=377
          +SECISEV ;SET C AND V BITS
          BITB #200,R0 ;TRY DOPNM DEST. MODE 0-BYTE
          BEQ DNMB0A ;BR TO ERROR IF Z BIT SET
          BVS DNMB0A ;BR TO ERROR IF V BIT SET
          BNC DNMB0A ;BR TO ERROR IF C BIT CLEAR.
          BMI DNMB0B ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 767 <====
          DNMB0A: MOV #241,-(R2) ;MOVE TO MAILBOX # ***** 241 *****
                  INC -(R2) ;SET MSGTYP TO FATAL ERROR
          DNMB0B: HALT ;CCS INCORRECT
                  COMB R0 ;CHECK DESTINATION DATA
                  BEQ TST133 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
                  ; CONDITIONAL BRANCH INST. AND <====
                  ; REPLACE THE MOVE INSTRUCTION <====
                  ; WHICH FOLLOWS W/ 767 <====
          MOV #242,-(R2) ;MOVE TO MAILBOX # ***** 242 *****
*****
```

```
3478 010540 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3479 010542 000000      HALT          ;DEST. DATA MODIFIED
3480                                     ;OR SEQUENCE ERROR
3481
3482 ;*****
3483 ;TEST 133 TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3484 ;*****
3485 TST133: INC      (R2)      ;UPDATE TEST NUMBER
3486 010544 005212 000133  CMP      #133,(R2)  ;SEQUENCE ERROR?
3487 010552 001017      BNE      TST134-10 ;BR TO ERROR HALT ON SEQ ERROR
3488 010554 005000      CLR      R0        ;R0=0
3489 010556 005010      CLR      (R0)      ;LOC=0
3490 010560 000241      CLC          ;CLEAR C BIT
3491 010562 032710 177777 BIT      #177777,(R0) ;TRY DOPNM DEST. MODE 1
3492 010566 100403      BMI      DNM1A     ;BR TO ERROR IF N BIT SET
3493 010570 102402      BVS      DNM1A     ;BR TO ERROR IF V BIT SET
3494 010572 103401      BCS      DNM1A     ;BR TO ERROR IF C BIT SET
3495 010574 001404      BEQ      DNM1B     ;
3496                                     ;
3497                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3498                                     ; CONDITIONAL BRANCH INST. AND <====
3499                                     ; REPLACE THE MOVE INSTRUCTION <====
3500                                     ; WHICH FOLLOWS W/ 767 <====
3501
3502 010576 000243  DNM1A:  MOV      #243,-(R2) ;MOVE TO MAILBOX # ***** 243 *****
3503 010578 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3504 010604 000000      HALT          ;COND. CODES INCORRECT
3505 010606 005710  DNM1B:  TST      (R0)      ;CHECK TEST DATA
3506 010610 001404      BEQ      TST134     ;
3507                                     ;
3508                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3509                                     ; CONDITIONAL BRANCH INST. AND <====
3510                                     ; REPLACE THE MOVE INSTRUCTION <====
3511                                     ; WHICH FOLLOWS W/ 761 <====
3512
3513 010612 012742 000244  MOV      #244,-(R2) ;MOVE TO MAILBOX # ***** 244 *****
3514 010616 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3515 010620 000000      HALT          ;DESTINATION DATA MODIFIED
3516                                     ;OR SEQUENCE ERROR
3517
3518 ;*****
3519 ;TEST 134 TEST DEST. MODE 2 W/ DOP NON-MODIFYING INST.
3520 ;*****
3521 TST134: INC      (R2)      ;UPDATE TEST NUMBER
3522 010624 022712 000134  CMP      #134,(R2)  ;SEQUENCE ERROR?
3523 010630 001027      BNE      TST135-10 ;BR TO ERROR HALT ON SEQ ERROR
3524 010632 005000      CLR      R0        ;R0=0
3525 010634 005010      CLR      (R0)      ;LOC=0
3526 010636 052710 125252 BIS      #125252,(R0) ;LOC=0=125252
3527 010640 032710 077777 BIT      #125252,(R0) ;TRY DOPNM INST W/ MODE 2
3528 010644 102402      BVS      DNM2A     ;BR TO ERROR IF V BIT SET
3529 010650 001401      BEQ      DNM2A     ;BR TO ERROR IF Z-BIT SET
3530 010652 100004      BPL      DNM2B     ;
3531                                     ;
3532                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3533                                     ; CONDITIONAL BRANCH INST. AND <====
3534                                     ; REPLACE THE MOVE INSTRUCTION <====
3535                                     ; WHICH FOLLOWS W/ 76 <====
3536
3537 010654 012742 000245  DNM2A:  MOV      #245,-(R2) ;MOVE TO MAILBOX # ***** 245 *****
3538 010654 012742 000245  MOV      #245,-(R2) ;MOVE TO MAILBOX # ***** 245 *****
```

```
3539 010660 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3540 010662 000000      HALT          ;COND. CODES INCORRECT
3541 010664 005300  DNM2B:  DEC      R0        ;DECREMENT R0 TO CHECK IT.
3542 010666 005300      DEC      R0
3543 010670 001404      BEQ      DNM2D     ;
3544                                     ;
3545                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3546                                     ; CONDITIONAL BRANCH INST. AND <====
3547                                     ; REPLACE THE MOVE INSTRUCTION <====
3548                                     ; WHICH FOLLOWS W/ 760 <====
3549
3550 010672 012742 000246  DNM2C:  MOV      #246,-(R2) ;MOVE TO MAILBOX # ***** 246 *****
3551 010674 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3552 010676 000000      HALT          ;MODE 2 REGISTER NOT INCREMENTED BY 2
3553 010700 000000      BEQ      TST135     ;CHECK DEST. DATA
3554 010702 022710 125252  DNM2D:  CMP      #125252,(R0) ;
3555 010706 001404      BEQ      TST135     ;
3556                                     ;
3557                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3558                                     ; CONDITIONAL BRANCH INST. AND <====
3559                                     ; REPLACE THE MOVE INSTRUCTION <====
3560                                     ; WHICH FOLLOWS W/ 751 <====
3561
3562 010710 012742 000247  MOV      #247,-(R2) ;MOVE TO MAILBOX # ***** 247 *****
3563 010714 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3564 010716 000000      HALT          ;DEST. DATA MODIFIED
3565                                     ;OR SEQUENCE ERROR
3566
3567 ;*****
3568 ;TEST 135 TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
3569 ;*****
3570 TST135: INC      (R2)      ;UPDATE TEST NUMBER
3571 010720 005212 000135  CMP      #135,(R2)  ;SEQUENCE ERROR?
3572 010722 022712      BNE      TST136-10 ;BR TO ERROR HALT ON SEQ ERROR
3573 010726 001051      CLR      R0        ;R0=0
3574 010730 005000      CLR      (R0)      ;LOC=0
3575 010732 005010      CLR      (R0)      ;LOC=0=52652
3576 010734 052710 052652 BIS      #52652,(R0) ;SET C AND V BITS
3577 010740 000263      +SECISEV #201,(R0)+ ;TRY DOPNM W/ MODE 2 EVEN BYTE
3578 010742 000263      BIT      DNM2A     ;BR TO ERROR IF Z-BIT SET
3579 010746 132720 000201  BEQ      DNM2A     ;BR TO ERROR IF C-BIT CLEAR
3580 010750 103002      BVS      DNM2A     ;BR TO ERROR IF V-BIT SET
3581 010752 102401      BEQ      DNM2B     ;
3582 010754 100404      BMI      DNM2B     ;
3583                                     ;
3584                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3585                                     ; CONDITIONAL BRANCH INST. AND <====
3586                                     ; REPLACE THE MOVE INSTRUCTION <====
3587                                     ; WHICH FOLLOWS W/ 765 <====
3588
3589 010756 012742 000250  DNM2A:  MOV      #250,-(R2) ;MOVE TO MAILBOX # ***** 250 *****
3590 010758 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3591 010762 000000      HALT          ;COND. CODES INCORRECT
3592 010764 005300  DNM2B:  DEC      R0        ;CHECK DEST. REGISTER.
3593 010766 005300      DEC      R0
3594 010770 001404      BEQ      DNM2C     ;
3595                                     ;
3596                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3597                                     ; CONDITIONAL BRANCH INST. AND <====
3598                                     ; REPLACE THE MOVE INSTRUCTION <====
3599                                     ; WHICH FOLLOWS W/ 757 <====
3600
3601 010772 012742 000251  MOV      #251,-(R2) ;MOVE TO MAILBOX # ***** 251 *****
3602 010774 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3603 011000 000000      HALT          ;DEST. REGISTER NOT INCREMENTED BY 1
```

```
3590 011002 005200 000201 DNMBC: INC R0 ;R0=1  
3591 011004 132710 BITB #201,(R0)+ ;TRY DOPNM INST. W/MODE 2-ODD BYTE  
3592 011010 001402 BEQ DNMBC2 ;BR TO ERROR IF Z-BIT SET  
3593 011012 102401 BVS DNMBC2 ;BR TO ERROR IF V-BIT SET  
3594 011014 100004 BPL DNMBC2E ;BR TO ERROR IF C-BIT CLEAR  
3595 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3596 ; CONDITIONAL BRANCH INST. AND <====  
3597 ; REPLACE THE MOVE INSTRUCTION <====  
3598 ; WHICH FOLLOWS W/ 745 <====  
3599 011016 012742 000252 DNMBC2D: MOV #252,-(R2) ;MOVE TO MAILBOX # ***** 252 *****  
3600 011022 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3601 011024 000000 HALT ;COND. CODES INCORRECT  
3602 011026 005300 DNMBC2E: R0 ;DEST. REGISTER NOT INCREMENTED BY 1  
3603 011030 005300 DEC R0 ;CHECK DEST. DATA IS UNMODIFIED  
3604 011032 001404 BEQ DNMBC2F  
3605 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3606 ; CONDITIONAL BRANCH INST. AND <====  
3607 ; REPLACE THE MOVE INSTRUCTION <====  
3608 ; WHICH FOLLOWS W/ 736 <====  
3609 011034 012742 000253 MOV #253,-(R2) ;MOVE TO MAILBOX # ***** 253 *****  
3610 011040 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3611 011042 000000 HALT ;DEST. REGISTER NOT INCREMENTED BY 1  
3612 011044 022710 052652 DNMBC2F: CMP #52652,(R0) ;CHECK DEST. DATA IS UNMODIFIED  
3613 011050 001404 BEQ TST136  
3614 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3615 ; CONDITIONAL BRANCH INST. AND <====  
3616 ; REPLACE THE MOVE INSTRUCTION <====  
3617 ; WHICH FOLLOWS W/ 727 <====  
3618 011052 012742 000254 MOV #254,-(R2) ;MOVE TO MAILBOX # ***** 254 *****  
3619 011056 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3620 011060 000000 HALT ;DEST. DATA WAS MODIFIED.  
3621 ; OR SEQUENCE ERROR  
3622  
3623  
3624  
3625  
3626  
3627  
3628 011062 005212 ;*****  
3629 011064 022712 ;TEST 136 TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST. <====  
3630 011070 001050 ;*****  
3631 011072 005000 TST136: INC (R2) ;UPDATE TEST NUMBER  
3632 011074 005010 CMP #136,(R2) ;SEQUENCE ERROR?  
3633 011076 052710 CLR R0 ;BR TO ERROR HALT ON SEQ ERROR  
3634 011102 105100 CLR (R0) ;R0=0  
3635 011104 005200 BIS #125125,(R0) ;LOC. 0=0  
3636 011106 005010 COMB R0 ;LOC. 0=125125  
3637 011108 005200 INC R0 ;R0=377  
3638 011110 005253 CLR (R0) ;R0=400  
3639 011112 132730 +SECISET ;C-BIT=V-BIT=1  
3640 011114 001403 BITB #201,(R0)+ ;TRY DOPNM W/MODE 3-EVEN BYTE  
3641 011116 102402 BEQ DNMBC3A ;BR TO ERROR IF Z BIT SET  
3642 011118 103001 BVS DNMBC3A ;BR TO ERROR IF V BIT SET  
3643 011120 103001 BCC DNMBC3A ;BR TO ERROR IF C BIT CLEAR  
3644 011122 100004 BPL DNMBC3B ;BR TO ERROR IF C-BIT CLEAR  
3645 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3646 ; CONDITIONAL BRANCH INST. AND <====  
3647 ; REPLACE THE MOVE INSTRUCTION <====
```

```
3648 011126 012742 000255 DNMBC3A: MOV #255,-(R2) ;MOVE TO MAILBOX # ***** 255 *****  
3649 011132 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3650 011134 000000 HALT ;COND. CODES INCORRECT  
3651 011136 022700 000402 DNMBC3B: CMP #402,R0 ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN  
3652 011142 001404 BEQ DNMBC3C  
3653 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3654 ; CONDITIONAL BRANCH INST. AND <====  
3655 ; REPLACE THE MOVE INSTRUCTION <====  
3656 ; WHICH FOLLOWS W/ 753 <====  
3657 011144 012742 000256 MOV #256,-(R2) ;MOVE TO MAILBOX # ***** 256 *****  
3658 011150 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3659 011152 000000 HALT ;DEST. REGISTER NOT INCREMENTED BY 2  
3660 011154 005200 DNMBC3C: INC R0 ;R0=404  
3661 011156 005200 INC R0  
3662 011160 132730 BITB #201,(R0)+ ;TRY DOPNM DEST MODE 3-BYTE(ODD)  
3663 011162 102402 BEQ DNMBC3D ;BR TO ERROR IF Z BIT SET  
3664 011164 102401 BVS DNMBC3D ;BR TO ERROR IF V BIT SET  
3665 011170 100404 BMI DNMBC3E ;BR TO ERROR IF V-BIT SET  
3666 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3667 ; CONDITIONAL BRANCH INST. AND <====  
3668 ; REPLACE THE MOVE INSTRUCTION <====  
3669 ; WHICH FOLLOWS W/ 740 <====  
3670 011172 012742 000257 DNMBC3D: MOV #257,-(R2) ;MOVE TO MAILBOX # ***** 257 *****  
3671 011176 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3672 011200 000000 HALT ;COND. CODES INCORRECT  
3673 011202 005004 DNMBC3E: R4 ;R=0  
3674 011204 022714 CMP #125125,(R4) ;CHECK DEST. DATA  
3675 011210 001404 BEQ TST137  
3676 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3677 ; CONDITIONAL BRANCH INST. AND <====  
3678 ; REPLACE THE MOVE INSTRUCTION <====  
3679 ; WHICH FOLLOWS W/ 730 <====  
3680 011212 012742 000260 MOV #260,-(R2) ;MOVE TO MAILBOX # ***** 260 *****  
3681 011216 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3682 011220 000000 HALT ;DEST. DATA MODIFIED  
3683 ; OR SEQUENCE ERROR  
3684  
3685  
3686  
3687  
3688 ;*****  
3689 011222 005212 ;TEST 137 TEST DEST. MODE 4 W/DOP NON-MODIFYING INST. <====  
3690 011224 022712 ;*****  
3691 011230 001033 TST137: INC (R2) ;UPDATE TEST NUMBER  
3692 011232 005000 CMP #137,(R2) ;SEQUENCE ERROR?  
3693 011234 005000 BNE TST140-10 ;BR TO ERROR HALT ON SEQ ERROR  
3694 011236 052710 CLR R0 ;R0=0  
3695 011242 052700 CLR (R0) ;LOC. 0=0  
3696 011246 000277 BIS #125252,(R0) ;LOC. 0=125125  
3697 011250 022710 SCC R0 ;R0=2  
3698 011254 100403 BIT ;SET ALL COND. CODE BITS  
3699 011256 102402 BMI DNM4A ;TRY DOPNM W/ MODE 4  
3700 011260 103001 BVS DNM4A ;BR TO ERROR IF N-BIT SET  
3701 011262 001004 BCC DNM4A ;BR TO ERROR IF V-BIT SET  
3702 ; BR TO ERROR IF C-BIT CHAR
```

```

3702 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3703 ; CONDITIONAL BRANCH INST. AND <====
3704 ; REPLACE THE MOVE INSTRUCTION <====
3705 ; WHICH FOLLOWS W/ 763 <====
3706 011264 012742 000261 DNM4A: MOV #261,-(R2) ;MOVE TO MAILBOX # ***** 261 *****
3707 011270 005242 ;SET MSGTYP TO FATAL ERROR
3708 011272 000000 ;COND. CODES INCORRECT
3709 011274 005700 ;CHECK DEST. REGISTER
3710 011276 001404 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3711 ; CONDITIONAL BRANCH INST. AND <====
3712 ; REPLACE THE MOVE INSTRUCTION <====
3713 ; WHICH FOLLOWS W/ 755 <====
3714 011300 012742 000262 DNM4B: MOV #262,-(R2) ;MOVE TO MAILBOX # ***** 262 *****
3715 011304 005242 ;SET MSGTYP TO FATAL ERROR
3716 011306 000000 ;COND. CODES INCORRECT
3717 011310 022737 125252 000000 DNM4C: CMP #125252,@#0 ;DEST. REGISTER NOT DECREMENTED BY 2
3718 011312 001404 BEQ TST140 ;CHECK DEST. DATA
3719 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3720 ; CONDITIONAL BRANCH INST. AND <====
3721 ; REPLACE THE MOVE INSTRUCTION <====
3722 ; WHICH FOLLOWS W/ 745 <====
3723 011320 012742 000263 DNM4D: MOV #263,-(R2) ;MOVE TO MAILBOX # ***** 263 *****
3724 011324 005242 ;SET MSGTYP TO FATAL ERROR
3725 011326 000000 HALT ;DEST. DATA MODIFIED
3726 ; OR SEQUENCE ERROR
3727
3728 ;*****
3729 ;TEST 140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
3730 ;*****
3731 TST140: INC (R2) ;UPDATE TEST NUMBER
3732 CMP #140,(R2) ;SEQUENCE ERROR?
3733 BNE TST141-10 ;BR TO ERROR HALT ON SEQ ERROR
3734 CLR R0 ;R0=0
3735 CLR (R0) ;LOC= 0=0
3736 BIS #52652,(R0) ;LOC= 0=52652
3737 #2,R0 ;R0=2
3738 ;COND. CODES=0
3739 ;TRY DOPNM INST W/MODE 4 ODD BYTE
3740 ;BR TO ERROR IF V BIT SET
3741 ;BR TO ERROR IF Z BIT SET
3742 ;BR TO ERROR IF C BIT SET
3743 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3744 ; CONDITIONAL BRANCH INST. AND <====
3745 ; REPLACE THE MOVE INSTRUCTION <====
3746 ; WHICH FOLLOWS W/ 763 <====
3747 011372 012742 000264 DNM4A: MOV #264,-(R2) ;MOVE TO MAILBOX # ***** 264 *****
3748 011374 005242 ;SET MSGTYP TO FATAL ERROR
3749 011400 000000 ;COND. CODES INCORRECT
3750 011402 022700 000001 DNM4B: CMP #1,R0 ;CHECK DEST. REGISTER
3751 011406 001404 BEQ DNM4C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3752 ; CONDITIONAL BRANCH INST. AND <====
3753 ; REPLACE THE MOVE INSTRUCTION <====
3754 ; WHICH FOLLOWS W/ 763 <====
3755
3756
3757
    
```

```

3758 ; REPLACE THE MOVE INSTRUCTION <====
3759 ; WHICH FOLLOWS W/ 754 <====
3760 011410 012742 000265 DNM4C: MOV #265,-(R2) ;MOVE TO MAILBOX # ***** 265 *****
3761 011412 005242 ;SET MSGTYP TO FATAL ERROR
3762 011416 000000 ;COND. CODES INCORRECT
3763 011420 132740 000201 DNM4D: BITB #201,-(R0) ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
3764 011424 001401 DNM4E: CMP DNM4D ;BR TO ERROR IF Z-BIT SET
3765 011426 100404 BEQ DNM4E
3766 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3767 ; CONDITIONAL BRANCH INST. AND <====
3768 ; REPLACE THE MOVE INSTRUCTION <====
3769 ; WHICH FOLLOWS W/ 744 <====
3770 011430 012742 000266 DNM4D: MOV #266,-(R2) ;MOVE TO MAILBOX # ***** 266 *****
3771 011434 005242 ;SET MSGTYP TO FATAL ERROR
3772 011436 000000 ;COND. CODES INCORRECT
3773 011440 005700 DNM4E: TST R0 ;CHECK DEST. REGISTER
3774 011442 001404 BEQ DNM4F
3775 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3776 ; CONDITIONAL BRANCH INST. AND <====
3777 ; REPLACE THE MOVE INSTRUCTION <====
3778 ; WHICH FOLLOWS W/ 736 <====
3779 011444 012742 000267 DNM4F: MOV #267,-(R2) ;MOVE TO MAILBOX # ***** 267 *****
3780 011450 005242 ;SET MSGTYP TO FATAL ERROR
3781 011452 000000 ;COND. REG. NOT DECREMENTED BY 1
3782 011454 022710 052652 DNM4G: CMP #52652,(R0) ;CHECK DESTINATION DATA
3783 011460 001404 BEQ TST141
3784 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3785 ; CONDITIONAL BRANCH INST. AND <====
3786 ; REPLACE THE MOVE INSTRUCTION <====
3787 ; WHICH FOLLOWS W/ 727 <====
3788 011462 012742 000270 DNM4H: MOV #270,-(R2) ;MOVE TO MAILBOX # ***** 270 *****
3789 011466 005242 ;SET MSGTYP TO FATAL ERROR
3790 011470 000000 HALT ;DEST. DATA MODIFIED
3791 ; OR SEQUENCE ERROR
3792
3793 ;*****
3794 ;TEST 141 TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
3795 ;*****
3796 TST141: INC (R2) ;UPDATE TEST NUMBER
3797 CMP #141,(R2) ;SEQUENCE ERROR?
3798 BNE TST142-10 ;BR TO ERROR HALT ON SEQ ERROR
3799 CLR R0 ;R0=0
3800 CLR (R0) ;LOC= 0=0
3801 BIS #100000,(R0) ;LOC= 0=100000
3802 #402,R0 ;R0=2
3803 ;SET ALL COND. CODE BITS
3804 ;TRY DOPNM W/MODE 5
3805 ;BR TO ERROR IF V-BIT SET
3806 ;BR TO ERROR IF C-BIT CLEAR
3807 ;BR TO ERROR IF Z-BIT SET
3808 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3809 ; CONDITIONAL BRANCH INST. AND <====
3810 ; REPLACE THE MOVE INSTRUCTION <====
3811 ; WHICH FOLLOWS W/ 763 <====
3812
3813
3814
    
```

```
3814 011534  
3815 011534 012742 000271  
3816 011540 005242  
3817 011542 000000  
3818 011544 022700 000400  
3819 011550 001404  
3820  
3821  
3822  
3823  
3824 011552 012742 000272  
3825 011556 005242  
3826 011560 000000 100000 000000  
3827 011562 022737  
3828 011570 001404  
3829  
3830  
3831  
3832  
3833  
3834 011572 012742 000273  
3835 011576 005242  
3836 011600 000000  
3837  
3838  
3839  
3840  
3841 011602 005212  
3842 011604 022712 000142  
3843 011610 001033  
3844 011612 005000  
3845 011614 005010  
3846 011616 052710 000001  
3847 011622 005100  
3848 011624 032760 000001 000001  
3849 011630 021402  
3850 011634 102402  
3851 011636 103001  
3852 011640 100004  
3853  
3854  
3855  
3856  
3857 011642  
3858 011642 012742 000274  
3859 011646 005242  
3860 011650 000000 177777  
3861 011652 001404  
3862 011656 001404  
3863  
3864  
3865  
3866 011660 012742 000275  
3867 011664 005242  
3868 011666 000000  
3869  
DNM5A: MOV #271,-(R2) ;MOVE TO MAILBOX # ***** 271 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
DNM5B: CMP #400,R0 ;CHECK DEST. REGISTER  
BEQ DNM5C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 754 <====  
MOV #272,-(R2) ;MOVE TO MAILBOX # ***** 272 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. REGISTER NOT DECREMENTED BY 2  
BEQ #100000,@#0 ;CHECK DESTINATION DATA  
TST142 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 744 <====  
MOV #273,-(R2) ;MOVE TO MAILBOX # ***** 273 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA INCORRECTLY MODIFIED  
; OR SEQUENCE ERROR  
;*****  
;TEST 142 TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.  
;*****  
TST142: INC (R2) ;UPDATE TEST NUMBER  
CMP #142,(R2) ;SEQUENCE ERROR?  
BNE TST143-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LCC=0  
BIS #1,(R0) ;LOC=0=1  
COM R0 ;RO=-1 C-BIT=1  
BIT #1,(R0) ;TRY DOPNM W/MODE 6  
BEQ DNM6A ;BR TO ERROR IF Z-BIT SET  
BVS DNM6A ;BR TO ERROR IF V-BIT SET  
BCC DNM6A ;BR TO ERROR IF C-BIT CLEAR  
BPL DNM6B ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 744 <====  
MOV #274,-(R2) ;MOVE TO MAILBOX # ***** 274 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
DNM6B: CMP #-1,R0 ;CHECK DEST. REGISTER  
BEQ DNM6C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 744 <====  
MOV #275,-(R2) ;MOVE TO MAILBOX # ***** 275 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. REGISTER MODIFIED
```

```
3870 011670 022737 000001 000000  
3871 011676 001404  
3872  
3873  
3874  
3875  
3876 011700 012742 000276  
3877 011704 005242  
3878 011706 000000  
3879  
3880  
3881  
3882  
3883  
3884 011710 005212  
3885 011712 022712 000143  
3886 011716 001034  
3887 011720 005000  
3888 011722 005010  
3889 011724 052710 125125  
3890 011730 052700 000001  
3891 011734 132770 000125 000403  
3892 011742 102402  
3893 011744 104402  
3894 011746 103401  
3895 011750 001404  
3896  
3897  
3898  
3899  
3900 011752  
3901 011752 012742 000277  
3902 011756 005242  
3903 011760 000000  
3904 011762 022700 000001  
3905 011766 001404  
3906  
3907  
3908  
3909  
3910 011770 012742 000300  
3911 011774 005242  
3912 011776 000000  
3913 012000 022737 125125 000000  
3914 012006 001404  
3915  
3916  
3917  
3918  
3919 012010 012742 000301  
3920 012014 005242  
3921 012016 000000  
3922  
3923  
3924  
3925  
DNM6C: CMP #1,@#0 ;CHECK DEST. DATA  
BEQ TST143 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 745 <====  
MOV #276,-(R2) ;MOVE TO MAILBOX # ***** 276 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA INCORRECT  
; OR SEQUENCE ERROR  
;*****  
;TEST 143 TEST DEST MODE 7 W/DOP NON-MODIFYING INST.  
;*****  
TST143: INC (R2) ;UPDATE TEST NUMBER  
CMP #143,(R2) ;SEQUENCE ERROR?  
BNE TST144-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LCC=0  
BIS #125125,(R0) ;LOC=0=125125  
COM R0 ;RO=1  
BIT #125,@403(R0) ;TRY DOPNM W/MODE 7  
BEQ DNM7A ;BR TO ERROR IF V-BIT SET  
BVS DNM7A ;BR TO ERROR IF N-BIT SET  
BCC DNM7A ;BR TO ERROR IF C-BIT SET  
BPL DNM7B ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====  
MOV #277,-(R2) ;MOVE TO MAILBOX # ***** 277 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
DNM7B: CMP #1,R0 ;CHECK DEST. REGISTER  
BEQ DNM7C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 754 <====  
MOV #300,-(R2) ;MOVE TO MAILBOX # ***** 300 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DESTINATION REGISTER MODIFIED  
DNM7C: CMP #125125,@#0 ;CHECK DEST. DATA  
BEQ TST144 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 744 <====  
MOV #301,-(R2) ;MOVE TO MAILBOX # ***** 301 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA INCORRECT  
; OR SEQUENCE ERROR  
;*****  
;
```

```
3926 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 1 INSTRUCTION.
3927 ; DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED TO LOC. 0
3928 ; USING MOV SRC MODE 0, DEST. MODE 1.
3929 ;
3930 ;*****
3931 ;TEST 144 TEST MOV DESTINATION MODE 1
3932 ;*****
3933 TST144: INC (R2) ;UPDATE TEST NUMBER
3934 012020 005212 000144 ;SEQUENCE ERROR?
3935 012022 022712 ;BR TO ERROR HALT ON SEQ ERROR
3936 012026 001016 ;R0=0
3937 012030 005000 ;LDC=0=0
3938 012032 005010 ;R0=-1
3939 012034 005100 ;R4 POINTS TO LOC. 0
3940 012040 010014 ;TRY MOVE MODE 0,1
3941 012042 102402 ;BR TO ERROR IF V SET
3942 012044 001401 ;BR TO ERROR IF Z SET
3943 012046 100404 ;
3944 ;
3945 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3946 ; CONDITIONAL BRANCH INST. AND <====
3947 ; REPLACE THE MOVE INSTRUCTION <====
3948 ; WHICH FOLLOWS W/ 770 <====
3949 MDM1A: MOV #302, -(R2) ;MOVE TO MAILBOX # ***** 302 *****
3950 012050 012742 000302 ;SET MSGTYP TO FATAL ERROR
3951 012054 005242 ;CONDITION CODE NOT CORRECT
3952 012056 000000 ;
3953 MDM1B: TST R4
3954 012060 005704 ;
3955 012062 001404 ;
3956 ;
3957 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3958 ; CONDITIONAL BRANCH INST. AND <====
3959 ; REPLACE THE MOVE INSTRUCTION <====
3960 ; WHICH FOLLOWS W/ 762 <====
3961 012074 012742 000303 MOV #303, -(R2) ;MOVE TO MAILBOX # ***** 303 *****
3962 012070 005242 ;SET MSGTYP TO FATAL ERROR
3963 012072 000000 ;DESTINATION REGISTER INCORRECTLY ALTERED
3964 ; OR SEQUENCE ERROR
3965 ;*****
3966 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 2 INSTRUCTION.
3967 ; DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED
3968 ; TO LOCATION 0 USING MOV SRC MODE 0, DEST. MODE 1.
3969 ;*****
3970 ;TEST 145 TEST MOV DESTINATION MODE 2
3971 ;*****
3972 TST145: INC (R2) ;UPDATE TEST NUMBER
3973 012074 005212 000145 ;SEQUENCE ERROR?
3974 012076 022712 ;BR TO ERROR HALT ON SEQ ERROR
3975 012102 001025 ;R0=0
3976 012104 005000 ;LDC=0=0
3977 012106 005010 ;R0=-1
3978 012110 005110 ;R0, (R0)+
3979 012112 100402 ;TRY MOVE MODE 0,2
3980 012116 102401 ;BR TO ERROR IF V SET
3981 012120 001404 ;BR TO ERROR IF Z SET
3982 ;
```

```
3982 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3983 ; CONDITIONAL BRANCH INST. AND <====
3984 ; REPLACE THE MOVE INSTRUCTION <====
3985 ; WHICH FOLLOWS W/ 771 <====
3986 MDM2A: MOV #304, -(R2) ;MOVE TO MAILBOX # ***** 304 *****
3987 012122 012742 000304 ;SET MSGTYP TO FATAL ERROR
3988 012126 005242 ;CC'S INCORRECT
3989 012130 000000 ;
3990 MDM2B: DEC R0
3991 012132 005300 ;
3992 012134 005300 ;
3993 012136 001404 ;
3994 ;
3995 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3996 ; CONDITIONAL BRANCH INST. AND <====
3997 ; REPLACE THE MOVE INSTRUCTION <====
3998 ; WHICH FOLLOWS W/ 762 <====
3999 MDM2C: MOV #305, -(R2) ;MOVE TO MAILBOX # ***** 305 *****
4000 012140 012742 000305 ;SET MSGTYP TO FATAL ERROR
4001 012144 005242 ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
4002 012150 005737 ;
4003 012154 001404 ;
4004 ;
4005 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4006 ; CONDITIONAL BRANCH INST. AND <====
4007 ; REPLACE THE MOVE INSTRUCTION <====
4008 ; WHICH FOLLOWS W/ 753 <====
4009 012156 012742 000306 MOV #306, -(R2) ;MOVE TO MAILBOX # ***** 306 *****
4010 012162 005242 ;SET MSGTYP TO FATAL ERROR
4011 012164 000000 ;DESTINATION DATA INCORRECT
4012 ; OR SEQUENCE ERROR
4013 ;*****
4014 ; THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
4015 ; INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
4016 ;*****
4017 ;TEST 146 TEST MOV-BYTE DESTINATION MODE 2
4018 ;*****
4019 TST146: INC (R2) ;UPDATE TEST NUMBER
4020 012166 005212 000146 ;SEQUENCE ERROR?
4021 012170 022712 ;BR TO ERROR HALT ON SEQ ERROR
4022 012174 001046 ;R0=0
4023 012176 005000 ;LDC=0=0
4024 012200 005010 ;TRY DESTINATION MODE 2 W/EVEN BYTE
4025 012202 112720 ;BR TO ERROR IF V SET
4026 012206 102402 ;BR TO ERROR IF Z SET
4027 012210 001401 ;
4028 012212 100004 ;
4029 ;
4030 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4031 ; CONDITIONAL BRANCH INST. AND <====
4032 ; REPLACE THE MOVE INSTRUCTION <====
4033 ; WHICH FOLLOWS W/ 771 <====
4034 MBDM2A: MOV #307, -(R2) ;MOVE TO MAILBOX # ***** 307 *****
4035 012214 012742 000307 ;SET MSGTYP TO FATAL ERROR
4036 012220 005242 ;CC'S INCORRECT
4037 012222 000000 ;
4038 MBDM2B: CMP #1, R0
```



```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 92
CFKAAC.P11 18-OCT-78 11:01 T146 TEST MOV-BYTE DESTINATION MODE 2 SEQ 0104

4038 012230 001404 BEQ MBDM2C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4039 ; CONDITIONAL BRANCH INST. AND <====
4040 ; REPLACE THE MOVE INSTRUCTION <====
4041 ; WHICH FOLLOWS W/ 160 <====
4042 012232 012742 000310 MOV #310,-(R2) ;MOVE TO MAILBOX # *****/ 310 *****
4043 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4044 012236 005242 000310 MBDM2C: MOV #252,(R0)+ ;REGISTER NOT INCREMENTED BY ONE
4045 012240 000000 HALT MBDM2D ;TRY DESTINATION MODE 2 W/ODD BYTE
4046 012242 112720 000252 MBDM2C: MOV #252,(R0)+
4047 012246 102402 BVS MBDM2D
4048 012250 001401 BEQ MBDM2E
4049 012252 100404 BMI MBDM2E
4050 ;
4051 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4052 ; CONDITIONAL BRANCH INST. AND <====
4053 ; REPLACE THE MOVE INSTRUCTION <====
4054 ; WHICH FOLLOWS W/ 751 <====
4055 012254 012742 000311 MBDM2D: MOV #311,-(R2) ;MOVE TO MAILBOX # ***** 311 *****
4056 012260 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4057 012262 000000 HALT MBDM2E: #2,R0 ;CC'S NOT SET CORRECT
4058 012264 022700 000002 MBDM2E: CMP #2,R0
4059 012270 001404 BEQ MBDM2F
4060 ;
4061 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4062 ; CONDITIONAL BRANCH INST. AND <====
4063 ; REPLACE THE MOVE INSTRUCTION <====
4064 ; WHICH FOLLOWS W/ 742 <====
4065 012272 012742 000312 MOV #312,-(R2) ;MOVE TO MAILBOX # ***** 312 *****
4066 012300 000000 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4067 012302 022737 125125 000000 MBDM2F: CMP #125125,@#0 ;REGISTER NOT INCREMENTED BY ONE
4068 012310 001404 BEQ TST147 ;CHECK DATA
4069 ;
4070 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4071 ; CONDITIONAL BRANCH INST. AND <====
4072 ; REPLACE THE MOVE INSTRUCTION <====
4073 ; WHICH FOLLOWS W/ 732 <====
4074 012312 012742 000313 MOV #313,-(R2) ;MOVE TO MAILBOX # ***** 313 *****
4075 012316 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4076 012320 000000 HALT ;DESTINATION DATA INCORRECT
4077 ; OR SEQUENCE ERROR
4078 ;
4079 ; *****
4080 ; THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
4081 ; AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR. ALSO MOV#B
4082 ; INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
4083 ; *****
4084 ; TEST 147 TEST MOV(B) DESTINATION MODE 3 *****
4085 ; *****
4086 012322 005212 TST147: INC (R2) ;UPDATE TEST NUMBER
4087 012324 022742 000147 CMP #147,(R2) ;SEQUENCE ERROR?
4088 012330 001057 000400 BNE #TST150-10 ;BR TO ERROR HALT ON SEQ ERROR
4089 012332 012700 000400 MOV #400,R0 ;R0=400
4090 012336 005010 CLR (R0) ;LOC. 400 POINTS TO LOC. 0
4091 012340 005037 000000 CLR @#0 ;LOC. 0=0
4092 012344 012730 125252 MOV #125252,@(R0)+ ;TRY MOV DESTINATION MODE 2
4093 012350 102402 MDM3A BVS MDM3A ;BR TO ERROR IF V SET

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 93
CFKAAC.P11 18-OCT-78 11:01 T147 TEST MOV(B) DESTINATION MODE 3 SEQ 0105

4094 012352 001401 BEQ MDM3A ;BR TO ERROR IF Z SET
4095 012354 100404 BMI MDM3B
4096 ;
4097 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4098 ; CONDITIONAL BRANCH INST. AND <====
4099 ; REPLACE THE MOVE INSTRUCTION <====
4100 ; WHICH FOLLOWS W/ 766 <====
4101 012356 012742 000314 MDM3A: MOV #314,-(R2) ;MOVE TO MAILBOX # ***** 314 *****
4102 012362 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4103 012364 000000 HALT MDM3B: #402,R0 ;CC'S INCORRECT
4104 012366 022700 000402 MDM3B: CMP #402,R0 ;CHECK DEST. MODE REGISTER
4105 012372 001404 BEQ MDM3C
4106 ;
4107 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4108 ; CONDITIONAL BRANCH INST. AND <====
4109 ; REPLACE THE MOVE INSTRUCTION <====
4110 ; WHICH FOLLOWS W/ 757 <====
4111 012374 012742 000315 MOV #315,-(R2) ;MOVE TO MAILBOX # *****/ 315 *****
4112 012400 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4113 012402 000000 HALT MDM3C: #125252,@#0 ;REGISTER NOT INCREMENTED BY 2
4114 012412 001404 CMP #125252,@#0 ;CHECK DESTINATION DATA
4115 BEQ MDM3D
4116 ;
4117 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4118 ; CONDITIONAL BRANCH INST. AND <====
4119 ; REPLACE THE MOVE INSTRUCTION <====
4120 ; WHICH FOLLOWS W/ 747 <====
4121 012414 012742 000316 MOV #316,-(R2) ;MOVE TO MAILBOX # ***** 316 *****
4122 012420 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4123 012424 000000 HALT MDM3D: #125,@#0 ;DESTINATION DATA INCORRECT
4124 012432 112737 000125 000000 MDM3D: MOV #125,@#0 ;TRY MOV#B DESTINATION MODE Z EVEN BYTE
4125 012434 022737 125125 000000 CMP #125125,@#0 ;CHECK DATA
4126 012440 001404 BEQ MDM3E
4127 ;
4128 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4129 ; CONDITIONAL BRANCH INST. AND <====
4130 ; REPLACE THE MOVE INSTRUCTION <====
4131 ; WHICH FOLLOWS W/ 734 <====
4132 012442 012742 000317 MOV #317,-(R2) ;MOVE TO MAILBOX # ***** 317 *****
4133 012446 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4134 012450 000000 HALT MDM3E: #525,@#1 ;DESTINATION DATA INCORRECT
4135 012452 112737 000525 000001 CMP #525,@#1 ;TRY MOV#B DESTINATION MODE 2 ODD BYTE
4136 012460 022737 052525 000000 MDM3E: CMP #525,@#0 ;CHECK DATA
4137 012466 001404 BEQ TST150
4138 ;
4139 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4140 ; CONDITIONAL BRANCH INST. AND <====
4141 ; REPLACE THE MOVE INSTRUCTION <====
4142 ; WHICH FOLLOWS W/ 721 <====
4143 012470 012742 000320 MOV #320,-(R2) ;MOVE TO MAILBOX # ***** 320 *****
4144 012474 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4145 012476 000000 HALT ;
4146 ; *****
4147 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4148 ; SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4149 ; R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4150 ; CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.

```

```

4150 ;*****
4151 ;TEST 150 TEST MOV DESTINATION MODE 4
4152 ;*****
4153 012500 005212 000150 TST150: INC (R2) ;UPDATE TEST NUMBER
4154 012502 022712 000150 CMP #150,(R2) ;SEQUENCE ERROR?
4155 012506 001026 000150 BNE TST151-10 ;BR TO ERROR HALT ON SEQ ERROR
4156 012510 005000 000150 CLR R0 ;R0=0
4157 012514 005010 000002 CLR (R0) ;LQC 0=0
4158 012520 012704 012345 MOV #12345,-(R4) ;TRY MOV DEST. MODE 4
4159 012524 102402 012345 BVS MDM4A ;BR TO ERROR IF V-BIT SET
4160 012526 001401 012345 BEQ MDM4A ;BR TO ERROR IF Z-BIT SET
4161 012530 100004 012345 BPL MDM4B
4162 ;
4163 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4164 ; CONDITIONAL BRANCH INST. AND <====
4165 ; REPLACE THE MOVE INSTRUCTION <====
4166 ; WHICH FOLLOWS W/ 767 <====
4167 012532 012742 000321 MDM4A: MOV #321,-(R2) ;MOVE TO MAILBOX # ***** 321 *****
4168 012536 005242 000321 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4169 012540 000000 000321 HALT ;COND. CODES NOT CORRECT
4170 012544 005704 000321 MDM4B: TST R4 ;CHECK DECREMENTING OF MODE 4 REG.
4171 012544 001404 000321 BEQ MDM4C
4172 ;
4173 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4174 ; CONDITIONAL BRANCH INST. AND <====
4175 ; REPLACE THE MOVE INSTRUCTION <====
4176 ; WHICH FOLLOWS W/ 761 <====
4177 012546 012742 000322 MOV #322,-(R2) ;MOVE TO MAILBOX # ***** 322 *****
4178 012552 005242 000322 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4179 012554 000000 012345 HALT ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
4180 012556 022710 012345 MDM4C: CMP #12345,(R0) ;CHECK DESTINATION DATA
4181 012562 001404 012345 BEQ TST151
4182 ;
4183 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4184 ; CONDITIONAL BRANCH INST. AND <====
4185 ; REPLACE THE MOVE INSTRUCTION <====
4186 ; WHICH FOLLOWS W/ 752 <====
4187 012564 012742 000323 MOV #323,-(R2) ;MOVE TO MAILBOX # ***** 323 *****
4188 012570 005242 000323 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4189 012572 000000 000323 HALT ;DESTINATION DATA INCORRECT
4190 ; OR SEQUENCE ERROR
4191 ;
4192 ;*****
4193 ; THIS TEST VERIFIES THE MOV(B) DESTINATION MODE 4 INSTRUCTION
4194 ; ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
4195 ; USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
4196 ; ADDRESSING REGISTER AND CMP AND CONDITIONAL BRANCH
4197 ; INSTRUCTIONS ARE USED TO VERIFY THE DATA.
4198 ;*****
4199 ;*****
4200 ;TEST 151 TEST MOV(B) DESTINATION MODE 4
4201 ;*****
4202 012574 005212 000151 TST151: INC (R2) ;UPDATE TEST NUMBER
4203 012576 022712 000151 CMP #151,(R2) ;SEQUENCE ERROR?
4204 012580 001046 000151 BNE TST152-10 ;BR TO ERROR HALT ON SEQ ERROR
4205 012604 005004 000151 CLR R4 ;R4=0
    
```

```

4206 012606 005014 000002 CLR (R4) ;LQC 0=0
4207 012610 012700 000002 MOV #25R0 ;R0 = 2
4208 012620 020027 000001 MOV #125125,-(R0) ;TRY MOV DEST. MODE 4--ODD BYTE
4209 012624 001404 000001 BEQ MDM4A ;CHECK THAT DEST. REG. WAS DECREMENTED
4210 ;
4211 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4212 ; CONDITIONAL BRANCH INST. AND <====
4213 ; REPLACE THE MOVE INSTRUCTION <====
4214 ; WHICH FOLLOWS W/ 761 <====
4215 012626 012742 000324 MOV #324,-(R2) ;MOVE TO MAILBOX # ***** 324 *****
4216 012632 005242 000324 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4217 012634 000000 052400 MBDM4A: HALT ;DESTINATION REG. NOT DECREMENTED BY 1
4218 012636 021427 052400 MBDM4A: CMP (R4),#52400 ;CHECK DEST. DATA
4219 012642 001404 052400 BEQ MBDM4B
4220 ;
4221 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4222 ; CONDITIONAL BRANCH INST. AND <====
4223 ; REPLACE THE MOVE INSTRUCTION <====
4224 ; WHICH FOLLOWS W/ 760 <====
4225 012644 012742 000325 MOV #325,-(R2) ;MOVE TO MAILBOX # ***** 325 *****
4226 012650 005242 000325 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4227 012654 000000 125125 MBDM4B: HALT ;DEST. DATA NOT CORRECT
4228 012660 102402 125125 MBDM4B: MOV #125125,-(R0) ;TRY MOV(B) DEST. MODE 4--EVEN BYTE
4229 012662 001401 125125 BVS MBDM4C ;BR. TO ERROR IF V-BIT SET
4230 012664 100004 125125 BEQ MBDM4C ;BR TO ERROR IF Z-BIT SET
4231 BPL MBDM4D
4232 ;
4233 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4234 ; CONDITIONAL BRANCH INST. AND <====
4235 ; REPLACE THE MOVE INSTRUCTION <====
4236 ; WHICH FOLLOWS W/ 747 <====
4237 012666 012742 000326 MBDM4C: MOV #326,-(R2) ;MOVE TO MAILBOX # ***** 326 *****
4238 012672 005242 000326 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4239 012674 000000 000326 MBDM4D: TST R0 ;COND. CODES INCORRECT
4240 012676 005700 000326 BEQ MBDM4E ;CHECK MODE 4 DEST. REGISTER
4241 ;
4242 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4243 ; CONDITIONAL BRANCH INST. AND <====
4244 ; REPLACE THE MOVE INSTRUCTION <====
4245 ; WHICH FOLLOWS W/ 741 <====
4246 012702 012742 000327 MOV #327,-(R2) ;MOVE TO MAILBOX # ***** 327 *****
4247 012706 005242 000327 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4248 012710 000000 052525 MBDM4E: HALT ;DESTINATION REG NOT DECREMENTED BY 1
4249 012712 021427 052525 MBDM4E: CMP (R4),#52525 ;CHECK DEST. DATA
4250 012716 001404 052525 BEQ TST152
4251 ;
4252 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4253 ; CONDITIONAL BRANCH INST. AND <====
4254 ; REPLACE THE MOVE INSTRUCTION <====
4255 ; WHICH FOLLOWS W/ 732 <====
4256 012720 012742 000330 MOV #330,-(R2) ;MOVE TO MAILBOX # ***** 330 *****
4257 012724 005242 000330 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4258 012726 000000 000330 HALT ;DESTINATION DATA INCORRECT
4259 ; OR SEQUENCE ERROR
4260 ;
4261 ;*****
4262 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOV(B)
    
```

```
4262 ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A  
4263 ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO  
4264 ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR  
4265 ;THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY  
4266 ;PROPER ADDRESSING AND DATA.  
4267 ;*****  
4268 ;TEST 152 TEST MOV DESTINATION MODE 5  
4269 ;*****  
4270 TST152: INC (R2) ;UPDATE TEST NUMBER  
4271 CMP #152,(R2) ;SEQUENCE ERROR?  
4272 BNE TST153-10 ;BR TO ERROR HALT ON SEQ ERROR  
4273 CLR R4 ;R4=0  
4274 CLR (R4) ;LOC 0 = 0  
4275 MOV #400,R0 ;RO=400  
4276 MOV #376,@-(R0) ;TRY MOV DEST. MODE 5  
4277 BVS MDM5A ;BR TO ERROR IF V-BIT SET  
4278 BEQ MDM5A ;BR TO ERROR IF Z-BIT SET  
4279 BPL MDM5B  
4280 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4281 ; CONDITIONAL BRANCH INST. AND <====  
4282 ; REPLACE THE MOVE INSTRUCTION <====  
4283 ; WHICH FOLLOWS W/ 767 <====  
4284  
4285 MDM5A: MOV #331,-(R2) ;MOVE TO MAILBOX # ***** 331 *****  
4286 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4287 HALT ;COND. CODES INCORRECT  
4288 MDM5B: CMP #376,R0 ;CHECK MODE 5 REG. WAS DECREMENTED  
4289 BEQ MDM5C  
4290 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4291 ; CONDITIONAL BRANCH INST. AND <====  
4292 ; REPLACE THE MOVE INSTRUCTION <====  
4293 ; WHICH FOLLOWS W/ 760 <====  
4294  
4295 MOV #332,-(R2) ;MOVE TO MAILBOX # ***** 332 *****  
4296 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4297 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2  
4298 MDM5C: CMP #4321,(R4) ;CHECK DEST. DATA  
4299 BEQ MDM5D  
4300 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4301 ; CONDITIONAL BRANCH INST. AND <====  
4302 ; REPLACE THE MOVE INSTRUCTION <====  
4303 ; WHICH FOLLOWS W/ 751 <====  
4304  
4305 MOV #333,-(R2) ;MOVE TO MAILBOX # ***** 333 *****  
4306 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4307 HALT ;DEST. DATA INCORRECT  
4308 MDM5D: MOV #406,R0 ;RO=406  
4309 MOV #377,@-(R0) ;TRY MOV DEST. MODE 5 --EVEN BYTE  
4310 CMP #404,R0 ;CHECK MODE 5 REG.  
4311 BEQ MDM5E  
4312 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4313 ; CONDITIONAL BRANCH INST. AND <====  
4314 ; REPLACE THE MOVE INSTRUCTION <====  
4315 ; WHICH FOLLOWS W/ 736 <====  
4316  
4317 MDM5E: MOV #334,-(R2) ;MOVE TO MAILBOX # ***** 334 *****  
4318 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4319 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
```

```
4318 MDM5E: CMP #177721,(R4) ;CHECK DEST. DATA  
4319 BEQ TST153  
4320 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4321 ; CONDITIONAL BRANCH INST. AND <====  
4322 ; REPLACE THE MOVE INSTRUCTION <====  
4323 ; WHICH FOLLOWS W/ 771 <====  
4324  
4325 MOV #335,-(R2) ;MOVE TO MAILBOX # ***** 335 *****  
4326 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4327 HALT ;DEST. DATA INCORRECT  
4328 ; OR SEQUENCE ERROR  
4329 ;*****  
4330 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE  
4331 ; DESTINATION MODE 6 INSTRUCTIONS. R0 IS USED TO SETUP TARGET LOC 0  
4332 ; FOR BOTH TESTS. PATTERNS OF ONES AND ZEROS ARE MOVED INTO R0  
4333 ; BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY  
4334 ; PROPER ADDRESSING AND DATA.  
4335 ;*****  
4336 ;TEST 153 TEST MOV DESTINATION MODE 6  
4337 ;*****  
4338 TST153: INC (R2) ;UPDATE TEST NUMBER  
4339 CMP #153,(R2) ;SEQUENCE ERROR?  
4340 BNE TST154-10 ;BR TO ERROR HALT ON SEQ ERROR  
4341 CLR R0 ;R0=0  
4342 CLR (R0) ;LOC 0=0  
4343 MOV #052525,-1(R0) ;TRY MOV DEST. MODE 6  
4344 INC R0 ;RO=1  
4345 MOV #052525,-1(R0) ;TRY MOV DEST. MODE 6  
4346 BVS MDM6A ;BR TO ERROR IF V-BIT SET  
4347 BEQ MDM6A ;BR TO ERROR IF Z-BIT SET  
4348 BPL MDM6B  
4349 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4350 ; CONDITIONAL BRANCH INST. AND <====  
4351 ; REPLACE THE MOVE INSTRUCTION <====  
4352 ; WHICH FOLLOWS W/ 767 <====  
4353  
4354 MDM6A: MOV #336,-(R2) ;MOVE TO MAILBOX # ***** 336 *****  
4355 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4356 HALT ;COND. CODES INCORRECT  
4357 MDM6B: CMP #1,R0 ;CHECK DEST. REGISTER UNALTERED  
4358 BEQ MDM6C  
4359 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4360 ; CONDITIONAL BRANCH INST. AND <====  
4361 ; REPLACE THE MOVE INSTRUCTION <====  
4362 ; WHICH FOLLOWS W/ 760 <====  
4363  
4364 MOV #337,-(R2) ;MOVE TO MAILBOX # ***** 337 *****  
4365 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4366 HALT ;DEST. REGISTER INCORRECTLY ALTERED  
4367 MDM6C: CMP #52525,@#0 ;CHECK DEST. DATA  
4368 BEQ MDM6D  
4369 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4370 ; CONDITIONAL BRANCH INST. AND <====  
4371 ; REPLACE THE MOVE INSTRUCTION <====  
4372 ; WHICH FOLLOWS W/ 760 <====  
4373  
4374 MOV #340,-(R2) ;MOVE TO MAILBOX # ***** 340 *****
```

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

```
4430 ; ; REPLACE THE MOVE INSTRUCTION <====  
4431 ; ; WHICH FOLLOWS W/ 757 <====  
4432 013314 012742 000344 MOV #344,-(R2) ;MOVE TO MAILBOX # ***** 344 *****  
4433 013320 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4434 013322 000000 HALT -(R2) ;DEST. REGISTER INCORRECTLY ALTERED  
4435 013324 022737 070707 000000 MDM7C: CMP #70707,@#0 ;CHECK DEST. DATA  
4436 013332 001404 BEQ MDM7D ;TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4437 ; ; CONDITIONAL BRANCH INST. AND <====  
4438 ; ; REPLACE THE MOVE INSTRUCTION <====  
4439 ; ; WHICH FOLLOWS W/ 747 <====  
4440 ; ;  
4441 013334 012742 000345 MOV #345,-(R2) ;MOVE TO MAILBOX # ***** 345 *****  
4442 013340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4443 013342 000000 HALT -(R2) ;DEST. DATA INCORRECT  
4444 013344 112770 107070 000001 MDM7D: MOVB #107070,@1(R0) ;TRY MOVW W/DEST MODE 7--ODD BYTE  
4445 013352 022700 000403 CMP #403,R0 ;CHECK MODE 7 DEST. REG.  
4446 013356 001404 BEQ MDM7E ;TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4447 ; ; CONDITIONAL BRANCH INST. AND <====  
4448 ; ; REPLACE THE MOVE INSTRUCTION <====  
4449 ; ; WHICH FOLLOWS W/ 735 <====  
4450 ; ;  
4451 013360 012742 000346 MOV #346,-(R2) ;MOVE TO MAILBOX # ***** 346 *****  
4452 013364 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4453 013366 000000 HALT -(R2) ;DEST. DATA INCORRECT  
4454 013370 022737 034307 000000 MDM7E: CMP #34307,@#0 ;CHECK DEST. DATA  
4455 013376 001404 BEQ TST155 ;TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4456 ; ; CONDITIONAL BRANCH INST. AND <====  
4457 ; ; REPLACE THE MOVE INSTRUCTION <====  
4458 ; ; WHICH FOLLOWS W/ 725 <====  
4459 ; ;  
4460 013400 012742 000347 MOV #347,-(R2) ;MOVE TO MAILBOX # ***** 347 *****  
4461 013404 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4462 013406 000000 HALT -(R2) ;DESTINATION DATA INCORRECT  
4463 ; ; OR SEQUENCE ERROR  
4464 ; ;  
4465 ; ;  
4466 ; ;  
4467 ; ;  
4468 ; ; *****  
4469 ; ; THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.  
4470 ; ; THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A  
4471 ; ; TABLE OF OPERANDS. THE TABLE OF OPERANDS AND THE WORK LOCATION IS  
4472 ; ; STORED FOLLOWING THE TEST CODE. A SERIES OF 5 DOP INSTRUCTIONS UTILIZES  
4473 ; ; THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF  
4474 ; ; VALUES. THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL  
4475 ; ; GO UNDETECTED. WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND  
4476 ; ; ODD ADDRESSES ARE USED IN THE TEST. THE LISTING SHOWS THE  
4477 ; ; EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.  
4478 ; ; *****  
4479 ; ; TEST 155 TEST MODE 4 W/ DOP INSTS *****  
4480 013410 005212 000155 TST155: INC (R2) ;UPDATE TEST NUMBER  
4481 013412 022712 000155 CMP #155,(R2) ;SEQUENCE ERROR?  
4482 013416 001015 BNE DOP4 ;BR TO ERROR HALT ON SEQ ERROR  
4483 013420 012770 MOV #TBL1,R0 ;INITIALIZE R0  
4484 013424 014037 MOV -(R0),@#TBL1 ;TBL1=125252  
4485 013430 064037 ADD -(R0),@#TBL1 ;TBL1=000377
```

```
4486 013434 144037 013472 BICB -(R0),@#TBL1 ;TBL1=000252  
4487 013440 154037 013473 BISB -(R0),@#TBL1+1 ;TBL1=125252  
4488 013442 024037 013474 CMP (R0),@#TBL1 ;CHECK RESULT  
4489 013450 001411  
4490  
4491  
4492  
4493  
4494 013452  
4495 013452 012742 000350 DOP4: MOV #350,-(R2) ;MOVE TO MAILBOX # ***** 350 *****  
4496 013456 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4497 013460 000000 HALT ;RESULT OF MODE 4 INSTS. INCORRECT  
4498  
4499  
4500 013462 125252 ; OR SEQUENCE ERROR  
4501 013464 052652  
4502 013466 053125  
4503 013470 125252  
4504 013472 000000  
4505  
4506  
4507  
4508  
4509  
4510  
4511  
4512  
4513  
4514  
4515  
4516  
4517  
4518 013474 005212  
4519 013476 022712 000156 TST156: INC (R2) ;UPDATE TEST NUMBER  
4520 013502 001015 CMP #156,(R2) ;SEQUENCE ERROR?  
4521 013504 012700 BNE DOP5 ;BR TO ERROR HALT ON SEQ ERROR  
4522 013510 015037 013472 MOV #TBL2+2,R0 ;INITIALIZE R0  
4523 013514 066037 013473 ADD (R0),@#TBL1 ;TBL1=125252  
4524 013520 146037 013472 ADD (R0),@#TBL1 ;TBL1=000377  
4525 013524 156037 013473 BICB (R0),@#TBL1 ;TBL1=000252  
4526 013530 025037 013473 BISB (R0),@#TBL1+1 ;TBL1=125252  
4527 013534 001411 CMP (R0),@#TBL1 ;CHECK RESULT  
4528  
4529  
4530  
4531  
4532 013536  
4533 013536 012742 000351 DOP5: MOV #351,-(R2) ;MOVE TO MAILBOX # ***** 351 *****  
4534 013540 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4535 013544 000000 HALT ;RESULT OF MODE 5 INSTS. INCORRECT  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555 013560 005212  
4556 013562 022712 000157 TST157: INC (R2) ;UPDATE TEST NUMBER  
4557 013566 001022 CMP #157,(R2) ;SEQUENCE ERROR?  
4558 013570 012700 BNE TST160-10 ;BR TO ERROR HALT ON SEQ ERROR  
4559 013574 016037 000002 013472 MOV #TBL1-4,R0 ;INITIALIZE R0  
4560 013602 066037 000000 013472 MOV (R0),@#TBL1 ;TBL1=125252  
4561 013610 146037 177777 013472 ADD (R0),@#TBL1 ;TBL1=000377  
4562 013616 156037 177776 013473 BICB -(R0),@#TBL1 ;TBL1=000252  
4563 013624 026037 177774 013472 BISB -4(R0),@#TBL1+1 ;TBL1=125252  
4564 013632 001404 CMP (R0),@#TBL1 ;CHECK RESULT  
4565  
4566  
4567  
4568  
4569 013634 012742 000352 MOV #352,-(R2) ;MOVE TO MAILBOX # ***** 352 *****  
4570 013640 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4571 013642 000000 HALT ;RESULT OF MODE 6 INSTS. INCORRECT  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586 013644 005212  
4587 013646 022712 000160 TST160: INC (R2) ;UPDATE TEST NUMBER  
4588 013652 001022 CMP #160,(R2) ;SEQUENCE ERROR?  
4589 013654 017400 BNE TST161-10 ;BR TO ERROR HALT ON SEQ ERROR  
4590 013660 017037 000004 013472 MOV #TBL2-4,R0 ;INITIALIZE R0  
4591 013666 067037 000002 013472 MOV (R0),@#TBL1 ;TBL1=125252  
4592 013674 147037 000000 013472 ADD (R0),@#TBL1 ;TBL1=000377  
4593 013702 157037 177776 013473 BICB (R0),@#TBL1 ;TBL1=000252  
4594 013710 027037 177774 013472 BISB (R0),@#TBL1+1 ;TBL1=125252  
4595 013716 001404 CMP (R0),@#TBL1 ;CHECK RESULT  
4596  
4597
```

```
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555 013560 005212  
4556 013562 022712 000157 TST157: INC (R2) ;UPDATE TEST NUMBER  
4557 013566 001022 CMP #157,(R2) ;SEQUENCE ERROR?  
4558 013570 012700 BNE TST160-10 ;BR TO ERROR HALT ON SEQ ERROR  
4559 013574 016037 000002 013472 MOV #TBL1-4,R0 ;INITIALIZE R0  
4560 013602 066037 000000 013472 MOV (R0),@#TBL1 ;TBL1=125252  
4561 013610 146037 177777 013472 ADD (R0),@#TBL1 ;TBL1=000377  
4562 013616 156037 177776 013473 BICB -(R0),@#TBL1 ;TBL1=000252  
4563 013624 026037 177774 013472 BISB -4(R0),@#TBL1+1 ;TBL1=125252  
4564 013632 001404 CMP (R0),@#TBL1 ;CHECK RESULT  
4565  
4566  
4567  
4568  
4569 013634 012742 000352 MOV #352,-(R2) ;MOVE TO MAILBOX # ***** 352 *****  
4570 013640 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4571 013642 000000 HALT ;RESULT OF MODE 6 INSTS. INCORRECT  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586 013644 005212  
4587 013646 022712 000160 TST160: INC (R2) ;UPDATE TEST NUMBER  
4588 013652 001022 CMP #160,(R2) ;SEQUENCE ERROR?  
4589 013654 017400 BNE TST161-10 ;BR TO ERROR HALT ON SEQ ERROR  
4590 013660 017037 000004 013472 MOV #TBL2-4,R0 ;INITIALIZE R0  
4591 013666 067037 000002 013472 MOV (R0),@#TBL1 ;TBL1=125252  
4592 013674 147037 000000 013472 ADD (R0),@#TBL1 ;TBL1=000377  
4593 013702 157037 177776 013473 BICB (R0),@#TBL1 ;TBL1=000252  
4594 013710 027037 177774 013472 BISB (R0),@#TBL1+1 ;TBL1=125252  
4595 013716 001404 CMP (R0),@#TBL1 ;CHECK RESULT  
4596  
4597
```

```

4598                                     ; REPLACE THE MOVE INSTRUCTION <====
4599                                     ; WHICH FOLLOWS W/ 756 <====
4600 013720 012742 000353      MOV    #353,-(R2)      ;MOVE TO MAILBOX # ***** 353 *****
4601 013724 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4602 013726 000000              HALT                    ;RESULT OF MODE 7 INSTS INCORRECT
4603                                     ; OR SEQUENCE ERROR
4604                                     ;*****
4605                                     ;
4606                                     ; THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
4607 ;RO IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED AND
4608 ;AN ROL INSTRUCTION IS EXECUTED WITH MODE 0. THE OPERATION IS CHECKED
4609 ;BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
4610 ;NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
4611                                     ;*****
4612                                     ;
4613 ;TEST 161 TEST ROTATE INSTRUCTIONS OF MODE 0
4614 ;*****
4615 013730 005212      TST161: INC    (R2)          ;UPDATE TEST NUMBER
4616 013732 022712      CMP    #161,(R2)       ;SEQUENCE ERROR?
4617 013736 012025      BNE    #161-10        ;BR TO ERROR HALT ON SEQ ERROR
4618 013740 012700      MOV    #125252,R0      ;INITIALIZE DATA
4619 013744 000261      SEC                    ;SET C-BIT
4620 013746 006100      ROL    R0              ;TRY ROL W/ MODE 0
4621 013750 102004      BVC    ROT0A          ;CC=0011
4622 013752 103003      BCC    ROT0A          ;
4623 013754 022700      CMP    #052525,R0      ;CHECK DATA
4624 013760 001404      BEQ    ROT0B          ;
4625                                     ;
4626                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4627                                     ; CONDITIONAL BRANCH INST. AND <====
4628                                     ; REPLACE THE MOVE INSTRUCTION <====
4629                                     ; WHICH FOLLOWS W/ 767 <====
4630 013762              ROT0A: MOV    #354,-(R2)      ;MOVE TO MAILBOX # ***** 354 *****
4631 013766 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4632 013770 000000              HALT                    ;ROL MODE 0 FAILED
4633 013772 012700      ROT0B: MOV    #125252,R0      ;INITIALIZE DATA
4634 013776 000261      SEC                    ;SET C-BIT
4635 014000 106100      ROLB   R0              ;TRY ROL W/ MODE 0 EVEN BYTE
4636 014002 102004      BVC    ROT0C          ;CC=0011
4637 014004 103003      BCC    ROT0C          ;
4638 014006 022700      CMP    #125125,R0      ;CHECK DATA
4639 014012 001404      BEQ    ROT0B          ;
4640                                     ;
4641                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4642                                     ; CONDITIONAL BRANCH INST. AND <====
4643                                     ; REPLACE THE MOVE INSTRUCTION <====
4644                                     ; WHICH FOLLOWS W/ 752 <====
4645 014014              ROT0C: MOV    #355,-(R2)      ;MOVE TO MAILBOX # ***** 355 *****
4646 014020 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4647 014022 000000              HALT                    ;ROLB MODE 0 FAILED
4648                                     ; OR SEQUENCE ERROR
    
```

```

4649                                     ;*****
4650                                     ;
4651                                     ; THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
4652 ;THE DATA TO BE ROTATED IS IN LOC 0. RO IS USED AS THE
4653 ;ADDRESSING REGISTER. THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
4654 ;THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
4655 ;THE C AND V BITS. THIS PROCEDURE IS THEN REPEATED TWICE MORE
4656 ;TO TEST THE BYTE ROTATES. FIRST ON BYTE 0, THEN ON BYTE 1.
4657                                     ;*****
4658                                     ;
4659 ;TEST 162 TEST ROTATE INSTRUCTIONS W/ MODE 1
4660 ;*****
4661 014024 005212      TST162: INC    (R2)          ;UPDATE TEST NUMBER
4662 014026 022712      CMP    #162,(R2)       ;SEQUENCE ERROR?
4663 014032 001051      BNE    #162-10        ;BR TO ERROR HALT ON SEQ ERROR
4664 014034 005000      CLR    R0              ;POINT TO LOC. 0
4665 014036 002710      MOV    #52525,(R0)      ;INITIALIZE DATA
4666 014042 000241      CLC                    ;CLEAR C-BIT
4667 014044 006110      ROL    (R0)              ;TRY ROL W/ MODE 1
4668 014046 102005      BVC    ROT1A          ;CC=1010
4669 014050 103404      BCS    ROT1A          ;
4670 014052 023727      CMP    #0,#125252      ;CHECK RESULT
4671 014060 001404      BEQ    ROT1B          ;
4672                                     ;
4673                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4674                                     ; CONDITIONAL BRANCH INST. AND <====
4675                                     ; REPLACE THE MOVE INSTRUCTION <====
4676                                     ; WHICH FOLLOWS W/ 765 <====
4677 014062              ROT1A: MOV    #356,-(R2)      ;MOVE TO MAILBOX # ***** 356 *****
4678 014066 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4679 014070 000000              HALT                    ;ROL MODE 1 FAILED
4680 014072 000261      ROT1B: SEC                    ;
4681 014074 012710      MOV    #125252,(R0)      ;INITIALIZE DATA
4682 014100 106110      ROLB   (R0)              ;TRY ROLB W/ MODE 1 EVEN BYTE
4683 014102 102005      BVC    ROT1C          ;CC=1011
4684 014104 103004      BCC    ROT1C          ;
4685 014106 022737      CMP    #125125,#0      ;TEST RESULT
4686 014114 001404      BEQ    ROT1D          ;
4687                                     ;
4688                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4689                                     ; CONDITIONAL BRANCH INST. AND <====
4690                                     ; REPLACE THE MOVE INSTRUCTION <====
4691                                     ; WHICH FOLLOWS W/ 747 <====
4692 014116              ROT1C: MOV    #357,-(R2)      ;MOVE TO MAILBOX # ***** 357 *****
4693 014122 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4694 014124 000000              HALT                    ;ROLB W/ MODE 1 EVEN BYTE FAILED
4695 014126 012710      ROT1D: MOV    #125252,(R0)      ;POINT TO ODD BYTE
4696 014132 005000      CLR    R0              ;
4697 014134 005200      INC    R0              ;POINT TO ODD BYTE
4698 014136 000261      SEC                    ;SET C-BIT
4699 014140 106110      ROLB   (R0)              ;TRY ROLB W/ MODE 1 ODD BYTE
4700 014142 102005      BVC    ROT1E          ;CC=0011
4701 014144 103004      BCC    ROT1E          ;
4702 014146 022737      CMP    #052652,#0      ;CHECK DATA
4703 014154 001404      BEQ    TST163          ;
    
```

```

4705 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4706 ; CONDITIONAL BRANCH INST. AND <====
4707 ; REPLACE THE MOVE INSTRUCTION <====
4708 ; WHICH FOLLOWS W/ 727 <====
4709 ROT1E: MOV #360,-(R2) ;MOVE TO MAILBOX # ***** 360 *****
4710 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4711 HALT ;ROLB W/ MODE 1 ODD BYTE FAILED
4712 ; OR SEQUENCE ERROR
4713
4714 ;*****
4715 ;
4716 ; THIS TEST VERIFIES MODE 2 ROTATE INSTRUCTIONS.
4717 ; THE SAME PROCEDURE AS IN THE OTHER ROTATE TESTS ARE USED. RO
4718 ; IS USED AS THE ADDRESSING REGISTER AND IS CHECKED FOR PROPER
4719 ; INCREMENTING. BYTE INSTRUCTIONS ARE ALSO CHECKED.
4720 ;*****
4721 ;*****
4722 ;TEST 163 TEST ROTATE INSTRUCTIONS W/ MODE 2
4723 ;*****
4724 ;*****
4725 TST163: INC (R2) ;UPDATE TEST NUMBER
4726 CMP #163,(R2) ;SEQUENCE ERROR?
4727 BNE TST164-10 ;BR TO ERROR HALT ON SEQ ERROR
4728 CLR RO ;POINT TO LOC 0
4729 MOV #173737,(RO) ;INITIALIZE DATA
4730 CLC ;CLEAR C-BIT
4731 ROLB (RO)+ ;TRY ROLB W/ MODE 2
4732 BCC ROT2A ;CHECK C-BIT
4733 CMP #167676,@#0 ;CHECK DATA
4734 BNE ROT2A ;BRANCH IF RESULT INCORRECT
4735 DEC RO ;TEST RO
4736 DEC RO
4737 BEQ ROT2B
4738
4739 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4740 ; CONDITIONAL BRANCH INST. AND <====
4741 ; REPLACE THE MOVE INSTRUCTION <====
4742 ; WHICH FOLLOWS W/ 763 <====
4743 ROT2A: MOV #361,-(R2) ;MOVE TO MAILBOX # ***** 361 *****
4744 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4745 HALT ;ROL W/ MODE 2 FAILED
4746 ROT2B: CLR RO ;POINT TO LOC 0
4747 MOV #4040,(RO) ;INITIALIZE DATA
4748 CLC ;CLEAR C-BIT
4749 ROLB (RO)+ ;TRY ROLB W/ MODE 2 EVEN BYTE
4750 BCC ROT2C ;CHECK C-BIT
4751 CMP #4100,@#0 ;CHECK DATA
4752 BNE ROT2C ;BRANCH IF DATA INCORRECT
4753 DEC RO ;CHECK RO
4754 BEQ ROT2D
4755
4756 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4757 ; CONDITIONAL BRANCH INST. AND <====
4758 ; REPLACE THE MOVE INSTRUCTION <====
4759 ; WHICH FOLLOWS W/ 743 <====
4760 ROT2C: MOV #362,-(R2) ;MOVE TO MAILBOX # ***** 362 *****
    
```

```

4761 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4762 HALT ;ROLB W/ MODE 2 EVEN BYTE FAILED
4763 ROT2D: CLR RO ;POINT TO LOC 0
4764 MOV #4040,(RO) ;INITIALIZE DATA
4765 INC RO ;POINT TO ODD BYTE OF DATA
4766 SEC ;SET C-BIT
4767 ROLB (RO)+ ;TRY ROLB W/ MODE 2 ODD BYTE
4768 BCS ROT2E ;CHECK C-BIT
4769 CMP #10440,@#0 ;CHECK DATA
4770 BNE ROT2E ;BRANCH IF DATA INCORRECT
4771 DEC RO ;CHECK RO
4772 DEC RO
4773 BEQ TST164
4774
4775 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4776 ; CONDITIONAL BRANCH INST. AND <====
4777 ; REPLACE THE MOVE INSTRUCTION <====
4778 ; WHICH FOLLOWS W/ 721 <====
4779 ROT2E: MOV #363,-(R2) ;MOVE TO MAILBOX # ***** 363 *****
4780 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4781 HALT ;ROLB W/ MODE 2 ODD BYTE FAILED
4782 ; OR SEQUENCE ERROR
    
```

```

*****
THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
TESTS. THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
MODE 37. BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
*****
TEST 164 TEST ROTATE INSTRUCTIONS W/ MODE 3
*****
TST164: INC (R2) ;UPDATE TEST NUMBER
        CMP #164,(R2) ;SEQUENCE ERROR?
        BNE TST165-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #52525,@#0 ;INITIALIZE DATA IN LOC 0
        SEC ;SET C-BIT
        ROL #0 ;TRY ROL W/ MODE 3
        BCS ROT3A ;CHECK C-BIT
        CMP #125253,@#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 #364,-(R2) ;MOVE TO MAILBOX # ***** 364 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 FAILED
ROT3B: MOV #125252,@#0 ;INITIALIZE DATA
        CLC ;CLEAR C-BIT
        ROLB #0 ;TRY ROL W/ MODE 3 EVEN BYTE
        BCC ROT3C ;CHECK C-BIT
        CMP #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 #365,-(R2) ;MOVE TO MAILBOX # ***** 365 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 EVEN BYTE FAILED
ROT3D: MOV #125252,@#0 ;INITIALIZE DATA IN LOC. 0
        SEC ;SET C-BIT
        ROLB #1 ;TRY ROL W/ MODE 3 ODD BYTE
        BCC ROT3E ;CHECK C-BIT
        CMP #052652,@#0 ;CHECK DATA
        BEQ TST165
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 727 <====
ROT3E: MOV #366,-(R2) ;MOVE TO MAILBOX # ***** 366 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 ODD BYTE FAILED

```

```

; OR SEQUENCE ERROR
*****
THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS. THE DATA IS
STORED IN LOC. 0. 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 165 TEST MODE 4 W/ ROTATE INSTRUCTIONS
*****
TST165: INC (R2) ;UPDATE TEST NUMBER
        CMP #165,(R2) ;SEQUENCE ERROR?
        BNE TST166-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #070707,@#0 ;INITIALIZE DATA IN LOC. 0
        MOV #2,RO ;INITIALIZE RO AS POINTER
        SEC ;SET C-BIT
        ROL -(RO) ;TRY ROL W/ MODE 4
        BCS ROT4 ;CHECK C-BIT
        CMP #161617,@#0 ;CHECK DATA
        BNE ROT4 ;BRANCH IF DATA INCORRECT
        TST RO ;CHECK MODE 4 REGISTER
        BEQ TST166
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
ROT4: MOV #367,-(R2) ;MOVE TO MAILBOX # ***** 367 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL MODE 4 FAILED
; OR SEQUENCE ERROR
*****
THIS TEST VERIFIES MODE 5 ROTATE INSTRUCTIONS.
THE DATA IS STORED IN A WORK LOCATION (ROTX) AT THE END OF THE
TEST CODE. LOC. 0 IS LOADED WITH THE ADDRESS OF THE DATA (ROTX).
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 166 TEST MODE 5 W/ ROTATE INSTRUCTIONS
*****
TST166: INC (R2) ;UPDATE TEST NUMBER
        CMP #166,(R2) ;SEQUENCE ERROR?
        BNE ROT5 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #ROTX,@#0 ;MOVE POINTER TO LOC. 0
        MOV #0 ;SET MODE 5 REF TO LOC. 0
        MOV #107070,ROTX ;INITIALIZE DATA
        CLC ;CLEAR C-BIT
        ROL @-(RO) ;TRY ROL W/ MODE 5

```



```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 108
CFKAAC.P11 18-OCT-78 11:01 T166 TEST MODE 5 W/ ROTATE INSTRUCTIONS SEQ 0120

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

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 109
CFKAAC.P11 18-OCT-78 11:01 T167 TEST MODE 6 W/ ROTATE INSTRUCTIONS SEQ 0121

4939 ; *****
4940 ; *****
4941 ; *****
4942 ; THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
4943 ; THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST). THE ROL INSTRUCTION
4944 ; ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
4945 ; (ROTXAD) FOLLOWING THE TEST CODE.
4946 ; *****
4947 ; *****
4948 ; *****
4949 ; *****
4950 014714 005212 TST170: INC (R2) ;UPDATE TEST NUMBER
4951 014716 022712 CMP #170,(R2) ;SEQUENCE ERROR?
4952 014722 001013 BNE ROT7 ;BR TO ERROR HALT ON SEQ ERROR
4953 014724 012737 MOV #52525,@#ROTX ;INITIALIZE DATA
4954 014732 012737 MOV #ROTX,@#ROTXAD ;INITIALIZE ADDRESS POINTER
4955 014740 000241 CLC ;CLEAR C-BIT
4956 014742 006177 ROL ROTXAD ;TRY ROL W/ MODE 7
4957 014746 103404 BCS ROT7 ;CHECK C-BIT
4958 014750 023757 CMP @#ROTX,#125252 ;CHECK DATA
4959 014756 001405 BEQ TST171
4960 ;
4961 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
4962 ; CONDITIONAL BRANCH INST. AND <===
4963 ; REPLACE THE MOVE INSTRUCTION <===
4964 ; WHICH FOLLOWS W/ 762 <===
4965 014760 012742 000372 ROT7: MOV #372,-(R2) ;MOVE TO MAILBOX # ***** 372 *****
4966 014764 005242 ;SET MSGTYP TO FATAL ERROR
4967 014766 000000 ;ROL W/ MODE 7 FAILED
4968 ; OR SEQUENCE ERROR
4969 014770 000000 ROTXAD: 0
4970 ;
4971 ; *****
4972 ; *****
4973 ; *****
4974 ; *****
4975 ; THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION. RO IS SET TO
4976 ; 177400. A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH
4977 ; IS USED TO CHECK THE SIGN OF THE RESULT. ALSO, A COMPARISON
4978 ; IS MADE TO CHECK THE DATA RESULTS.
4979 ; *****
4980 ; *****
4981 ; *****
4982 014772 005212 TST171: INC (R2) ;UPDATE TEST NUMBER
4983 014774 022712 CMP #171,(R2) ;SEQUENCE ERROR?
4984 015000 001013 BNE TST172-10 ;BR TO ERROR HALT ON SEQ ERROR
4985 015002 012700 MOV #177400,RO ;MOVE TEST PATTERN TO RO
4986 015006 000300 SWAB #0 ;TRY SWAB MODE 0
4987 015010 100404 BMI SBO
4988 ;
4989 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
4990 ; CONDITIONAL BRANCH INST. AND <===
4991 ; REPLACE THE MOVE INSTRUCTION <===
4992 ; WHICH FOLLOWS W/ 774 <===
4993 015012 012742 000373 ROTXAD: MOV #373,-(R2) ;MOVE TO MAILBOX # ***** 373 *****
4994 015016 005242 ;SET MSGTYP TO FATAL ERROR
4995 015020 000000 ;SWAB DID NOT SET CC'S CORRECT

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 110 SEQ 0122
 CFKAAC.P11 18-OCT-78 11:01 T171 TEST MODE 0 W/ SWAB INST.

```

4995 015022 022780 000377 SB0: CMP #377,R0 ;CHECK RESULT
4996 015026 001404 BEQ TST172 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4997 ; ; CONDITIONAL BRANCH INST. AND <====
4998 ; ; REPLACE THE MOVE INSTRUCTION <====
4999 ; ; WHICH FOLLOWS W/ 765 <====
5000 ; ;
5001 015030 012742 000374 MOV #374,-(R2) ;MOVE TO MAILBOX # ***** 374 *****
5002 015034 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5003 015036 000000 HALT ;RESULT OF SWAB MODE 0 FAILED
5004 ; ; OR SEQUENCE ERROR
5005 ;
5006 ;
5007 ;
5008 ;
5009 ; THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION. THE TEST
5010 ; PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE ADDRESSING
5011 ; REGISTER IN THE MODE 1 SWAB. THE DATA RESULTS ARE CHECKED WITH
5012 ; A COMPARE.
5013 ;
5014 ;
5015 ; *****
5016 ; TEST 172 TEST MODE 1 W/ SWAB INST
5017 ; *****
5018 ; TST172: INC (R2) ;UPDATE TEST NUMBER
5019 015040 005212 000172 CMP #172,(R2) ;SEQUENCE ERROR?
5020 015042 022712 BNE TST173-10 ;BR TO ERROR HALT ON SEQ ERROR
5021 015046 012737 125652 000000 MOV #125652,@#0 ;MOVE TEST PATTERN TO LOC. 0
5022 015056 005000 CLR R0 ;R0=0
5023 015060 000310 SWAB (R0) ;TRY SWAB MODE 1
5024 015062 022737 125253 000000 CMP #125253,@#0 ;CHECK RESULT
5025 015070 001404 BEQ TST173 ;CHECK RESULT
5026 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5027 ; ; CONDITIONAL BRANCH INST. AND <====
5028 ; ; REPLACE THE MOVE INSTRUCTION <====
5029 ; ; WHICH FOLLOWS W/ 767 <====
5030 015072 012742 000375 MOV #375,-(R2) ;MOVE TO MAILBOX # ***** 375 *****
5031 015076 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5032 015100 000000 HALT ;RESULT OF SWAB MODE 1 FAILED
5033 ; ; OR SEQUENCE ERROR

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 111 SEQ 0123
 CFKAAC.P11 18-OCT-78 11:01 T172 TEST MODE 1 W/ SWAB INST.

```

5033 ;
5034 ; *****
5035 ; THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION. THE TEST
5036 ; PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE MODE
5037 ; 2 ADDRESSING REGISTER. THE RESULTS ARE CHECKED WITH A COMPARE.
5038 ; R0 IS CHECKED FOR PROPER DECREMENTING.
5039 ;
5040 ;
5041 ; *****
5042 ; TEST 173 TEST MODE 2 W/ SWAB INST
5043 ; *****
5044 015102 005212 000173 TST173: INC (R2) ;UPDATE TEST NUMBER
5045 015104 022712 000173 CMP #173,(R2) ;SEQUENCE ERROR?
5046 015110 001020 BNE TST174-10 ;BR TO ERROR HALT ON SEQ ERROR
5047 015112 012737 125152 000000 MOV #125152,@#0 ;MOVE TEST PATTERN TO LOC. 0
5048 015120 005000 CLR R0 ;R0=0
5049 015122 000320 SWAB (R0)+ ;TRY SWAB MODE 2
5050 015124 022737 065252 000000 CMP #65252,@#0 ;CHECK RESULT
5051 015132 001404 BEQ SB2 ;CHECK RESULT
5052 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5053 ; ; CONDITIONAL BRANCH INST. AND <====
5054 ; ; REPLACE THE MOVE INSTRUCTION <====
5055 ; ; WHICH FOLLOWS W/ 767 <====
5056 015134 012742 000376 MOV #376,-(R2) ;MOVE TO MAILBOX # ***** 376 *****
5057 015140 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5058 015142 000000 HALT ;RESULT OF SWAB MODE 0 FAILED
5059 015144 162700 000002 SB2: SUB #2,R0 ;CHECK EFFECT OF REG.
5060 015150 001404 BEQ TST174 ;CHECK EFFECT OF REG.
5061 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5062 ; ; CONDITIONAL BRANCH INST. AND <====
5063 ; ; REPLACE THE MOVE INSTRUCTION <====
5064 ; ; WHICH FOLLOWS W/ 760 <====
5065 015152 012742 000377 MOV #377,-(R2) ;MOVE TO MAILBOX # ***** 377 *****
5066 015156 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5067 015160 000000 HALT ;REGISTER VALUE INCORRECT
5068 ; ; OR SEQUENCE ERROR
5069 ;
5070 ;
5071 ;
5072 ; *****
5073 ; THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION. THE TEST
5074 ; PATTERN IS MOVED TO LOC 0. A MODE 3 SWAB INSTRUCTION IS EXECUTED
5075 ; USING R7 AS THE ADDRESSING REGISTER. A COMPARE VERIFIES THE
5076 ; DATA RESULTS.
5077 ;
5078 ; *****
5079 ; TEST 174 TEST MODE 3 W/SWAB INST.
5080 ; *****
5081 015162 005212 000174 TST174: INC (R2) ;UPDATE TEST NUMBER
5082 015164 022712 000174 CMP #174,(R2) ;SEQUENCE ERROR?
5083 015170 015037 BNE TST175-10 ;BR TO ERROR HALT ON SEQ ERROR
5084 015172 012737 000377 000000 MOV #377,@#0 ;MOVE TEST PATTERN TO LOC. 0
5085 015200 000337 000000 MOV #0 ;TRY SWAB W/ MODE 3
5086 015204 022737 177400 000000 SWAB @#0 ;CHECK RESULT
5087 015212 001404 BEQ TST175 ;CHECK RESULT

```

```

5088 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5089 ; CONDITIONAL BRANCH INST. AND <====
5090 ; REPLACE THE MOVE INSTRUCTION <====
5091 ; WHICH FOLLOWS W/ 767 <====
5092 015214 012742 000400 MOV #400,-(R2) ;MOVE TO MAILBOX # ***** 400 *****
5093 015220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5094 015222 000000 HALT ;RESULT OF SWAB INCORRECT
5095 ; OR SEQUENCE ERROR

```

```

5096 ;*****
5097 ;
5098 ; THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS. THE DATA
5099 ; IS MOVED TO LOC 0. RO IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
5100 ; REGISTER. THE DATA IS CHECKED WITH A COMPARE AND RO IS CHECKED
5101 ; FOR PROPER DECREMENTING.
5102 ;*****
5103 ;
5104 ;*****
5105 ;
5106 ;TEST 175 TEST MODE 4 W/ SWAB INST
5107 ;*****
5108 TST175: INC (R2) ;UPDATE TEST NUMBER
5109 CMP #175,(R2) ;SEQUENCE ERROR?
5110 BNE TST176-10 ;BR TO ERROR HALT ON SEQ ERROR
5111 MOV #125652,@#0 ;MOVE TEST PATTERN TO LOC. 0
5112 MOV #2,RO ;SET UP REGISTER POINTER
5113 SWAB -(RO) ;TRY SWAB MODE 4
5114 CMP #125253,@#0 ;CHECK RESULT
5115 BEQ SB4
5116 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5117 ; CONDITIONAL BRANCH INST. AND <====
5118 ; REPLACE THE MOVE INSTRUCTION <====
5119 ; WHICH FOLLOWS W/ 766 <====
5120 015260 012742 000401 MOV #401,-(R2) ;MOVE TO MAILBOX # ***** 401 *****
5121 015264 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5122 015266 000000 HALT ;RESULT OF SWAB INCORRECT
5123 015270 005700 SB4: TST RO ;CHECK EFFECT ON REG.
5124 015272 001404 BEQ TST176
5125 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5126 ; CONDITIONAL BRANCH INST. AND <====
5127 ; REPLACE THE MOVE INSTRUCTION <====
5128 ; WHICH FOLLOWS W/ 760 <====
5129 015274 012742 000402 MOV #402,-(R2) ;MOVE TO MAILBOX # ***** 402 *****
5130 015300 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5131 015302 000000 HALT ;REGISTER VALUE INCORRECT
5132 ; OR SEQUENCE ERROR
5133

```

134
135
136
137
138
139
140
141
142
143
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

015304	005212		
015306	022712	000176	
015312	001021		
015314	012700	015372	
015320	012767	125125	000040
015330	000350		
015336	022767	052652	000030
	001404		
015340	012742	000403	
015344	005242		
015346	000000		
015350	020027	015370	
015354	001406		
015356			
015358			
015362	012742	000404	
015364	005242		
015364	000000		
015366			
015370	000000		
015370	015366		

```
*****  
} THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION. THE TEST USES  
} TWO LOCATIONS FOLLOWING THE TEST CODE. SB5X HOLDS THE DATA  
} SB5XAD IS A POINTER TO THE DATA LOCATION. THE DATA IS MOVED TO  
} SB5X AND RO IS SET TO TWO PLUS THE ADDRESS OF SB5XAD. FOLLOWING  
} THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA. RO IS  
} CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.  
*****  
} TEST 176 TEST MODE 5 W/ SWAB INST.  
*****  
TST176: INC (R2) ;UPDATE TEST NUMBER  
CMP #176,(R2) ;SEQUENCE ERROR?  
BNE SB5 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #SB5XAD+2,RO ;SET UP POINTER TO WORK LOCATION  
MOV #125125,SB5X ;MOVE PATTERN TO WORK LOCATION  
SWAB 6-(RO) ;TRY SWAB MODE 5  
CMP #52652,SB5X ;CHECK RESULT  
BEQ SB5A ;  
} TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
} CONDITIONAL BRANCH INST. AND <====  
} REPLACE THE MOVE INSTRUCTION <====  
} WHICH FOLLOWS W/ 766 <====  
} ***** 403 *****  
MOV #403, -(R2) ;MOVE TO MAILBOX # *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB INCORRECT  
SB5A: CMP RO, #SB5XAD ;CHECK RESULT OF REG.  
BEQ TS177 ;  
} TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
} CONDITIONAL BRANCH INST. AND <====  
} REPLACE THE MOVE INSTRUCTION <====  
} WHICH FOLLOWS W/ 757 <====  
SB5: MOV #404, -(R2) ;MOVE TO MAILBOX # ***** 404 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;REGISTER VALUE INCORRECT  
SB5X: 0 ;OR SEQUENCE ERROR  
SB5XAD: SB5X ;WORK LOCATION
```

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
201
202
203
204
205
206
207

015372	005212		
015374	022712	000177	
015400	001013		
015402	012767	125125	000030
015410	012700	015432	
015414	000360	000006	000006
015420	022760	052652	
015426	001405		
015430			
015430	012742	000405	
015434	005242		
015436	000000		
015440			
015440	000000		

```
*****  
} 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. RO, THE ADDRESSING REGISTER  
} IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.  
} THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET. THE DATA IS  
} VERIFIED WITH A COMPARE.  
*****  
} TEST 177 TEST MODE 6 W/ SWAB INST.  
*****  
TST177: INC (R2) ;UPDATE TEST NUMBER  
CMP #177,(R2) ;SEQUENCE ERROR?  
BNE SB6 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #125125,SB6X ;MOVE PATTERN TO WORK LOCATION  
MOV #SB6X-6,RO ;MOVE OFFSET POINTER TO RO  
SWAB 6-(RO) ;TRY SWAB W/ MODE 6  
CMP #52652,6(R0) ;CHECK RESULT  
BEQ TS1200 ;  
} TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
} CONDITIONAL BRANCH INST. AND <====  
} REPLACE THE MOVE INSTRUCTION <====  
} WHICH FOLLOWS W/ 765 <====  
} ***** 405 *****  
MOV #405, -(R2) ;MOVE TO MAILBOX # *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB INCORRECT  
SB6: CMP RO, #SB6XAD ;OR SEQUENCE ERROR  
SB6X: 0 ;WORK LOCATION
```

```
5208 ;*****  
5209 ; THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION. THIS TEST  
5210 ; USES TWO LOCATIONS FOLLOWING THE TEST CODE: A WORK LOCATION  
5211 ; (SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD). DATA IS MOVED  
5212 ; TO THE WORK LOCATION. R0 IS LOADED WITH 72 LESS THAN THE ADDRESS  
5213 ; OF THE ADDRESS POINTER. THE DATA IS SWAB'ED USING A MODE 7  
5214 ; INSTRUCTION WITH AN OFFSET OF +72. THE DATA IS VERIFIED WITH A  
5215 ; COMPARE.  
5216 ;*****  
5217 ; TEST 200 TEST MODE 7 W/ SWAB INST.  
5218 ;*****  
5219 ; TST200: INC (R2) ; UPDATE TEST NUMBER  
5220 ; CMP #200,(R2) ; SEQUENCE ERROR?  
5221 ; BNE SB7 ; BR TO ERROR HALT ON SEQ ERROR  
5222 ; MOV #177400,SB7X ; MOVE PATTERN TO WORK LOCATION  
5223 ; MOV #SB7XAD-72,R0 ; MOVE OFFSET POINTER TO R0  
5224 ; SWAB #72(R0) ; TRY SWAB MODE 7  
5225 ; CMP #72(R0),#377 ; CHECK RESULTS  
5226 ; BEQ TST201 ;  
5227 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
5228 ; ; CONDITIONAL BRANCH INST. AND <=====  
5229 ; ; REPLACE THE MOVE INSTRUCTION <=====  
5230 ; ; WHICH FOLLOWS W/ 765 <=====  
5231 ;  
5232 ; SB7: MOV #406,-(R2) ; MOVE TO MAILBOX # ***** 406 *****  
5233 ; INC -(R2) ; SET MSGTYP TO FATAL ERROR  
5234 ; HALT ; RESULT OF SWAB INCORRECT  
5235 ; ; OR SEQUENCE ERROR  
5236 ; SB7X: 0 ; WORK LOCATION  
5237 ; SB7XAD: SB7X ; POINTER TO WORK LOCATION  
5238 ;  
5239 ;  
5240 ;  
5241 ;
```

```
5242 ;*****  
5243 ; THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.  
5244 ; BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST  
5245 ; UTILIZES SEVERAL DIFFERENT TECHNIQUES. THE CODE IS NOT EXECUTED  
5246 ; IN A LINEAR FASHION. THE DIFFERENT MODES ARE EXECUTED IN ORDER  
5247 ; FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP  
5248 ; PROGS THRU THE TEST CODE. THE ORDER OF APPEARANCE OF THE CODE  
5249 ; IS:  
5250 ; JUMP MODE 1  
5251 ; JUMP MODE 3  
5252 ; JUMP MODE 2  
5253 ; JUMP MODE 4  
5254 ; JUMP MODE 5  
5255 ; JUMP MODE 6  
5256 ; JUMP MODE 7  
5257 ; AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE  
5258 ; JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE. EACH CODE  
5259 ; BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN  
5260 ; THAT BLOCK. A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK. FOR  
5261 ; EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS  
5262 ; OF THE PREVIOUS MODE 2 JUMP. (ANY REGISTER CHANGES ARE VERIFIED  
5263 ; AND THE SEQUENCE CHECK IS MADE). THEN THE REGISTERS ARE SETUP  
5264 ; FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE  
5265 ; CHECKER IS UPDATED AND THE JUMP IS EXECUTED.  
5266 ; IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN  
5267 ; DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT  
5268 ; THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE  
5269 ; REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)  
5270 ;*****  
5271 ; TEST 201 TEST THE JMP INSTRUCTION IN ALL MODES  
5272 ;*****  
5273 ; TST201: INC (R2) ; UPDATE TEST NUMBER  
5274 ; CMP #201,(R2) ; SEQUENCE ERROR?  
5275 ; BNE JMPCK+6 ; BR TO ERROR HALT ON SEQ ERROR  
5276 ; CLR JMPSEQ ; ESTABLISH A SEQUENCE CHECKER  
5277 ; MOV #JMP2,R0 ; SET R0=JUMP TARGET  
5278 ; JMP (R0) ; TRY JMP MODE 1  
5279 ; CMP #+2,R0 ; CHECK RESULT OF MODE 2 JUMP  
5280 ; BEQ JMP3A ;  
5281 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
5282 ; ; CONDITIONAL BRANCH INST. AND <=====  
5283 ; ; REPLACE THE MOVE INSTRUCTION <=====  
5284 ; ; WHICH FOLLOWS W/ 770 <=====  
5285 ; ; MOVE TO MAILBOX # ***** 407 *****  
5286 ; ; SET MSGTYP TO FATAL ERROR  
5287 ; ; REGISTER VALUE AFTER JMP MODE 2 INCORRECT  
5288 ; ; MAKE SURE JUMPS ARE IN SEQUENCE: JMPSEQ=1?  
5289 ; MOV #407,-(R2) ;  
5290 ; INC -(R2) ;  
5291 ; HALT ;  
5292 ; JMP3A: CMP JMPSEQ,#1 ;  
5293 ; BEQ JMP3B ;  
5294 ; ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
5295 ; ; CONDITIONAL BRANCH INST. AND <=====  
5296 ; ; REPLACE THE MOVE INSTRUCTION <=====  
5297 ; ; WHICH FOLLOWS W/ 760 <=====  
5298 ;  
5299 ;  
5300 ;
```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 118
CFKAAC.P11 18-OCT-78 11:01 T201 TEST THE JMP INSTRUCTION IN ALL MODES SEQ 0130
5298 015564 012742 000410 MOV #410,-(R2) ;MOVE TO MAILBOX # ***** 410 *****
5299 015570 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5300 015572 000000 HALT ;SHOULD BE HERE FROM JMP MODE 2 ONLY
5301 015600 012700 015606 JMP3B: MOV #I JMP4,R0 ;POINT R0 TO INDIRECT JMP ADDR.
5302 015600 005267 000252 INC JMPSEQ ;UPDATE SEQUENCE CHECKER
5303 015604 000130 JMP @-(R0)+ ;TRY JMP MODE 3
5304 015606 015640 IJMP4: JMP4 ;ADDRESS INDIRECT JUMP
5305
5306 015610 005767 000242 JMP2: TST JMPSEQ ;CHECK THAT JMPs ARE IN SEQUENCE: JMPSEQ=0?
5307 015614 001404 BEQ JMP2A
5308
5309 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5310 ; CONDITIONAL BRANCH INST. AND <====
5311 ; REPLACE THE MOVE INSTRUCTION <====
5312 ; WHICH FOLLOWS W/ 743 <====
5313 015616 012742 000411 MOV #411,-(R2) ;MOVE TO MAILBOX # ***** 411 *****
5314 015622 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5315 015624 000000 HALT ;SHOULD BE HERE FROM JMP MODE 1 ONLY
5316 015626 005267 000224 JMP2A: INC JMPSEQ ;UPDATE SEQUENCE CHECKER
5317 015632 012700 015536 INC JMP3,R0 ;SET R0=JUMP TARGET
5318 015636 000120 JMP (R0)+ ;TRY A JUMP MODE 2 TO "JMP3"
5319 015640 022700 015610 JMP4: CMP #I JMP4+2,R0 ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
5320 015644 001404 BEQ JMP4A
5321
5322 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5323 ; CONDITIONAL BRANCH INST. AND <====
5324 ; REPLACE THE MOVE INSTRUCTION <====
5325 ; WHICH FOLLOWS W/ 727 <====
5326 015646 012742 000412 MOV #412,-(R2) ;MOVE TO MAILBOX # ***** 412 *****
5327 015652 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5328 015654 000000 HALT ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5329 015656 022767 000002 000172 JMP4A: CMP #2 JMPSEQ ;CHECK JUMP SEQUENCE: JMPSEQ=2?
5330 015664 001404 BEQ JMP4B
5331
5332 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5333 ; CONDITIONAL BRANCH INST. AND <====
5334 ; REPLACE THE MOVE INSTRUCTION <====
5335 ; WHICH FOLLOWS W/ 717 <====
5336 015666 012742 000413 MOV #413,-(R2) ;MOVE TO MAILBOX # ***** 413 *****
5337 015672 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5338 015674 000000 HALT ;SHOULD BE ONLY FROM MODE 3 JUMP
5339 015676 012700 015746 JMP4B: MOV #JMP5+2,R0 ;SET UP POINTER TO JUMP TARGET
5340 015702 005267 000150 INC JMPSEQ ;UPDATE SEQUENCE CHECKER
5341 015706 000140 JMP -(R0) ;TRY JUMP MODE 4 TO "JMP4"
5342
5343 015710 022767 000004 000140 JMP6: CMP #4 JMPSEQ ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5344 015716 001404 BEQ JMP6A
5345
5346 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5347 ; CONDITIONAL BRANCH INST. AND <====
5348 ; REPLACE THE MOVE INSTRUCTION <====
5349 ; WHICH FOLLOWS W/ 702 <====
5350 015720 012742 000414 MOV #414,-(R2) ;MOVE TO MAILBOX # ***** 414 *****
5351 015724 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5352 015726 000000 HALT ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
5353 015730 012700 016376 JMP6A: MOV #JMP7+376,R0 ;SET UP OFFSET POINTER TO JUMP TARGET
5354 015734 005267 000116 INC JMPSEQ ;UPDATE JUMP SEQUENCE
5355 015740 000160 177402 JMP -376(R0) ;TRY MODE 6 JUMP
5356
5357 015744 022767 000003 000104 JMP5: CMP #3,JMPSEQ ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 119
CFKAAC.P11 18-OCT-78 11:01 T201 TEST THE JMP INSTRUCTION IN ALL MODES SEQ 0131
5354 015752 001404 BEQ JMP5A
5355
5356 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5357 ; CONDITIONAL BRANCH INST. AND <====
5358 ; REPLACE THE MOVE INSTRUCTION <====
5359 ; WHICH FOLLOWS W/ 664 <====
5360 015754 012742 000415 MOV #415,-(R2) ;MOVE TO MAILBOX # ***** 415 *****
5361 015760 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5362 015762 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5363 015764 012700 016000 JMP5A: MOV #I JMP5+2,R0 ;SET UP POINTER TO INDIRECT JUMP ADDR.
5364 015770 005267 000062 INC JMPSEQ ;UPDATE JUMP SEQUENCE
5365 015774 000150 IJMP5: JMP @-(R0) ;TRY JUMP MODE 5 TO "JMP5"
5366 015776 015710 ;INDIRECT ADDRESS POINTER
5367
5368 016000 022767 000005 000050 JMP7: CMP #5,JMPSEQ ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5369 016006 001404 BEQ JMP7A
5370
5371 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5372 ; CONDITIONAL BRANCH INST. AND <====
5373 ; REPLACE THE MOVE INSTRUCTION <====
5374 ; WHICH FOLLOWS W/ 646 <====
5375 016010 012742 000416 MOV #416,-(R2) ;MOVE TO MAILBOX # ***** 416 *****
5376 016014 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5377 016016 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5378 016020 012700 016044 JMP7A: MOV #I JMP+10,R0 ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5379 016024 005267 000026 INC JMPSEQ ;UPDATE JUMP SEQUENCE
5380 016030 000170 IJMP: JMP @-10(R0) ;TRY MODE 7 JUMP
5381 016034 016036 ;INDIRECT ADDRESS
5382
5383 016036 026727 000014 000006 JMPCK: CMP JMPSEQ,#6 ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5384 016044 001405 BEQ TST202
5385
5386 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5387 ; CONDITIONAL BRANCH INST. AND <====
5388 ; REPLACE THE MOVE INSTRUCTION <====
5389 ; WHICH FOLLOWS W/ 627 <====
5390 016046 012742 000417 MOV #417,-(R2) ;MOVE TO MAILBOX # ***** 417 *****
5391 016052 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5392 016054 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5393 016056 000000 JMPSEQ: 0 ;OR SEQUENCE ERROR

```

```

5392 ;*****
5393 ; THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
5394 ; THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
5395 ; IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST) EACH
5396 ; BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE
5397 ; CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
5398 ; THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
5399 ; SAVED ON THE STACK AND FINALLY CHECKING THAT ANY MODE ADDRESS
5400 ; REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
5401 ; SUCCESSFUL. R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
5402 ; IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
5403 ; DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT
5404 ; THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
5405 ; REGISTER SAVED).
5406 ;*****
5407 ;*****
5408 ;*****
5409 ;*****
5410 ;*****
5411 ;*****
5412 ;*****
5413 ;*****
5414 ;*****
5415 ;*****
5416 ;*****
5417 ;*****
5418 ;*****
5419 ;*****
5420 ;*****
5421 ;*****
5422 ;*****
5423 ;*****
5424 ;*****
5425 ;*****
5426 ;*****
5427 ;*****
5428 ;*****
5429 ;*****
5430 ;*****
5431 ;*****
5432 ;*****
5433 ;*****
5434 ;*****
5435 ;*****
5436 ;*****
5437 ;*****
5438 ;*****
5439 ;*****
5440 ;*****
5441 ;*****
5442 ;*****
5443 ;*****
5444 ;*****
5445 ;*****
5446 ;*****
5447 ;*****
5448 ;*****
5449 ;*****
5450 ;*****
5451 ;*****
5452 ;*****
5453 ;*****
5454 ;*****
5455 ;*****
5456 ;*****
5457 ;*****
5458 ;*****
5459 ;*****
5460 ;*****
5461 ;*****
5462 ;*****
5463 ;*****
5464 ;*****
5465 ;*****
5466 ;*****
5467 ;*****
5468 ;*****
5469 ;*****
5470 ;*****
5471 ;*****
5472 ;*****
5473 ;*****
5474 ;*****
5475 ;*****
5476 ;*****
5477 ;*****
5478 ;*****
5479 ;*****
5480 ;*****
5481 ;*****
5482 ;*****
5483 ;*****
5484 ;*****
5485 ;*****
5486 ;*****
5487 ;*****
5488 ;*****
5489 ;*****
5490 ;*****
5491 ;*****
5492 ;*****
5493 ;*****
5494 ;*****
5495 ;*****
5496 ;*****
5497 ;*****
5498 ;*****
5499 ;*****
5500 ;*****
5501 ;*****
5502 ;*****
5503 ;*****

```

```

5448 ;*****
5449 ;*****
5450 ;*****
5451 ;*****
5452 ;*****
5453 ;*****
5454 ;*****
5455 ;*****
5456 ;*****
5457 ;*****
5458 ;*****
5459 ;*****
5460 ;*****
5461 ;*****
5462 ;*****
5463 ;*****
5464 ;*****
5465 ;*****
5466 ;*****
5467 ;*****
5468 ;*****
5469 ;*****
5470 ;*****
5471 ;*****
5472 ;*****
5473 ;*****
5474 ;*****
5475 ;*****
5476 ;*****
5477 ;*****
5478 ;*****
5479 ;*****
5480 ;*****
5481 ;*****
5482 ;*****
5483 ;*****
5484 ;*****
5485 ;*****
5486 ;*****
5487 ;*****
5488 ;*****
5489 ;*****
5490 ;*****
5491 ;*****
5492 ;*****
5493 ;*****
5494 ;*****
5495 ;*****
5496 ;*****
5497 ;*****
5498 ;*****
5499 ;*****
5500 ;*****
5501 ;*****
5502 ;*****
5503 ;*****

```

```

CPKAACO 11/34 BSC INST TST MACV11 30A(1052) 18-OCT-78 11:06 PAGE 122
CPKAAC.P11 18-OCT-78 11:01 T202 TEST JSR INSTRUCTION W/ ALL MODES SEQ 0134

000102 JSR6B: INC #0#JSRSEQ ;UPDATE SEQUENCE CHECKER
          JSR R1,JSR7 ;TRY JSR MODE 6
          JSR5: CMP #3,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=3?
          BNE JSR5A ;BRANCH IF OUT OF SEQUENCE
          CMP #JSR6,R1 ;PROPER PC SAVED?
          BNE JSR5A ;BRANCH IF PC WRONG
          CMP #JSR5,R0 ;CHECK MODE 4 REGISTER
          BEQ JSR5B

          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 623 <====

016422 JSR5A: MOV #425,-(R2) ;MOVE TO MAILBOX # ***** 425 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;JSR MODE 4 MALFUNCTIONED
          JSR5B: INC #0#JSRSEQ ;UPDATE SEQUENCE CHECKER
          MOV #JSR6AD+2,R0 ;POINT R0 TO TARGET ADDRESS
          JSR R1,@(R0) ;TRY JSR MODE 5

016444 JSR7: CMP #5,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=5?
          BNE JSR7A ;BRANCH IF OUT OF SEQUENCE
          CMP #JSR5,R1 ;PROPER PC SAVED?
          BEQ JSR7B

          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 603 <====

016462 JSR7A: MOV #426,-(R2) ;MOVE TO MAILBOX # ***** 426 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;JSR MODE 6 FAILED
          JSR7B: INC #0#JSRSEQ ;UPDATE SEQUENCE CHECKER
          JSR R1,@JSRCKAD ;TRY JSR MODE 7

          JSR6AD: JSR6 016332 ;MODE 5 TARGET ADDRESS
          JSRCKAD: JSRCK 016510 ;MODE 7 TARGET ADDRESS
          JSRSEQ: 0 000000 ;SEQUENCE CHECKER

177770 JSRCK: CMP #6,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=6?
          BNE JSRCK1 ;BRANCH IF OUT OF SEQUENCE
          CMP #JSR6AD,R1 ;PROPER PC SAVED?
          BEQ TST203

          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 561 <====

016526 JSRCK1: MOV #427,-(R2) ;MOVE TO MAILBOX # ***** 427 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;JSR MODE 7 MALFUNCTIONED
          ; OR SEQUENCE ERROR

```

```

CPKAACO 11/34 BSC INST TST MACV11 30A(1052) 18-OCT-78 11:06 PAGE 123
CPKAAC.P11 18-OCT-78 11:01 T202 TEST JSR INSTRUCTION W/ ALL MODES SEQ 0135

;*****
; THIS TEST VERIFIES THE RTS INSTRUCTION. THE STACK POINTER
; IS INITIALIZED AND A TEST PATTERN STORED ON STACK. R0 IS LOADED
; WITH RETURN ADDRESS. AN RTS IS EXECUTED, AND, AT THE TARGET
; ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
; STACK.
;*****
TEST 203 TEST RTS INSTRUCTION
TST203: INC (R2) ;UPDATE TEST NUMBER
          CMP #203,(R2) ;SEQUENCE ERROR?
          BNE TST204-10 ;BR TO ERROR HALT ON SEQ ERROR
          MOV #TST204-10,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 <====

016536 RTS1: MOV #430,-(R2) ;MOVE TO MAILBOX # ***** 430 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;RTS FAILED
          CMP #52525,R0 ;CHECK THAT R0 RESTORED FROM STACK
          BEQ TST204

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

016602 RTS1: MOV #431,-(R2) ;MOVE TO MAILBOX # ***** 431 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;RTS MALFUNCTIONED
          ; OR SEQUENCE ERROR

```



```
*****
;
; THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
; OF FOUR INSTRUCTIONS. THE GROUP CONSISTS OF THE INSTRUCTIONS:
; MOV, BIC, BIT AND BIS. THESE INSTRUCTIONS ARE SIMILAR IN THE
; WAY THEY AFFECT 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 7 AND V BITS
; ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS, THE C-BIT
; IS LOADED WITH THE DESIRED RESULT. THE INSTRUCTION IS EXECUTED
; WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VERIFIED WITH
; A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS. THE DATA IS CHOSEN
; TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
*****
;*****
;TEST 204 TEST MOV INSTRUCTION
;*****
TST204: INC (R2) ;UPDATE TEST NUMBER
        CMP #204,(R2) ;SEQUENCE ERROR?
        BNE TST205-10 ;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 #432,-(R2) ;MOVE TO MAILBOX # ***** 432 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;MOV DID NOT SET CC'S CORRECTLY
;*****
MOV2: SCC ;CC=1011
        CLZ ;CC=0101
        MOV #0,R0 ;C OR Z = 0?
        BHI MOV3
        BVS MOV3
        BPL TST205
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
;*****
MOV3: MOV #433,-(R2) ;MOVE TO MAILBOX # ***** 433 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;MOV DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR
;*****
;TEST 205 TEST BIT INSTRUCTION
;*****
TST205: INC (R2) ;UPDATE TEST NUMBER
        CMP #205,(R2) ;SEQUENCE ERROR?
        BNE TST206-10 ;BR TO ERROR HALT ON SEQ ERROR
;*****
```

```
*****
;*****
;TEST 206 TEST BIC INSTRUCTION
;*****
TST206: INC (R2) ;UPDATE TEST NUMBER
        CMP #206,(R2) ;SEQUENCE ERROR?
        BNE TST207-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #177777,R0
        SCC ;CC=0110
        +CLN!CLC
        BIC #77777,R0 ;CC=1000
        BLOS BIC1
        BVS BIC1
        BMI BIC2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
;*****
BIC1: MOV #436,-(R2) ;MOVE TO MAILBOX # ***** 436 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BIC DID NOT SET CC'S CORRECTLY
;*****
BIC2: SCC ;CC=1011
        CLZ ;CC=0101
        MOV #100000,R0
        BHI BIC3
        BPL TST206
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
;*****
BIT1: MOV #434,-(R2) ;MOVE TO MAILBOX # ***** 434 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BIT DID NOT SET CC'S CORRECTLY
;*****
BIT2: SCC ;CC=1011
        CLZ ;CC=0101
        BIT #77776,R0
        BHI BIT3
        BVS BIT3
        BPL TST206
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
;*****
BIT3: MOV #435,-(R2) ;MOVE TO MAILBOX # ***** 435 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BIT DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR
;*****
;TEST 207 TEST BIC INSTRUCTION
;*****
TST207: INC (R2) ;UPDATE TEST NUMBER
        CMP #207,(R2) ;SEQUENCE ERROR?
        BNE TST208-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #177777,R0
        SCC ;CC=0110
        +CLN!CLC
        BIC #77777,R0 ;CC=1000
        BLOS BIC1
        BVS BIC1
        BMI BIC2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
;*****
BIC1: MOV #436,-(R2) ;MOVE TO MAILBOX # ***** 436 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BIC DID NOT SET CC'S CORRECTLY
;*****
BIC2: SCC ;CC=1011
        CLZ ;CC=0101
        MOV #100000,R0
        BHI BIC3
        BPL TST207
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
;*****
```

```

5702 017042 102401      BVS      BIC3
5703 017044 100004      BPL      TST207
5704
5705
5706
5707
5708 017046
5709 017046 012742 000437      BIC3:    MOV      #437,-(R2)      ;MOVE TO MAILBOX # ***** 437 *****
5710 017052 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5711 017054 000000          HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5712
5713
5714
5715
5716 017056 005212          TST207: INC      (R2)          ;UPDATE TEST NUMBER
5717 017060 022712 000207      CMP      #207,(R2)      ;SEQUENCE ERROR?
5718 017064 061025          BNE     TST210-10      ;BR TO ERROR HALT ON SEQ ERROR
5719 017066 005000          CLR      R0            ;R0=0
5720 017070 000277          SCC      ;CC=1010
5721 017072 000251          +CLN!CLC
5722 017074 052700          BIS     #0,R0          ;CC=0100 R0=0
5723 017100 103403          BVS     BIS1
5724 017102 102402          BVS     BIS1
5725 017104 100401          BMI     BIS1
5726 017106 001404          BEQ     BIS2
5727
5728
5729
5730
5731 017110
5732 017110 012742 000440      BIS1:    MOV      #440,-(R2)      ;MOVE TO MAILBOX # ***** 440 *****
5733 017114 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5734 017116 000000          HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5735 017120 000277          BIS2:    HALT                    ;CC=0111
5736 017122 000250          SCC      CLN
5737 017124 052700 177777      BIS     #177777,R0      ;CC=1001
5738 017130 103003          BCC     BIS3
5739 017132 102402          BVS     BIS3
5740 017134 001401          BEQ     BIS3
5741 017136 100404          BMI     TST210
5742
5743
5744
5745
5746 017140
5747 017140 012742 000441      BIS3:    MOV      #441,-(R2)      ;MOVE TO MAILBOX # ***** 441 *****
5748 017144 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5749 017146 000000          HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766 017150 005212          TST210: INC      (R2)          ;UPDATE TEST NUMBER
5767 017152 022712 000210      CMP      #210,(R2)      ;SEQUENCE ERROR?
5768 017156 001037          BNE     TST211-10      ;BR TO ERROR HALT ON SEQ ERROR
5769 017160 012700 077777      MOV      #077777,R0      ;R0=077777
5770 017164 000257          CCC      ;CC=0100
5771 017166 000264          SEZ
5772 017170 005200          INC     R0            ;CC=1010 R0=10000
5773 017172 101402          BLOS   INC1
5774 017174 100001          BPL     INC1
5775 017176 102404          BVS     INC2
5776
5777
5778
5779
5780 017200
5781 017200 012742 000442      INC1:    MOV      #442,-(R2)      ;MOVE TO MAILBOX # ***** 442 *****
5782 017204 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5783 017206 000000          HALT                    ;INC DID NOT SET CC'S CORRECTLY
5784 017210 052700 077777      INC2:    BIS     #777777,R0      ;R0=177777
5785 017214 000261          SEC      ;CC=1011
5786 017216 000244          CLZ
5787 017220 005200          INC     R0            ;CC=0101 R0=0
5788 017222 100403          BMI     INC3
5789 017224 102402          BVS     INC3
5790 017226 103001          BCC     INC3
5791 017230 001404          BEQ     INC4
5792
5793
5794
5795
5796 017232
5797 017232 012742 000443      INC3:    MOV      #443,-(R2)      ;MOVE TO MAILBOX # ***** 443 *****
5798 017236 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5799 017240 000000          HALT                    ;INC DID NOT SET CC'S CORRECTLY
5800
5801 017242 000277          INC4:    SCC      ;CC=1110
5802 017244 000241          CLC
5803 017246 005200          INC     R0            ;CC=0000 R0=1
5804 017250 101402          BLOS   INC5
5805 017252 100401          BMI     INC5
5806 017254 100004          BPL     TST211
    
```

```

5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766 017150 005212          TST210: INC      (R2)          ;UPDATE TEST NUMBER
5767 017152 022712 000210      CMP      #210,(R2)      ;SEQUENCE ERROR?
5768 017156 001037          BNE     TST211-10      ;BR TO ERROR HALT ON SEQ ERROR
5769 017160 012700 077777      MOV      #077777,R0      ;R0=077777
5770 017164 000257          CCC      ;CC=0100
5771 017166 000264          SEZ
5772 017170 005200          INC     R0            ;CC=1010 R0=10000
5773 017172 101402          BLOS   INC1
5774 017174 100001          BPL     INC1
5775 017176 102404          BVS     INC2
5776
5777
5778
5779
5780 017200
5781 017200 012742 000442      INC1:    MOV      #442,-(R2)      ;MOVE TO MAILBOX # ***** 442 *****
5782 017204 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5783 017206 000000          HALT                    ;INC DID NOT SET CC'S CORRECTLY
5784 017210 052700 077777      INC2:    BIS     #777777,R0      ;R0=177777
5785 017214 000261          SEC      ;CC=1011
5786 017216 000244          CLZ
5787 017220 005200          INC     R0            ;CC=0101 R0=0
5788 017222 100403          BMI     INC3
5789 017224 102402          BVS     INC3
5790 017226 103001          BCC     INC3
5791 017230 001404          BEQ     INC4
5792
5793
5794
5795
5796 017232
5797 017232 012742 000443      INC3:    MOV      #443,-(R2)      ;MOVE TO MAILBOX # ***** 443 *****
5798 017236 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5799 017240 000000          HALT                    ;INC DID NOT SET CC'S CORRECTLY
5800
5801 017242 000277          INC4:    SCC      ;CC=1110
5802 017244 000241          CLC
5803 017246 005200          INC     R0            ;CC=0000 R0=1
5804 017250 101402          BLOS   INC5
5805 017252 100401          BMI     INC5
5806 017254 100004          BPL     TST211
    
```

```

5807 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5808 ; CONDITIONAL BRANCH INST. AND <====
5809 ; REPLACE THE MOVE INSTRUCTION <====
5810 ; WHICH FOLLOWS W/ 741 <====
5811 INC5: MOV #444,-(R2) ;MOVE TO MAILBOX # ***** 444 *****
5812 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5813 HALT ;INC DID NOT SET CC'S CORRECTLY
5814 ; OR SEQUENCE ERROR
5815 ;*****
5816 ;*****
5817 ;*****
5818 ;*****
5819 ;*****
5820 ;*****
5821 ;*****
5822 ;*****
5823 ;*****
5824 ;*****
5825 ;*****
5826 ;*****
5827 ;*****
5828 ;*****
5829 ;*****
5830 ;*****
5831 ;*****
5832 ;*****
5833 ;*****
5834 ;*****
5835 ;*****
5836 ;*****
5837 ;*****
5838 ;*****
5839 ;*****
5840 ;*****
5841 ;*****
5842 ;*****
5843 ;*****
5844 ;*****
5845 ;*****
5846 ;*****
5847 ;*****
5848 ;*****
5849 ;*****
5850 ;*****
5851 ;*****
5852 ;*****
5853 ;*****
5854 ;*****
5855 ;*****
5856 ;*****
5857 ;*****
5858 ;*****
5859 ;*****
5860 ;*****
5861 ;*****
5862 ;*****
    
```

```

5863 ;*****
5864 ;*****
5865 ;*****
5866 ;*****
5867 ;*****
5868 ;*****
5869 ;*****
5870 ;*****
5871 ;*****
5872 ;*****
5873 ;*****
5874 ;*****
5875 ;*****
5876 ;*****
5877 ;*****
5878 ;*****
5879 ;*****
5880 ;*****
5881 ;*****
5882 ;*****
5883 ;*****
    
```

```

5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897 017430 005212 000212
5898 017432 022712
5899 017436 010014
5900 017440 000277
5901 017442 000244
5902 017444 005000
5903 017446 100403
5904 017450 102402
5905 017452 103401
5906 017454 001404
5907
5908
5909
5910
5911
5912 017456 012742 000451
5913 017462 005242
5914 017464 000000
5915
5916
5917
5918
5919
5920 017466 005212 000213
5921 017470 022712
5922 017474 010014
5923 017476 000277
5924 017500 000244
5925 017502 005700
5926 017504 100403
5927 017506 102402
5928 017512 001404
5929
5930
5931
5932
5933
5934 017514
5935 017514 012742 000452
5936 017520 005242
5937 017522 000000
5938 017524 005300
5939 017526 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 212 TEST CLR INSTRUCTION
;*****
TST212: INC (R2) ;UPDATE TEST NUMBER
CMP #212,(R2) ;SEQUENCE ERROR?
BNE TST213-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
CLR R0 ;CC=0100 R0=0
BMI CLR1
BVS CLR1
BCS CLR1
BEQ TST213

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

CLR1: MOV #451,-(R2) ;MOVE TO MAILBOX # ***** 451 *****
INC ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

;*****
; TEST 213 TEST TST INSTRUCTION
;*****
TST213: INC (R2) ;UPDATE TEST NUMBER
CMP #213,(R2) ;SEQUENCE ERROR?
BNE TST214-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
TST R0 ;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 #452,-(R2) ;MOVE TO MAILBOX # ***** 452 *****
INC ;SET MSGTYP TO FATAL ERROR
HALT ;TEST DID NOT SET CC'S CORRECTLY
TEST2: DEC R0 ;MAKE R0 NEGATIVE
SCC ;CC=0111

```

5940 017530 000250
5941 017532 005700
5942 017534 101402
5943 017536 102401
5944 017540 100404
5945
5946
5947
5948
5949 017542
5950 017542 012742 000453
5951 017546 005242
5952 017550 000000
5953
5954
5955
5956
5957 017552 005212 000214
5958 017554 022712
5959 017560 001023
5960 017562 012700 170000
5961 017566 000277
5962 017570 000250
5963 017572 000300
5964 017574 101402
5965 017576 102401
5966 017600 100404
5967
5968
5969
5970
5971 017602
5972 017602 012742 000454
5973 017606 005242
5974 017610 000000
5975 017612 000277
5976 017614 000244
5977 017616 000300
5978 017620 102402
5979 017622 103402
5980 017624 100401
5981 017626 001404
5982
5983
5984
5985
5986 017630
5987 017630 012742 000455
5988 017634 005242
5989 017636 000000

```

CLN
TST R0 ;CC=1000
BLOS TEST3
BVS TEST3
BMI TST214

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

TEST3: MOV #453,-(R2) ;MOVE TO MAILBOX # ***** 453 *****
INC ;SET MSGTYP TO FATAL ERROR
HALT ;TEST DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

;*****
; TEST 214 TEST SWAB INSTRUCTION
;*****
TST214: INC (R2) ;UPDATE TEST NUMBER
CMP #214,(R2) ;SEQUENCE ERROR?
BNE TST215-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #170000,R0 ;R0=170000
SCC ;CC=0111
CLN
SWAB R0 ;CC=1000 R0=360
BLOS SWB1
BVS SWB1
BMI SWB2

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

SWB1: MOV #454,-(R2) ;MOVE TO MAILBOX # ***** 454 *****
INC ;SET MSGTYP TO FATAL ERROR
HALT ;SWAB DID NOT SET CC'S CORRECTLY
SWB2: SCC ;CC=1011
CLZ
SWAB R0 ;CC=0100 R0=170000
BVS SWB3
BCS SWB3
BMI SWB3
BEQ TST215

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

SWB3: MOV #455,-(R2) ;MOVE TO MAILBOX # ***** 455 *****
INC ;SET MSGTYP TO FATAL ERROR
HALT


```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 134
CFKAAC.P11 18-OCT-78 11:01 T216 TEST ADC INSTRUCTION SEQ 0146

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

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 135
CFKAAC.P11 18-OCT-78 11:01 T216 TEST ADC INSTRUCTION SEQ 0147

6130 ;*****
6131 ;
6132 ;
6133 ; THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
6134 ; CMP, AND COM INSTRUCTIONS. EACH OF THESE INSTRUCTIONS GENERATE
6135 ; THE C AND V BITS IDENTICALLY. THE CONDITION CODES ARE PRESET
6136 ; THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
6137 ; OF CONDITIONAL BRANCH INSTRUCTIONS. THIS PROCEDURE IS REPEATED
6138 ; SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
6139 ; COMBINATIONS OF THE C AND V BITS.
6140 ;*****
6141 ;
6142 ;TEST 217 TEST NEG INSTRUCTION
6143 ;*****
6144 020142 005212 TST217: INC (R2) ;UPDATE TEST NUMBER
6145 020144 022712 CMP #217,(R2) ;SEQUENCE ERROR?
6146 020150 001042 BNE TST220-10 ;BR TO ERROR HALT ON SEQ ERROR
6147 020152 012700 MOV #1,R0
6148 020156 000277 SCC ;CC=0110
6149 020160 000251 +CLN!CLC
6150 020162 005400 NEG R0 ;CC=1001 R0=177777
6151 020164 103003 BCC NEG1
6152 020166 102402 BVS NEG1
6153 020170 001401 BEQ NEG1
6154 020172 100404 BMI NEG2
6155 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6156 ; CONDITIONAL BRANCH INST. AND <====
6157 ; REPLACE THE MOVE INSTRUCTION <====
6158 ; WHICH FOLLOWS W/ 767 <====
6159 020174
6160 020174 012742 000466 NEG1: MOV #466,-(R2) ;MOVE TO MAILBOX # ***** 466 *****
6161 020200 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6162 020202 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
6163 020204 042700 BIC #77777,R0 ;
6164 020210 000257 CCC ;CC=0100
6165 020212 000264 SEZ
6166 020214 005400 NEG R0 ;CC=1011 R0=100000
6167 020216 102003 BVC NEG3
6168 020220 103002 BCC NEG3
6169 020222 001401 BEQ NEG3
6170 020224 100404 BMI NEG4
6171 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6172 ; CONDITIONAL BRANCH INST. AND <====
6173 ; REPLACE THE MOVE INSTRUCTION <====
6174 ; WHICH FOLLOWS W/ 752 <====
6175 020226
6176 020226 012742 000467 NEG3: MOV #467,-(R2) ;MOVE TO MAILBOX # ***** 467 *****
6177 020232 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6178 020234 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
6179 020236 005000 CLR R0 ;
6180 020240 000277 SCC ;CC=1011
6181 020242 000244 CLZ
6182 020244 005400 NEG R0 ;CC=0100 R0=0
6183 020246 102403 BVS NEG5
6184 020250 103402 BCS NEG5
6185 020252 001001 BNE NEG5

```

```

6186 020254 100004 BPL TST220 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6187 ; ; ; ; ; <====
6188 ; ; ; ; ; <====
6189 ; ; ; ; ; <====
6190 ; ; ; ; ; <====
6191 020256 ; ; ; ; ; <====
6192 020256 012742 000470 NEG5: MOV #470,-(R2) ;MOVE TO MAILBOX # ***** 470 *****
6193 020256 005242 ;SET MSGTYP TO FATAL ERROR
6194 020264 000000 ;NEG DID NOT SET CC'S CORRECTLY
6195 ; ; ; ; ; <====
6196 ; ; ; ; ; <====
6197 ; ; ; ; ; <====
6198 ; ; ; ; ; <====
6199 ; ; ; ; ; <====
6200 020266 005212 ;*****
6201 020270 022712 000220 TST220: INC (R2) ;UPDATE TEST NUMBER
6202 020274 001060 ;SEQUENCE ERROR?
6203 020276 012700 000005 MOV #5,R0 ;BR TO ERROR HALT ON SEQ ERROR
6204 020302 000257 ; ; ; ; ; <====
6205 020304 020271 ; ; ; ; ; <====
6206 020306 022700 000005 +SENISEC #5,R0 ;CC=1010
6207 020312 101002 ; ; ; ; ; <====
6208 020314 102401 ; ; ; ; ; <====
6209 020316 100004 ; ; ; ; ; <====
6210 ; ; ; ; ; <====
6211 ; ; ; ; ; <====
6212 ; ; ; ; ; <====
6213 ; ; ; ; ; <====
6214 020320 ; ; ; ; ; <====
6215 020320 012742 000471 CMP1: MOV #471,-(R2) ;MOVE TO MAILBOX # ***** 471 *****
6216 020324 005242 ;SET MSGTYP TO FATAL ERROR
6217 020326 000000 ;CMP DID NOT SET CC'S CORRECTLY
6218 020330 012700 100000 CMP2: MOV #10000,R0
6219 020334 000277 ; ; ; ; ; <====
6220 020336 000242 ; ; ; ; ; <====
6221 020340 020027 ; ; ; ; ; <====
6222 020344 101402 ; ; ; ; ; <====
6223 020346 102001 ; ; ; ; ; <====
6224 020350 100004 ; ; ; ; ; <====
6225 ; ; ; ; ; <====
6226 ; ; ; ; ; <====
6227 ; ; ; ; ; <====
6228 ; ; ; ; ; <====
6229 ; ; ; ; ; <====
6230 020352 ; ; ; ; ; <====
6231 020352 012742 000472 CMP3: MOV #472,-(R2) ;MOVE TO MAILBOX # ***** 472 *****
6232 020356 005242 ;SET MSGTYP TO FATAL ERROR
6233 020360 000000 ;CMP DID NOT SET CC'S CORRECTLY
6234 020362 052700 040000 CMP4: BIS #40000,R0
6235 020366 000257 ; ; ; ; ; <====
6236 020370 000264 ; ; ; ; ; <====
6237 020372 022700 040000 ; ; ; ; ; <====
6238 020376 102003 ; ; ; ; ; <====
6239 020400 103002 ; ; ; ; ; <====
6240 020402 001401 ; ; ; ; ; <====
6241 020404 100404 ; ; ; ; ; <====
6242 ; ; ; ; ; <====
6243 ; ; ; ; ; <====
6244 ; ; ; ; ; <====
6245 ; ; ; ; ; <====
6246 020406 ; ; ; ; ; <====
6247 020406 012742 000473 CMP5: MOV #473,-(R2) ;MOVE TO MAILBOX # ***** 473 *****
6248 020412 005242 ;SET MSGTYP TO FATAL ERROR
6249 020414 000000 ;CMP DID NOT SET CC'S CORRECTLY
6250 020422 000277 ; ; ; ; ; <====
6251 020424 022700 177777 CMP6: BIC #40000,R0
6252 020430 101402 ; ; ; ; ; <====
6253 020432 102401 ; ; ; ; ; <====
6254 020434 100004 ; ; ; ; ; <====
6255 ; ; ; ; ; <====
6256 ; ; ; ; ; <====
6257 ; ; ; ; ; <====
6258 ; ; ; ; ; <====
6259 020436 ; ; ; ; ; <====
6260 020436 012742 000474 CMP7: MOV #474,-(R2) ;MOVE TO MAILBOX # ***** 474 *****
6261 020442 005242 ;SET MSGTYP TO FATAL ERROR
6262 020444 000000 ;CMP DID NOT SET CC'S CORRECTLY
6263 ; ; ; ; ; <====
6264 ; ; ; ; ; <====
6265 ; ; ; ; ; <====
6266 ; ; ; ; ; <====
6267 ; ; ; ; ; <====
6268 020446 005212 ;*****
6269 020450 022712 000221 TST221: INC (R2) ;UPDATE TEST NUMBER
6270 020454 011010 ;SEQUENCE ERROR?
6271 020456 012700 177777 MOV #1,R0 ;BR TO ERROR HALT ON SEQ ERROR
6272 020462 000257 ; ; ; ; ; <====
6273 020464 000265 ; ; ; ; ; <====
6274 020466 005100 ; ; ; ; ; <====
6275 020470 101002 ; ; ; ; ; <====
6276 020472 102401 ; ; ; ; ; <====
6277 020474 100004 ; ; ; ; ; <====
6278 ; ; ; ; ; <====
6279 ; ; ; ; ; <====
6280 ; ; ; ; ; <====
6281 ; ; ; ; ; <====
6282 020476 ; ; ; ; ; <====
6283 020476 012742 000475 COM1: MOV #475,-(R2) ;MOVE TO MAILBOX # ***** 475 *****
6284 020502 005242 ;SET MSGTYP TO FATAL ERROR
6285 020504 000000 ;COM DID NOT SET CC'S CORRECTLY
6286 ; ; ; ; ; <====
6287 ; ; ; ; ; <====

```

```

6242 ; ; ; ; ; <====
6243 ; ; ; ; ; <====
6244 ; ; ; ; ; <====
6245 ; ; ; ; ; <====
6246 020406 ; ; ; ; ; <====
6247 020412 005242 000473 CMP5: MOV #473,-(R2) ;MOVE TO MAILBOX # ***** 473 *****
6248 020414 000000 ;SET MSGTYP TO FATAL ERROR
6249 020416 042700 040000 CMP6: BIC #40000,R0
6250 020422 000277 ; ; ; ; ; <====
6251 020424 022700 177777 ; ; ; ; ; <====
6252 020430 101402 ; ; ; ; ; <====
6253 020432 102401 ; ; ; ; ; <====
6254 020434 100004 ; ; ; ; ; <====
6255 ; ; ; ; ; <====
6256 ; ; ; ; ; <====
6257 ; ; ; ; ; <====
6258 ; ; ; ; ; <====
6259 020436 ; ; ; ; ; <====
6260 020436 012742 000474 CMP7: MOV #474,-(R2) ;MOVE TO MAILBOX # ***** 474 *****
6261 020442 005242 ;SET MSGTYP TO FATAL ERROR
6262 020444 000000 ;CMP DID NOT SET CC'S CORRECTLY
6263 ; ; ; ; ; <====
6264 ; ; ; ; ; <====
6265 ; ; ; ; ; <====
6266 ; ; ; ; ; <====
6267 ; ; ; ; ; <====
6268 020446 005212 ;*****
6269 020450 022712 000221 TST221: INC (R2) ;UPDATE TEST NUMBER
6270 020454 011010 ;SEQUENCE ERROR?
6271 020456 012700 177777 MOV #1,R0 ;BR TO ERROR HALT ON SEQ ERROR
6272 020462 000257 ; ; ; ; ; <====
6273 020464 000265 ; ; ; ; ; <====
6274 020466 005100 ; ; ; ; ; <====
6275 020470 101002 ; ; ; ; ; <====
6276 020472 102401 ; ; ; ; ; <====
6277 020474 100004 ; ; ; ; ; <====
6278 ; ; ; ; ; <====
6279 ; ; ; ; ; <====
6280 ; ; ; ; ; <====
6281 ; ; ; ; ; <====
6282 020476 ; ; ; ; ; <====
6283 020476 012742 000475 COM1: MOV #475,-(R2) ;MOVE TO MAILBOX # ***** 475 *****
6284 020502 005242 ;SET MSGTYP TO FATAL ERROR
6285 020504 000000 ;COM DID NOT SET CC'S CORRECTLY
6286 ; ; ; ; ; <====
6287 ; ; ; ; ; <====

```

```
6288 ;*****  
6289 ;  
6290 ; THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB  
6291 ; AND SBC INSTRUCTIONS. BOTH OF THESE INSTRUCTIONS HANDLE THE  
6292 ; C AND V BITS IDENTICALLY. THE PROCEDURE IS TO PRESET THE CONDITION  
6293 ; CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND  
6294 ; THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL  
6295 ; BRANCHES. THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT  
6296 ; DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.  
6297 ;  
6298 ;*****  
6299 ;  
6300 ; TEST 222 TEST SUB INSTRUCTION  
6301 ;*****  
6302 TST222: INC (R2) ;UPDATE TEST NUMBER  
6303 CMP #223,(R2) ;SEQUENCE ERROR?  
6304 BNE #125252-10 ;BR TO ERROR HALT ON SEQ ERROR  
6305 MOV #125252,R0  
6306 CCC ;CC=1010  
6307 +SENISEC  
6308 SUB #125252,R0 ;CC=0101 R0=0  
6309 BHI SUB1  
6310 BVS SUB1  
6311 BPL SUB2  
6312 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6313 ; CONDITIONAL BRANCH INST. AND <====  
6314 ; REPLACE THE MOVE INSTRUCTION <====  
6315 ; WHICH FOLLOWS W/ 767 <====  
6316 020540 SUB1: MOV #476,-(R2) ;MOVE TO MAILBOX # ***** 476 *****  
6317 020540 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6318 020544 005242 HALT ;SUB DID NOT SET CC'S CORRECTLY  
6319 020546 000000 SUB2: BIC #100000,R0 ;CC=1101  
6320 020550 100000 SCC  
6321 020554 000277 CLV  
6322 020556 000242 SUB #77777,R0 ;CC=0010 R0=1  
6323 020560 162700 BLOS SUB3  
6324 020564 101402 SUB3  
6325 020566 102001 BVC SUB3  
6326 020570 100004 BPL SUB4  
6327 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6328 ; CONDITIONAL BRANCH INST. AND <====  
6329 ; REPLACE THE MOVE INSTRUCTION <====  
6330 ; WHICH FOLLOWS W/ 752 <====  
6331 020572 SUB3: MOV #477,-(R2) ;MOVE TO MAILBOX # ***** 477 *****  
6332 020576 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6333 020580 000000 HALT ;R0=177777  
6334 020600 000000 SUB4: COM R0 ;CC=11111  
6335 020602 005100 SCC  
6336 020604 000277  
6337 020606 162700 SUB #100000,R0 ;CC=0000 R0=77777  
6338 020612 101402 BLOS SUB5  
6339 020614 102401 BVS SUB5  
6340 020616 100004 BPL SUB6  
6341 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6342 ; CONDITIONAL BRANCH INST. AND <====  
6343 ;
```

```
6344 ;  
6345 ; REPLACE THE MOVE INSTRUCTION <====  
6346 ; WHICH FOLLOWS W/ 737 <====  
6347 020620 SUB5: MOV #500,-(R2) ;MOVE TO MAILBOX # ***** 500 *****  
6348 020624 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6349 020626 000000 HALT ;SUB DID NOT SET CC'S CORRECTLY  
6350 020630 000257 SUB6: CCC ;CC=0100  
6351 020632 000264 SEZ  
6352 020634 162700 SUB #140000,R0 ;CC=1011  
6353 020640 102003 BVC SUB7  
6354 020642 103403 BCC SUB7  
6355 020644 001401 BIC SUB7  
6356 020646 100404 BHI TST223  
6357 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6358 ; CONDITIONAL BRANCH INST. AND <====  
6359 ; REPLACE THE MOVE INSTRUCTION <====  
6360 ; WHICH FOLLOWS W/ 723 <====  
6361 020650 SUB7: MOV #501,-(R2) ;MOVE TO MAILBOX # ***** 501 *****  
6362 020650 012742 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6363 020654 005242 HALT  
6364 020656 000000  
6365 ;*****  
6366 ; TEST 223 TEST SBC INSTRUCTION  
6367 ;*****  
6368 ;  
6369 TST223: INC (R2) ;UPDATE TEST NUMBER  
6370 020662 022715 CMP #223,(R2) ;SEQUENCE ERROR?  
6371 020666 001053 BNE TST224-10 ;BR TO ERROR HALT ON SEQ ERROR  
6372 020670 012700 MOV #1,R0  
6373 020674 000277 SCC ;CC=1011  
6374 020676 000244 CLZ  
6375 020700 005600 SBC R0 ;CC=0100 R=0  
6376 020702 103403 BCS SBC1  
6377 020704 102402 BCC SBC1  
6378 020706 100401 BMI SRC1  
6379 020710 001404 BEQ SBC2  
6380 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6381 ; CONDITIONAL BRANCH INST. AND <====  
6382 ; REPLACE THE MOVE INSTRUCTION <====  
6383 ; WHICH FOLLOWS W/ 767 <====  
6384 020712 SBC1: MOV #502,-(R2) ;MOVE TO MAILBOX # ***** 502 *****  
6385 020716 012742 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6386 020720 005242 HALT ;SBC DID NOT SET CC'S CORRECTLY  
6387 020722 000000 SBC2: SCC ;CC=1010  
6388 020724 000277 +GLZICLC  
6389 020726 000245 SBC R0 ;CC=0100 R=0  
6390 020730 103403 BCS SBC3  
6391 020732 102402 BVS SBC3  
6392 020734 100401 BMI SBC3  
6393 020736 001404 BEQ SBC4  
6394 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
6395 ; CONDITIONAL BRANCH INST. AND <====  
6396 ; REPLACE THE MOVE INSTRUCTION <====  
6397 ; WHICH FOLLOWS W/ 754 <====  
6398 020740 SBC3:  
6399
```



```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 140
CFKAAC-P11 18-OCT-78 11:01 T223 TEST SBC INSTRUCTION SEQ 0152

6400 020740 012742 000503      MOV      #503,-(R2)      ;MOVE TO MAILBOX # ***** 503 *****
6401 020744 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6402 020746 000000              HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6403 020750 000277      SBC4:   SCC                    ;CC=0111
6404 020752 000250              CLN                    ;
6405 020754 005600              SBC      R0              ;CC=1001 R0=177777
6406 020756 103003              BCC      SBC5
6407 020760 102402              BVS      SBC5
6408 020762 001401              BEQ      SBC5
6409 020764 100404              BMI      SBC6
6410
6411
6412
6413
6414
6415 020766 012742 000504      MOV      #504,-(R2)      ;MOVE TO MAILBOX # ***** 504 *****
6416 020772 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6417 020774 000000              HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6418 020776 042700      SBC6:   BIC      #77777,R0 ;RO=10000
6419 021002 000277              SCC                    ;CC=1101
6420 021004 000242              CLV
6421 021006 005600              SBC      R0              ;CC=0010
6422 021010 101402              BLOS
6423 021012 102001              BVC      SBC7
6424 021014 100004              BPL      TST224
6425
6426
6427
6428
6429 021016 012742 000505      MOV      #505,-(R2)      ;MOVE TO MAILBOX # ***** 505 *****
6430 021018 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6431 021020 000000              HALT                    ;SBC DID NOT SET CC'S CORRECTLY
6432 021024 000000
6433
6434

```

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 141
CFKAAC-P11 18-OCT-78 11:01 T223 TEST SBC INSTRUCTION SEQ 0153

6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448 021026 005212
6449 021030 022712 000224
6450 021034 001053
6451 021036 012700 144000
6452 021042 000257
6453 021044 000266
6454 021046 006100
6455 021050 103003
6456 021052 102402
6457 021054 001401
6458 021056 100404
6459
6460
6461
6462
6463 021060
6464 021064 012742 000506      MOV      #506,-(R2)      ;MOVE TO MAILBOX # ***** 506 *****
6465 021066 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6466 021068 000000              HALT                    ;
6467 021070 000277      ROL1:   SCC                    ;CC=1100
6468 021072 000243              +CLV!CLC
6469 021074 006100              ROL      R0              ;CC=0011 R0=020000
6470 021076 103003              BCC      ROL3
6471 021100 102002              BVC      ROL3
6472 021102 001401              BEQ      ROL3
6473 021104 100004              BPL      ROL4
6474
6475
6476
6477
6478 021106
6479 021108 012742 000507      MOV      #507,-(R2)      ;MOVE TO MAILBOX # ***** 507 *****
6480 021112 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6481 021114 000000              HALT                    ;ROL DID NOT SET CC'S CORRECTLY
6482 021116 000277      ROL4:   SCC                    ;CC=0111
6483 021120 000250              CLN                    ;
6484 021122 006100              ROL      R0              ;CC=0000 R0=040001
6485 021124 101402              BLOS
6486 021126 102401              BVS      ROL5
6487 021130 100004              BPL      ROL6
6488
6489
6490

```

```

6491
6492 021132
6493 021132 012742 000510 ROL5: MOV #510,-(R2) ;MOVE TO MAILBOX # ***** 510 *****
6494 021136 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6495 021142 000000 ;HALT ;ROL DID NOT SET CC'S CORRECTLY
6496 021142 000257 ROL6: CCC ;CC=101
6497 021144 000265 +SEZISEC ;CC=1010 R0=100003
6498 021146 006100 ROL R0
6499 021150 101405 BLOS ROL7
6500 021156 107403 BVC ROL7
6501 021154 100003 BPL ROL7
6502 021156 022700 100003 CMP #100003,R0
6503 021162 001404 BEQ TST225
6504
6505 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6506 ; CONDITIONAL BRANCH INST. AND <====
6507 ; REPLACE THE MOVE INSTRUCTION <====
6508 ; WHICH FOLLOWS W/ 742 <====
6509
6510 021164 012742 000511 ROL7: MOV #511,-(R2) ;MOVE TO MAILBOX # ***** 511 *****
6511 021170 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6512 021172 000000 ;HALT ;ROL MalfUNCTIONED
6513 ; OR SEQUENCE ERROR
6514
6515 *****
6516 ;TEST 225 TEST ROR INSTRUCTION *****
6517 ;TST225: INC (R2) ;UPDATE TEST NUMBER
6518 ;CMP #225,(R2) ;SEQUENCE ERROR?
6519 ;BNE TST226-10 ;BR TO ERROR HALT ON SEQ ERROR
6520 ;MOV #23,R0 ;R0=23
6521 ;SCC ;CC=0111
6522 ;CLN ;CC=1001 R0=100011
6523 ;ROR R0
6524 ;ROR1
6525 ;ROR1
6526 ;ROR1
6527 ;BMI ROR2
6528
6529 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6530 ; CONDITIONAL BRANCH INST. AND <====
6531 ; REPLACE THE MOVE INSTRUCTION <====
6532 ; WHICH FOLLOWS W/ 767 <====
6533
6534 021226 012742 000512 ROR1: MOV #512,-(R2) ;MOVE TO MAILBOX # ***** 512 *****
6535 021232 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6536 021234 000000 ;HALT ;ROL DID NOT SET CC'S CORRECTLY
6537 021240 000257 ROR2: CCC ;CC=1100
6538 021242 000274 +SENISEZ ;CC=0011 R0=040004
6539 021242 006000 ROR R0
6540 021244 102003 BVC ROR3
6541 021246 103002 BCC ROR3
6542 021250 101401 BEQ ROR3
6543 021252 100004 BPL ROR4
6544
6545 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6546 ; CONDITIONAL BRANCH INST. AND <====
6547 ; REPLACE THE MOVE INSTRUCTION <====
6548 ; WHICH FOLLOWS W/ 754 <====
    
```

```

6547 021254 012742 000513 ROR4: MOV #513,-(R2) ;MOVE TO MAILBOX # ***** 513 *****
6548 021260 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6549 021262 000000 ;HALT ;ROL DID NOT SET CC'S CORRECTLY
6550 021264 000277 ROR4: CCC ;CC=1110
6551 021266 000241 CLC ;CC=0000 R0=020002
6552 021270 006000 ROR R0
6553 021274 101403 BLOS ROR5
6554 021274 102402 BVS ROR5
6555 021276 001401 BEQ ROR5
6556 021300 100004 BPL ROR6
6557
6558 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6559 ; CONDITIONAL BRANCH INST. AND <====
6560 ; REPLACE THE MOVE INSTRUCTION <====
6561 ; WHICH FOLLOWS W/ 741 <====
6562
6563 021302 012742 000514 ROR5: MOV #514,-(R2) ;MOVE TO MAILBOX # ***** 514 *****
6564 021306 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6565 021310 000000 ;HALT ;ROL DID NOT SET CC'S CORRECTLY
6566 021312 000257 ROR6: CCC ;CC=0101
6567 021314 000265 +SEZISEZ ;CC=1010 R0=110001
6568 021316 006000 ROR R0
6569 021320 101402 BLOS ROR7
6570 021324 102001 BVC ROR7
6571 021324 100404 BMI TST226
6572
6573 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6574 ; CONDITIONAL BRANCH INST. AND <====
6575 ; REPLACE THE MOVE INSTRUCTION <====
6576 ; WHICH FOLLOWS W/ 727 <====
6577
6578 021326 012742 000515 ROR7: MOV #515,-(R2) ;MOVE TO MAILBOX # ***** 515 *****
6579 021332 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6580 021334 000000 ;HALT ;ROL DID NOT PRODUCE CORRECT RESULTS
6581 ; OR SEQUENCE ERROR
6582
6583 *****
6584 ;TEST 226 TEST ASL INSTRUCTION *****
6585 ;TST226: INC (R2) ;UPDATE TEST NUMBER
6586 ;CMP #226,(R2) ;SEQUENCE ERROR?
6587 ;BNE TST227-10 ;BR TO ERROR HALT ON SEQ ERROR
6588 ;MOV #144000,R0 ;R0=144000
6589 ;CCC ;CC=0110
6590 ;+SENISEZ ;CC=1001 R0=110000
6591 ;ASL R0
6592 ;BCC ASL1
6593 ;BVS ASL1
6594 ;BEQ ASL1
6595 ;BMI ASL2
6596
6597 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6598 ; CONDITIONAL BRANCH INST. AND <====
6599 ; REPLACE THE MOVE INSTRUCTION <====
6600 ; WHICH FOLLOWS W/ 767 <====
6601
6602 021370 012742 000516 ASL1: MOV #516,-(R2) ;MOVE TO MAILBOX # ***** 516 *****
6603 021374 005242 ;INC ;SET MSGTYP TO FATAL ERROR
6604 021376 000000 ;HALT ;ROL DID NOT SET CC'S CORRECTLY
6605 021400 000277 ASL2: SCC ;CC=1100
    
```

```

6603 021402 000243 +CLV1CLC
6604 021404 006300 ASL R0 ;CC=0011 R0=020000
6605 021406 103003 BCC ASL3
6606 021410 102002 BVC ASL3
6607 021412 001401 BEQ ASL3
6608 021414 100004 BPL ASL4
6609
6610 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6611 ; CONDITIONAL BRANCH INST. AND <====
6612 ; REPLACE THE MOVE INSTRUCTION <====
6613 ; WHICH FOLLOWS W/ 754 <====
6614 021416 000517 ASL3: MOV #517,-(R2) ;MOVE TO MAILBOX # ***** 517 *****
6615 021422 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6616 021424 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6617 021426 000277 ASL4: SCC ;CC=0111
6618 021430 000250 CLN
6619 021432 006300 ASL R0 ;CC=0000 R0=040000
6620 021434 101402 BLOS ASL5
6621 021436 102401 BVS ASL5
6622 021440 100004 BPL ASL6
6623
6624 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6625 ; CONDITIONAL BRANCH INST. AND <====
6626 ; REPLACE THE MOVE INSTRUCTION <====
6627 ; WHICH FOLLOWS W/ 742 <====
6628 021442 000520 ASL5: MOV #520,-(R2) ;MOVE TO MAILBOX # ***** 520 *****
6629 021446 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6630 021450 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6631 021452 000257 ASL6: CFC ;CC=0101
6632 021454 000255 +SEZ1SEC
6633 021456 006300 ASL R0 ;CC=1010 R0=100000
6634 021460 103406 BCS ASL7
6635 021462 001405 BEQ ASL7
6636 021464 102004 BVC ASL7
6637 021466 100003 BPL ASL7
6638 021470 022700 CMP #100000,R0
6639 021474 001404 BEQ TST227
6640
6641 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6642 ; CONDITIONAL BRANCH INST. AND <====
6643 ; REPLACE THE MOVE INSTRUCTION <====
6644 ; WHICH FOLLOWS W/ 724 <====
6645 021476 000521 ASL7: MOV #521,-(R2) ;MOVE TO MAILBOX # ***** 521 *****
6646 021502 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6647 021504 000000 HALT ;ASL MALFUNCTIONED
6648 ; OR SEQUENCE ERROR
    
```

```

*****
;TEST 227 TEST ASR INSTRUCTION
*****
TST227: INC (R2) ;UPDATE TEST NUMBER
        CMP #227,(R2) ;SEQUENCE ERROR?
        BNE #52750-10 ;R TO ERROR HALT ON SEQ ERROR
        MOV #100023,R0 ;R0=100023
        SCC ;CC=0110
        CLN
        ASR R0 ;CC=1001 RP=140011
        BVS ASR1
        BVC ASR1
        BEQ ASR1
        BMI ASR2
6663
6664 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6665 ; CONDITIONAL BRANCH INST. AND <====
6666 ; REPLACE THE MOVE INSTRUCTION <====
6667 ; WHICH FOLLOWS W/ 767 <====
6668 021540 000522 ASR1: MOV #522,-(R2) ;MOVE TO MAILBOX # ***** 522 *****
6669 021544 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6670 021546 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
6671 021550 100000 ASR2: BIC #100000,R0 ;R0=40011
6672 021554 000277 SCC ;CC=1100
6673 021556 000243 +CLV1CLC
6674 021560 006200 ASR R0 ;CC=0011 R0=020004
6675 021562 103003 BVC ASR3
6676 021564 103002 BCC ASR3
6677 021566 001401 BEQ ASR3
6678 021570 100004 BPL ASR4
6679
6680 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6681 ; CONDITIONAL BRANCH INST. AND <====
6682 ; REPLACE THE MOVE INSTRUCTION <====
6683 ; WHICH FOLLOWS W/ 752 <====
6684 021572 000523 ASR3: MOV #523,-(R2) ;MOVE TO MAILBOX # ***** 523 *****
6685 021576 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6686 021600 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
6687 021602 000277 ASR4: SCC ;CC=1111
6688
6689 ASR R0 ;CC=0000 R0=010002
6690 021606 101403 BLOS ASR5
6691 021610 102402 BVS ASR5
6692 021612 001401 BEQ ASR5
6693 021614 100004 BPL ASR6
6694
6695 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6696 ; CONDITIONAL BRANCH INST. AND <====
6697 ; REPLACE THE MOVE INSTRUCTION <====
6698 ; WHICH FOLLOWS W/ 740 <====
6699 021616 000524 ASR5: MOV #524,-(R2) ;MOVE TO MAILBOX # ***** 524 *****
6700 021622 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6701 021624 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
6702 021626 052700 ASR6: BLS #100000,R0 ;R0=110002
6703 021632 000257 CFC ;CC=0101
6704 021634 000265 +SEZ1SEC
    
```

```

CFKAAC.P11 18-OCT-78 11:01
6705 021636 006200 ASR R0 ;C=1010 R0=144001
6706 021640 101406 BLOS ASR7
6707 021642 102005 BVC ASR7
6708 021644 100004 BPL ASR7
6709 021646 001403 BEQ ASR7
6710 021650 022700 144001 CMP #144001,R0 ;CHECK RESULT OF ASR'S
6711 021654 001404 BEQ TST230
6712
6713 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6714 ; CONDITIONAL BRANCH INST. AND <====
6715 ; REPLACE THE MOVE INSTRUCTION <====
6716 ; WHICH FOLLOWS W/ 720 <====
6717
6718 ASR7: MOV #525,-(R2) ;MOVE TO MAILBOX # ***** 525 *****
6719 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6720 HALT ;ASR DID NOT FUNCTION CORRECTLY
6721 ; OR SEQUENCE ERROR
6722
6723 ;*****
6724 ; THIS TEST VERIFIES THE SXT INSTRUCTION- CONDITION CODES
6725 ; ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
6726 ; THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
6727 ; CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
6728 ; IS VERIFIED BY CONDITIONAL BRANCHES.
6729
6730 ;*****
6731 ; TEST 230 TEST THE SXT INSTRUCTION
6732 ;*****
6733 TST230: INC (R2) ;UPDATE TEST NUMBER
6734 CMP #230,(R2) ;SEQUENCE ERROR?
6735 BNE TST231-10 ;BR TO ERROR HALT ON SEQ ERROR
6736 CLR R0
6737 SCC ;SET CC=1011
6738 CLZ
6739 SXT R0 ;TRY SXT
6740 BPL SXT0 ;TEST CC=1001
6741 BEQ SXT0
6742 BCC SXT0
6743 CMP #-1,R0 ;CHECK DATA RESULT
6744 BEQ SXT1
6745
6746 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6747 ; CONDITIONAL BRANCH INST. AND <====
6748 ; REPLACE THE MOVE INSTRUCTION <====
6749 ; WHICH FOLLOWS W/ 765 <====
6750
6751 SXT0: MOV #526,-(R2) ;MOVE TO MAILBOX # ***** 526 *****
6752 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6753 HALT ;RESULTS OF SXT INCORRECT
6754 CLR R0 ;R0=0
6755 CLR (R0) ;LOC- 0=0
6756 COM (R0) ;LOC- 0=177777
6757 CCC ;SET CC=0110
6758 +SEZ1SEV
6759
6760

```

```

CFKAAC.P11 18-OCT-78 11:01
6761 021746 006710 SXT (R0)
6762 021750 001005 BNE SXT2 ;TEST CC=0100
6763 021752 103404 BCS SXT2
6764 021754 102403 BVS SXT2
6765 021756 100402 BMI SXT2
6766 021760 005110 TST (R0)
6767 021762 001404 BEQ TST231
6768
6769 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6770 ; CONDITIONAL BRANCH INST. AND <====
6771 ; REPLACE THE MOVE INSTRUCTION <====
6772 ; WHICH FOLLOWS W/ 745 <====
6773
6774 SXT2: MOV #527,-(R2) ;MOVE TO MAILBOX # ***** 527 *****
6775 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6776 HALT ;RESULTS OF SXT INCORRECT
6777 ; OR SEQUENCE ERROR

```

```
*****
;
; THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
; OF ONES AND ZEROS ARE MOVED TO DATA REGISTERS R0 AND R1.
; AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
; EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
; REPRODUCE THE ORIGINAL VALUE IF R0=31525.
;
;*****
TEST 231 TEST THE XOR INSTRUCTION
;*****
TST231: INC (R2) ;UPDATE TEST NUMBER
; INC #231,(R2) ;SEQUENCE ERROR?
; CMP #231,(R2) ;BR TO ERROR HALT ON SEQ ERROR
; BNE TST232-10
; MOV #7463,R0 ;SET UP R0
; MOV #31525,R1 ;SET UP R1
; SCC ;SET CC=1110
; CLC
; XOR R1,R0 ;TRY XOR
; BLOS XOR1 ;CC=0000?
; BVS XOR1
; BEQ XOR1
; BMI XOR1
; BMI #36146,R0 ;DATA RESULT CORRECT?
; CMP XOR2
; BEQ
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
;
XOR1: MOV #530,-(R2) ;MOVE TO MAILBOX # ***** 530 *****
; HALT -(R2) ;SET MSGTYP TO FATAL ERROR
;
XOR2: MOV R1,R4
; SEC
; CLC
; XOR R4,R0 ;TRY XOR MODE 0,0
; BLOS XOR3 ;CC=0000?
; BVS XOR3
; BEQ XOR3
; BMI XOR3
; CMP #7463,R0
; BEQ TST232
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 743 <====
;
XOR3: MOV #531,-(R2) ;MOVE TO MAILBOX # ***** 531 *****
; INC -(R2) ;SET MSGTYP TO FATAL ERROR
; HALT ;RESULT OF XOR INCORRECT
; OR SEQUENCE ERROR
```

```
*****
;
; THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
; COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
; BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
; WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
;
;*****
TEST 232 TEST SOB INSTRUCTION
;*****
TST232: INC (R2) ;UPDATE TEST NUMBER
; CMP #232,(R2) ;SEQUENCE ERROR?
; BNE TST233-10 ;BR TO ERROR HALT ON SEQ ERROR
; MOV #525,R0
; MOV R0,R4
; SCC ;SET CC=1111
; BHI SOB2 ;CC=1111?
; BPL SOB2
; BVS SOB3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
;
SOB2: MOV #532,-(R2) ;MOVE TO MAILBOX # ***** 532 *****
; INC -(R2) ;SET MSGTYP TO FATAL ERROR
;
SOB3: DEC R4 ;COUNT ITERATIONS
; SCC ;CC=1111
; SOB R0,SOB1 ;DO SOB W/ R0
; BHI SOB4 ;CHECK CC=1111
; BPL SOB4
; BVC SOB4
; TST R4 ;ITERATION COUNT OK?
; BEQ TST233
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
;
SOB4: MOV #533,-(R2) ;MOVE TO MAILBOX # ***** 533 *****
; INC -(R2) ;SET MSGTYP TO FATAL ERROR
; HALT ;INCORRECT # OF BRANCHES OR CC'S CHANGED
; OR SEQUENCE ERROR
```

```

6873 ;*****
6874 ;
6875 ; THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
6876 ; OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
6877 ; THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
6878 ; OF THE TWO ROUTINES IN THE TEST.
6879 ;*****
6880 ;
6881 ; TEST 233 TEST MARK INSTRUCTION
6882 ;*****
6883 022174 005212 TST233: INC (R2) ;UPDATE TEST NUMBER
6884 022176 022712 CMP #233,(R2) ;SEQUENCE ERROR?
6885 022202 001062 BNE TST234-10 ;BR TO ERROR HALT ON SEQ ERROR
6886 022210 012746 MOV #125252,SP ;PUT R5 VALUE ON STACK
6887 022214 162706 SUB #74,SP ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
6888 022220 012705 MOV #MRK1,R5 ;SET NEW PC IN R5
6889 022224 012746 MOV #6436,-(SP) ;PUT MARK 36 INST. ON STACK
6890 022230 000277 SCC ;SET CC=1111
6891 022232 000137 JMP #400 ;XFER CONTRL TO MARK 36 INST. ON STACK
6892 022236 012742 MOV #534,-(R2) ;MOVE TO MAILBOX # ***** 534 *****
6893 022242 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6894 022244 000000 HALT ;MARK INST. SHOULD HAVE JUMPED TO MRK1
6895 022246 101010 BHI MRK2 ;TEST CC UNAFFECTED
6896 022250 100007 BPL MRK2 ;IE. CC=1111
6897 022252 102009 BVC MRK2
6898 022254 020529 CMP #R5,#125252 ;CHECK R5 RESTORED FROM STACK
6899 022260 001003 BNE MRK2
6900 022262 022706 CMP #STBOT,R6 ;CHECK STACK POINTER READJUSTED CORRECTLY.
6901 022266 001404 BEQ MRK3
6902 ;
6903 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6904 ; CONDITIONAL BRANCH INST. AND <====
6905 ; REPLACE THE MOVE INSTRUCTION <====
6906 ; WHICH FOLLOWS W/ 746 <====
6907 022270 MRK2: MOV #535,-(R2) ;MOVE TO MAILBOX # ***** 535 *****
6908 022274 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6909 022276 000000 HALT ;RESULTS OF MARK INCORRECT
6910 022300 MRK3: MOV #52525,-(SP) ;PUT MARK 0 INST. ON STACK
6911 022304 012746 MOV #6400,-(SP) ;SET ADDR. OF MARK INST. IN R5
6912 022310 010605 MOV SP,R5 ;DO JSR
6913 022312 004737 JSR PC,MRK4
6914 022316 000137 JMP #MRK5
6915 022322 MRK4: RTS R5 ;DO RTS WITH R5 TO MARK INST ON STACK
6916 022324 012742 MOV #536,-(R2) ;MOVE TO MAILBOX # ***** 536 *****
6917 022330 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6918 022332 000000 HALT ;RTS MARK SEQUENCE FAILED
6919 022334 000500 MRK5: CMP #STBOT,R6 ;STACK ADJUSTED CORRECTLY
6920 022340 001003 BNE MRK6 ;IF NOT: BR
6921 022342 022705 CMP #52525,R5 ;CHECK IF R5 RESTORED FROM STACK
6922 022346 001404 BEQ TST234
6923 ;
6924 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6925 ; CONDITIONAL BRANCH INST. AND <====
6926 ; REPLACE THE MOVE INSTRUCTION <====
6927 ; WHICH FOLLOWS W/ 716 <====
6928 022350 MRK6:

```

```

6929 022350 012742 000537 MOV #537,-(R2) ;MOVE TO MAILBOX # ***** 537 *****
6930 022354 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6931 022356 000000 HALT ;RESULTS OF MARK INCORRECT
6932 ; OR SEQUENCE ERROR

```

```
6933 177776 PS=177776
6934 *****
6935 THESE NEXT SEVEN TESTS VERIFY THE MTPS INSTRUCTION IN ALL
6936 MODES. THE PSW IS DEFINED BY AN EQUATE STATEMENT BEFORE THE
6937 FIRST MTPS TEST. IN EACH TEST A PATTERN OF ONES AND
6938 ZEROS IS SET IN A DATA REGISTER AND MOVED TO THE PSW.
6939 THE DATA IN THE PSW, AND THE DATA REGISTER ADDRESS,
6940 ARE CHECKED TO VERIFY PROPER EXECUTION OF THE INSTRUCTION.
6941 *****
6942 ;TEST 234 TEST MTPS INSTRUCTION
6943 *****
6944 ;TST234: INC (R2) ;UPDATE TEST NUMBER
6945 CMP #234,(R2) ;SEQUENCE ERROR?
6946 BNE #T235-10 ;BR TO ERROR HALT ON SEQ ERROR
6947 MOV #377,R0
6948 CCC
6949 MTPS R0
6950 CMP #357,PS
6951 BEQ MTPS1
6952 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
6953 ; CONDITIONAL BRANCH INST. AND <===
6954 ; REPLACE THE MOVE INSTRUCTION <===
6955 ; WHICH FOLLOWS W/ 770 <===
6956 *****
6957 ;TST234: MOV #540,-(R2) ;MOVE TO MAILBOX # ***** 540 *****
6958 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6959 HALT ;MTPS FAILED
6960 MTPS1: CLR R0 ;CC-1111
6961 CLR (R0) ;TRY MTPS MODE 1
6962 SCC ;CHECK PS
6963 MTPS (R0)
6964 BMS MTPS1A
6965 BVS MTPS1A
6966 BCS MTPS1A
6967 BNE TST235
6968 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
6969 ; CONDITIONAL BRANCH INST. AND <===
6970 ; REPLACE THE MOVE INSTRUCTION <===
6971 ; WHICH FOLLOWS W/ 754 <===
6972 *****
6973 ;TST234: MOV #541,-(R2) ;MOVE TO MAILBOX # ***** 541 *****
6974 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6975 HALT ;MTPS FAILED
6976 ; OR SEQUENCE ERROR
6977 *****
6978 ;TEST 235 TEST MTPS MODE 2
6979 *****
6980 ;TST235: INC (R2) ;UPDATE TEST NUMBER
6981 CMP #235,(R2) ;SEQUENCE ERROR?
6982 BNE #T236-10 ;BR TO ERROR HALT ON SEQ ERROR
6983 CLR R0 ;R0=0
6984 MOV #-1,(R0) ;R0=-1
6985 CLR #PS ;PS=0
6986 MTPS (R0)+ ;TRY MTPS W/MODE 2
6987 *****
6988
```

```
6989 022474 022737 000357 177776 CMP #357,@#PS ;CHECK DATA
6990 022502 001404 BEQ MTPS2
6991 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
6992 ; CONDITIONAL BRANCH INST. AND <===
6993 ; REPLACE THE MOVE INSTRUCTION <===
6994 ; WHICH FOLLOWS W/ 766 <===
6995 *****
6996 ;TST235: MOV #542,-(R2) ;MOVE TO MAILBOX # ***** 542 *****
6997 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6998 HALT ;DEST. DATA INCORRECT
6999 MTPS2: CMP #1,R0 ;CHECK DEST. REGISTER.
7000 BEQ TST236
7001 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7002 ; CONDITIONAL BRANCH INST. AND <===
7003 ; REPLACE THE MOVE INSTRUCTION <===
7004 ; WHICH FOLLOWS W/ 763 <===
7005 *****
7006 ;TST235: MOV #543,-(R2) ;MOVE TO MAILBOX # ***** 543 *****
7007 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7008 HALT ;DEST REGISTER NOT INCREMENTED BY 1
7009 ; OR SEQUENCE ERROR
7010 *****
7011 ;TEST 236 TEST MTPS MODE 3
7012 *****
7013 ;TST236: INC (R2) ;UPDATE TEST NUMBER
7014 CMP #236,(R2) ;SEQUENCE ERROR?
7015 BNE #T237-10 ;BR TO ERROR HALT ON SEQ ERROR
7016 MOV #402,R0 ;R0=402
7017 CLR (R0) ;LOC. 402=0
7018 MOV #52652,@#0 ;LOC. 0=52652
7019 CLR @#PS ;PS=0
7020 MTPS (R0)+ ;TRY MTPS W/MODE 3
7021 BEQ MTPS3 ;CHECK DEST. DATA
7022 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7023 ; CONDITIONAL BRANCH INST. AND <===
7024 ; REPLACE THE MOVE INSTRUCTION <===
7025 ; WHICH FOLLOWS W/ 763 <===
7026 *****
7027 ;TST236: MOV #544,-(R2) ;MOVE TO MAILBOX # ***** 544 *****
7028 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7029 HALT ;DEST. DATA INCORRECT
7030 MTPS3: CMP #404,R0 ;CHECK MODE 3 REGISTER.
7031 BEQ TST237
7032 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7033 ; CONDITIONAL BRANCH INST. AND <===
7034 ; REPLACE THE MOVE INSTRUCTION <===
7035 ; WHICH FOLLOWS W/ 754 <===
7036 *****
7037 ;TST236: MOV #545,-(R2) ;MOVE TO MAILBOX # ***** 545 *****
7038 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7039 HALT ;MODE 3 REGISTER INCORRECT
7040 ; OR SEQUENCE ERROR
7041 *****
7042 ;TEST 237 TEST MTPS MODE 4
7043 *****
7044 ;TST237: INC (R2) ;UPDATE TEST NUMBER
7045 CMP #237,(R2) ;SEQUENCE ERROR?
7046 *****
```

7045 022630 001022 BNE TST240-10 ;BR TO ERROR HALT ON SEQ ERROR
7046 022632 012700 000001 MOV #1,R0 ;R0=1
7047 022636 012737 000000 MOV #125125,@#0 ;LOC 0 = 125125
7048 022644 005037 177776 CLR @#PS ;PS=0
7049 022652 022737 000105 MTPS -R0 ;TRY MTPS W/MODE 4
7050 022652 022737 000105 CMP #105,@#PS ;CHECK DEST. DATA
7051 022660 001404 BEQ MTPS4 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
7056 022662 012742 000546 MOV #546,-(R2) ;MOVE TO MAILBOX # ***** 546 *****
7057 022666 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7058 022670 000000 HALT ;DEST. DATA INCORRECT
7059 022672 005700 MTPS4: R0 ;CHECK MODE 4 REGISTER
7060 022674 001404 BEQ TST240 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
7065 022676 012742 000547 MOV #547,-(R2) ;MOVE TO MAILBOX # ***** 547 *****
7066 022704 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7067 022704 000000 HALT ;MODE 4 REGISTER NOT DECREMENTED BY 1
7068 ; OR SEQUENCE ERROR
; *****
;TEST 240 TEST MTPS MODE 5
;*****
TST240: INC (R2) ;UPDATE TEST NUMBER
CMP #240,(R2) ;SEQUENCE ERROR?
BNE TST241-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #404,R0 ;R0=404
MTPS #177400,@#0 ;LOC 0=177400
SCC ;SET ALL COND. CODES
MTPS @-(R0) ;TRY MTPS W/MODE 5
TST @#PS ;CHECK DEST. DATA.
BEQ MTPS5 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
7086 022742 012742 000550 MOV #550,-(R2) ;MOVE TO MAILBOX # ***** 550 *****
7087 022746 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7088 022750 000000 HALT ;DESTINATION DATA INCORRECT
7089 022752 002700 MTPS5: R0 ;CHECK MODE 5 REGISTER
7090 022756 001404 BEQ TST241 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
7095 022760 012742 000551 MOV #551,-(R2) ;MOVE TO MAILBOX # ***** 551 *****
7096 022764 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7097 022766 000000 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
7098 ; OR SEQUENCE ERROR
; *****

7101 ;TEST 241 TEST MTPS MODE 6
7102 ;*****
7103 022770 005212 TST241: INC (R2) ;UPDATE TEST NUMBER
7104 022772 022712 000241 CMP #241,(R2) ;SEQUENCE ERROR?
7105 022776 001037 000000 BNE TST242-10 ;BR TO ERROR HALT ON SEQ ERROR
7106 023000 012700 000406 MOV #406,R0 ;LOC 0=52652
7107 023006 012700 000406 MTPS #406,R0 ;R0=406
7108 023012 005037 177776 CLR @#PS ;PS=0
7109 023016 106460 177372 MTPS -406(R0) ;TRY MTPS W/MODE 6
7110 023022 022737 000252 CMP #252,@#PS ;CHECK DEST. DATA
7111 023030 001404 BEQ MTPS6 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 763 <====
7116 023032 012742 000552 MOV #552,-(R2) ;MOVE TO MAILBOX # ***** 552 *****
7117 023040 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7118 023040 000000 HALT ;DEST. DATA INCORRECT
7119 023042 022700 000406 MTPS6: R0 ;CHECK MODE 6 REGISTER
7120 023046 001404 BEQ TST242 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
7125 023050 012742 000553 MOV #553,-(R2) ;MOVE TO MAILBOX # ***** 553 *****
7126 023054 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7127 023056 000000 HALT ;MODE 6 REGISTER MODIFIED
7128 ; OR SEQUENCE ERROR
; *****
;TEST 242 TEST MTPS MODE 7
;*****
7132 023060 005212 TST242: INC (R2) ;UPDATE TEST NUMBER
7133 023062 022712 000242 CMP #242,(R2) ;SEQUENCE ERROR?
7134 023066 001024 000000 BNE TST243-10 ;BR TO ERROR HALT ON SEQ ERROR
7135 023070 012737 052652 000000 MOV #52652,@#0 ;LOC 0=52652
7136 023076 012700 000410 MTPS #410,R0 ;R0=410
7137 023076 012700 000410 MTPS @-(R0) ;TRY MTPS W/MODE 7
7138 023102 005037 177776 CLR @#PS ;CHECK DEST. DATA
7139 023106 106470 000105 MTPS #105,@#PS ;
7140 023112 022737 000105 CMP #105,@#PS ;
7141 023120 001404 BEQ MTPS7 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 763 <====
7146 023122 012742 000554 MOV #554,-(R2) ;MOVE TO MAILBOX # ***** 554 *****
7147 023126 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7148 023130 000000 HALT ;DESTINATION DATA INCORRECT
7149 023132 022700 000410 MTPS7: R0 ;CHECK MODE 7 REGISTER
7150 023136 001404 BEQ TST243 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
7155 023140 012742 000555 MOV #555,-(R2) ;MOVE TO MAILBOX # ***** 555 *****
7156 023144 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR

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

```

;*****
;
; THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
; MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROS IS MOVED TO THE
; PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
; BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
; USED TO CHECK PROPER ADDRESSING AND DATA.
;*****
;TEST 243 TEST MFPs INSTRUCTION
;*****
TST243: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
;BR TO ERROR HALT ON SEQ ERROR
CMP #243,(R2)
BNE #T244-10
MOV #377,@#PS
MFPS R0
CMP #177757,R0
BEQ MFPS1

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV #556,-(R2) ;MOVE TO MAILBOX # ***** 556 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MFPS FAILED

MFPS1: CLR R0
MOV #-1,@#0
CLR @#PS
MFPS (R0)
TSTB @#0
BEQ TST244

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 753 <====
MOV #557,-(R2) ;MOVE TO MAILBOX # ***** 557 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MFPS FAILED
; OR SEQUENCE ERROR
;*****
;TEST 244 TEST MFPs MODE 2
;*****
TST244: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
;BR TO ERROR HALT ON SEQ ERROR
CMP #244,(R2)
BNE #T245-10
CLR R0
R0=0
CLR (R0)
;LOC=0=0
MOV #377,@#PS ;SET PS=357
MFPS (R0)+ ;TRY MFPS W/MODE 2
BCC MFPS2A ;BR TO ERROR IF C BIT CLEAR
BVS MFPS2A ;BR TO ERROR IF V BIT SET
BEQ MFPS2A ;BR TO ERROR IF Z BIT SET
BMI MFPS2B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

```

7216
7217
7218
7219 023276 012742 000560 MFPS2A: MOV #560,-(R2) ;MOVE TO MAILBOX # ***** 560 *****
7220 023276 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7221 023304 000000 ;HALT ;COND. CODES INCORRECT
7222 023306 022737 000357 000000 MFPS2B: CMP #357,@#0 ;CHECK DEST. DATA
7223 023314 001404 ;BEQ
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 756
7225
7226
7227
7228
7229 023316 012742 000561 MOV #561,-(R2) ;MOVE TO MAILBOX # ***** 561 *****
7230 023322 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7231 023324 000000 ;HALT ;DEST. DATA INCORRECT
7232 023326 022700 000001 MFPS2C: CMP #1,R0 ;CHECK MODE Z REGISTER
7233 023332 001404 ;BEQ TST245
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 747
7235
7236
7237
7238
7239 023334 012742 000562 MOV #562,-(R2) ;MOVE TO MAILBOX # ***** 562 *****
7240 023340 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7241 023342 000000 ;HALT ;MODE 2 REGISTER NOT INCREMENTED 1
7242
7243
7244
7245
;*****
;TEST 245 TEST MPPS MODE 3
;*****
TST245: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
;BR TO ERROR HALT ON SEQ ERROR
;R0=406
;LOC =0
;PS=252
;TRY MPPS WITH MODE 3
;BR TO ERROR IF C-BIT SET
;BR TO ERROR IF V-BIT SET
;BR TO ERROR IF Z-BIT SET
7246 023344 005212 000245 INC (R2) ;UPDATE TEST NUMBER
7247 023346 022712 000000 BNE TST246-10 ;SEQUENCE ERROR?
7248 023352 001033 000406 MOV #406,R0 ;BR TO ERROR HALT ON SEQ ERROR
7249 023354 012700 000000 CLR @#0 ;R0=406
7250 023360 005037 000000 MFPS @#PS ;LOC =0
7251 023364 012737 000252 MFPS #252,@#PS ;TRY MPPS WITH MODE 3
7252 023372 106730 ;BVS MPPS3A ;BR TO ERROR IF C-BIT SET
7253 023374 103403 ;BVS MPPS3A ;BR TO ERROR IF V-BIT SET
7254 023376 ;BEQ MPPS3A ;BR TO ERROR IF Z-BIT SET
7255 023400 001401 ;BMI MPPS3B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 764
7257
7258
7259
7260
7261 023404 012742 000563 MFPS3A: MOV #563,-(R2) ;MOVE TO MAILBOX # ***** 563 *****
7262 023410 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7263 023412 000000 ;HALT ;CONDITION CODES INCORRECT
7264 023414 022737 125000 000000 MFPS3B: CMP #125000,@#0 ;CHECK DEST. DATA
7265 023422 001404 ;BEQ MPPS3C
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 754
7267
7268
7269
7270 023424 012742 000564 MOV #564,-(R2) ;MOVE TO MAILBOX # ***** 564 *****
7271

```

```

7272 023430 005242 000410 MFPS3C: INC -(R2) ;SET MSGTYP TO FATAL ERROR
7273 023432 000000 ;HALT ;DEST. DATA INCORRECT
7274 023434 020027 000000 CMP R0,#410 ;CHECK MODE 3 REGISTER.
7275 023440 001404 ;BEQ TST246
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 745
7277
7278
7279
7280 023442 012742 000565 MOV #565,-(R2) ;MOVE TO MAILBOX # ***** 565 *****
7281 023446 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7282 023450 000000 ;HALT ;MODE 3 REGISTER NOT INCREMENTED BY 2
7283
7284
7285
7286
;*****
;TEST 246 TEST MPPS MODE 4
;*****
TST246: INC (R2) ;UPDATE TEST NUMBER
;SEQUENCE ERROR?
;BR TO ERROR HALT ON SEQ ERROR
;R0=2
;LOC =0
;PS=125
;TRY MPPS W/MODE 4
;BR TO ERROR IF C-BIT CLEAR
;BR TO ERROR IF V-BIT SET
;BR TO ERROR IF Z-BIT SET
7288 023452 005212 000246 INC (R2) ;UPDATE TEST NUMBER
7289 023454 022712 000000 BNE TST247-10 ;SEQUENCE ERROR?
7290 023460 001033 000002 MOV #2,R0 ;BR TO ERROR HALT ON SEQ ERROR
7291 023462 012700 000000 CLR @#0 ;R0=2
7292 023466 005037 000000 MFPS @#PS ;LOC =0
7293 023472 012737 000125 MFPS #125,@#PS ;TRY MPPS W/MODE 4
7294 023500 106740 ;BVS MPPS4A ;BR TO ERROR IF C-BIT CLEAR
7295 023502 103003 ;BVS MPPS4A ;BR TO ERROR IF V-BIT SET
7296 023504 102402 ;BEQ MPPS4A ;BR TO ERROR IF Z-BIT SET
7297 023506 001401 ;BPL MPPS4B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 764
7299
7300
7301
7302
7303 023512 012742 000566 MFPS4A: MOV #566,-(R2) ;MOVE TO MAILBOX # ***** 566 *****
7304 023516 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7305 023518 000000 ;HALT ;COND. CODES INCORRECT
7306 023520 022737 042400 000000 MFPS4B: CMP #42400,@#0 ;CHECK DEST. DATA
7307 023522 001404 ;BEQ MPPS4C
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 745
7309
7310
7311
7312
7313 023532 012742 000567 MOV #567,-(R2) ;MOVE TO MAILBOX # ***** 567 *****
7314 023536 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7315 023540 000000 ;HALT ;DEST. DATA INCORRECT
7316 023542 020027 000001 MFPS4C: CMP R0,#1 ;CHECK MODE 4 REGISTER
7317 023546 001404 ;BEQ TST247
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 745
7319
7320
7321
7322 023550 012742 000570 MOV #570,-(R2) ;MOVE TO MAILBOX # ***** 570 *****
7323 023552 005242 ;INC -(R2) ;SET MSGTYP TO FATAL ERROR
7324 023556 000000 ;HALT ;MODE 4 REGISTER NOT DECREMENTED BY 1
7325
7326
7327
;*****

```

```

7328 ;TEST 247 TEST MFPS MODE 5
7329 ;*****
7330 023560 005212 ;UPDATE TEST NUMBER
7331 023562 022712 ;SEQUENCE ERROR?
7332 023566 001033 ;BR TO ERROR HALT ON SEQ ERROR
7333 023570 012700 ;RO=410
7334 023574 012737 000000 ;LOC 0=-1
7335 023602 005037 177776 ;PS=0
7336 023606 106750 ;TRY MFPS W/MODE 5
7337 023610 103403 ;BR TO ERROR IF C-BIT SET
7338 023612 102402 ;BR TO ERROR IF V-BIT SET
7339 023614 100401 ;BR TO ERROR IF N-BIT SET
7340 023616 001404 ;
7341 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7342 ; CONDITIONAL BRANCH INST. AND <====
7343 ; REPLACE THE MOVE INSTRUCTION <====
7344 ; WHICH FOLLOWS W/ 764 <====
7345 023620 MFPS5A: MOV #571,-(R2) ;MOVE TO MAILBOX # ***** 571 *****
7346 023624 005242 000571 ;SET MSGTYP TO FATAL ERROR
7347 023626 000000 ;COND. CODES INCORRECT
7348 023630 022737 000377 000000 MFPS5B: CMP #377,@#0 ;CHECK DEST. DATA
7349 023636 001404 ;
7350 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7351 ; CONDITIONAL BRANCH INST. AND <====
7352 ; REPLACE THE MOVE INSTRUCTION <====
7353 ; WHICH FOLLOWS W/ 754 <====
7354 023640 012742 000572 ;MOVE TO MAILBOX # ***** 572 *****
7355 023644 005242 ;SET MSGTYP TO FATAL ERROR
7356 023646 000000 ;COND. CODES INCORRECT
7357 023650 020000 000406 MFPS5C: CMP R0,#406 ;CHECK MODE 5 REGISTER
7358 023654 001404 ;
7359 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7360 ; CONDITIONAL BRANCH INST. AND <====
7361 ; REPLACE THE MOVE INSTRUCTION <====
7362 ; WHICH FOLLOWS W/ 745 <====
7363 023656 012742 000573 ;MOVE TO MAILBOX # ***** 573 *****
7364 023662 005242 ;SET MSGTYP TO FATAL ERROR
7365 023664 000000 ;MODE 5 REGISTER NOT DECREMENTED BY 2
7366 ; OR SEQUENCE ERROR
7367 ;*****
7368 ;*****
7369 ;*****
7370 ;*****
7371 ;*****
7372 023666 005212 ;TEST 250 TEST MFPS MODE 6
7373 023670 022712 ;*****
7374 023674 001034 ;UPDATE TEST NUMBER
7375 023676 012700 ;SEQUENCE ERROR?
7376 023702 005037 000000 ;BR TO ERROR HALT ON SEQ ERROR
7377 023706 012737 000252 177776 ;RO=410
7378 023714 106760 ;LOC 0=0
7379 023720 102403 ;PS=252
7380 023722 103402 ;TRY MFPS W/MODE 6
7381 023724 001404 ;BR TO ERROR IF V-BIT SET
7382 023726 100404 ;BR TO ERROR IF C-BIT SET
7383 ;BR TO ERROR IF Z-BIT SET
7384 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

```

7384 ;*****
7385 ;*****
7386 ;*****
7387 023730 012742 000574 MFPS6A: MOV #574,-(R2) ;MOVE TO MAILBOX # ***** 574 *****
7388 023734 005242 ;SET MSGTYP TO FATAL ERROR
7389 023736 000000 ;COND. CODES INCORRECT
7390 023740 022737 000252 000000 MFPS6B: CMP #252,@#0 ;CHECK DEST. DATA
7391 023746 001404 ;
7392 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7393 ; CONDITIONAL BRANCH INST. AND <====
7394 ; REPLACE THE MOVE INSTRUCTION <====
7395 ; WHICH FOLLOWS W/ 753 <====
7396 023750 012742 000575 ;MOVE TO MAILBOX # ***** 575 *****
7397 023754 005242 ;SET MSGTYP TO FATAL ERROR
7398 023756 000000 ;DEST. DATA INCORRECT
7399 023760 022700 000401 MFPS6C: CMP #401,R0 ;CHECK DEST. REGISTER
7400 023764 001404 ;
7401 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7402 ; CONDITIONAL BRANCH INST. AND <====
7403 ; REPLACE THE MOVE INSTRUCTION <====
7404 ; WHICH FOLLOWS W/ 744 <====
7405 023766 012742 000576 ;MOVE TO MAILBOX # ***** 576 *****
7406 023772 005242 ;SET MSGTYP TO FATAL ERROR
7407 023774 000000 ;DEST. DATA INCORRECT
7408 ; OR SEQUENCE ERROR
7409 ;*****
7410 ;*****
7411 ;*****
7412 ;*****
7413 ;*****
7414 023776 005212 ;TEST 251 TEST MFPS MODE 7
7415 024000 022712 ;*****
7416 024004 001034 ;UPDATE TEST NUMBER
7417 024006 012700 ;SEQUENCE ERROR?
7418 024012 005037 000777 ;BR TO ERROR HALT ON SEQ ERROR
7419 024016 012737 000000 ;RO=777
7420 024024 106770 177407 000000 ;LOC 0=0
7421 024030 102403 ;PS=125
7422 024032 103002 ;TRY MFPS W/MODE 7
7423 024034 001401 ;BR TO ERROR IF V-BIT SET
7424 024036 100004 ;BR TO ERROR IF C-BIT SET
7425 ;BR TO ERROR IF Z-BIT SET
7426 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7427 ; CONDITIONAL BRANCH INST. AND <====
7428 ; REPLACE THE MOVE INSTRUCTION <====
7429 ; WHICH FOLLOWS W/ 763 <====
7430 024040 MFPS7A: MOV #577,-(R2) ;MOVE TO MAILBOX # ***** 577 *****
7431 024044 005242 ;SET MSGTYP TO FATAL ERROR
7432 024046 000000 ;CONDITION CODE INCORRECT
7433 024050 022737 042400 000000 MFPS7B: CMP #42400,@#0 ;CHECK DESTINATION DATA
7434 024056 001404 ;
7435 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7436 ; CONDITIONAL BRANCH INST. AND <====
7437 ; REPLACE THE MOVE INSTRUCTION <====
7438 ; WHICH FOLLOWS W/ 753 <====
7439 024060 012742 000600 ;MOVE TO MAILBOX # ***** 600 *****

```

```

7440 024064 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7441 024066 000000      HALT              ;DEST. DATA INCORRECT
7442 024070 024700 000777 MFPS7C: CMP      #777,R0 ;CHECK MODE 7 REGISTER
7443 024074 001404      BEQ      15T252
7444
7445
7446
7447
7448 024076 012742 000601 MOV      #601,-(R2) ;MOVE TO MAILBOX # ***** 601 *****
7449 024102 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7450 024104 000000      HALT              ;MODE 7 REGISTER MODIFIED
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464 024106 005212      INC      (R2)      ;UPDATE TEST NUMBER
7465 024114 001014 000252 CNE     #256,(R2) ;SEQUENCE ERROR?
7466 024116 001014 026060 000377 BNE     15T256-10 ;BR TO ERROR HALT ON SEQ ERROR
7467 024116 123727 026060 000377 CMPB    @#PASSPT,#377 ;ONLY DUE RESET EVERY 256. PASSES
7468 024124 001014      REST              ;BR IF TO SKIP TEST
7469 024126 012737 000357 177776 MOV      #357,@#PS ;MOV ONES TO PSW
7470 024134 000005      RESET            ;PSW CORRECT?
7471 024134 000005      BEQ      15T253
7472 024144 001404
7473
7474
7475
7476
7477 024146 012742 000602 MOV      #602,-(R2) ;MOVE TO MAILBOX # ***** 602 *****
7478 024152 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7479 024154 000000      HALT              ;RESET ALTERED PSW
7480
7481 024156
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491 024156 005212      INC      (R2)      ;UPDATE TEST NUMBER
7492 024160 022712 000253 CNE     #256,(R2) ;SEQUENCE ERROR?
7493 024164 001014 140000 153602 BNE     15T256-10 ;BR TO ERROR HALT ON SEQ ERROR
7494 024164 001014      BCS     @#SRM,PS ;SET USER MODE
7495 024174 012706 000001      MOV      #1,R6 ;SET BIT0

```

```

*****
;
; THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
; THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
; CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
; CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 256 (DECIMAL)
; PASSES.
*****

```

```

TEST 252 TEST THAT RESET DOES NOT CLEAR PSW
*****

```

```

REST:
*****
;
; THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
; DATA PATH COMPONENTS WITH USER MODE SET.
*****

```

```

TEST 253 TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
*****

```

```

7496 024200 000241      CLC              ;CLEAR C-BIT
7497 024202 006106      ROL      R6      ;ROTATE 1 POSITION
7498 024204 103376      BCC     USP1     ;BR IF NOT ALL DONE
7499 024206 001404      BEQ     USP1A    ;BR IF NO BITS PICKED
7500 024210 024767 140000 153560 BIC     @#SRM,PS ;CLEAR USER MODE
7501 024216 012742 140000 153512 MOV      #603,-(R2) ;MOVE TO MAILBOX # ***** 603 *****
7502 024222 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7503 024224 000000      HALT              ;USER MODE R6 PICKED A BIT
7504 024226
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516 024226 005212      INC      (R2)      ;UPDATE TEST NUMBER
7517 024230 022712 000254 CNE     #254,(R2) ;SEQUENCE ERROR?
7518 024234 001036 140000 153532 BNE     USP4-14 ;BR TO ERROR HALT ON SEQ ERROR
7519 024236 052767 140000 153512 BIS     @#SRM,PS ;SET USER MODE
7520 024244 012706 177777      MOV      #1,R6 ;SET USER R6 TO ALL ONES
7521 024250 022706 177777      CMP      #1,R6 ;READ AND CHECK USER R6
7522 024254 001404      BEQ     USP5     ;BR IF NO ERROR
7523 024256 042767 140000 153512 BIC     @#SRM,PS ;CLEAR USER MODE
7524 024264 012742 000604 MOV      #604,-(R2) ;MOVE TO MAILBOX # ***** 604 *****
7525 024270 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7526 024274 000000      HALT              ;USER R6 WILL NOT HOLD ALL ONES
7527 024274 042767 140000 153474 USP2:   BIC     @#SRM,PS ;SET KERNEL MODE
7528 024302 042706 177777      CMP      #1,R6 ;KERNEL MODE R6 ADDR. FROM USER MODE?
7529 024306 001004      BNE     USP3
7530
7531
7532
7533
7534 024310 012742 000605 MOV      #605,-(R2) ;MOVE TO MAILBOX # ***** 605 *****
7535 024314 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7536 024316 000000      HALT              ;DUAL ADDRESSING ERROR USER/KERNEL R6
7537 024320 005009      CLR      R6 ;CLEAR KERNEL MODE SP
7538 024322 052767 140000 153446 BIS     @#SRM,PS ;SET USER MODE
7539 024330 022706 177777      CMP      #1,R6 ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7540 024334 001404      BEQ     USP4     ;BR IF NO ERROR
7541 024336 012742 000606 MOV      #606,-(R2) ;MOVE TO MAILBOX # ***** 606 *****
7542 024342 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
7543 024344 000000      HALT              ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7544 024346 012706 140000 153416 USP4:   MOV      @#SRM,PS ;RESTORE SP USER
7545 024352 042767 140000 153416 BIC     @#SRM,PS ;SET KERNEL MODE
7546 024360 012706 000500      MOV      @#STBOT,R6 ;RESTORE SP KERNEL
7547
7548
7549
7550
7551

```

```

*****
;
; THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
; WITH R6 IN MODE 0.
*****

```

```

7552 ;
7553 ;
7554 ;
7555 ;
7556 024364 005212
7557 024366 022712 000255
7558 024372 001032
7559 024374 012706 000500
7560 024400 013767 140000
7561 024406 012706 026424 153370
7562 024412 006506
7563 024414 022767 140000 153354
7564 024422 001407
7565 024424 042767 140000 153344
7566 024432 012742 000607
7567 024436 005242
7568 024440 000000
7569 024442 022767 000500 001752
7570 024450 001407
7571 024452 042767 140000 153316
7572 024460 012742 000610
7573 024464 005242
7574 024466 000000
7575 024470
7576
7577
7578
7579
7580 024470 005212
7581 024472 022712 000256
7582 024476 001032
7583 024500 005067 153272
7584 024504 005006
7585 024506 012767 140000 153262
7586 024514 012706 026424
7587 024520 012746 000500
7588 024524 006606
7589 024526 022767 140000 153242
7590 024534 001407
7591 024536 042767 140000 153232
7592 024544 012742 000611
7593 024550 005242
7594 024552 000000
7595 024554 005067 153216
7596 024560 020627 000500
7597 024564 001404
7598
7599
7600
7601
7602 024566 012742 000612
7603 024572 005242
7604 024574 000000
7605
7606

```

```

7607 ;
7608 ;
7609 ;
7610 ;
7611 ;
7612 ;
7613 ;
7614 ;
7615 ;
7616 ;
7617 ;
7618 ;
7619 ;
7620 ;
7621 ;
7622 ;
7623 ;
7624 ;
7625 ;
7626 ;
7627 ;
7628 ;
7629 ;
7630 ;
7631 ;
7632 ;
7633 ;
7634 024576 005212
7635 024600 022712 000257
7636 024604 001062
7637 024606 012700 026214
7638 024612 012704 026252
7639 024616 012767 000017 000142
7640 024624 012067 000110
7641 024630 012401
7642 024632 012767 177777 000074
7643 024640 012703 000020
7644 024644 005267 000064
7645 024650 032701 100000
7646 024654 013705 177776
7647 024660 042705 177773
7648 024664 000165 024670
7649 024670 000167 000020
7650 024674 012767 024770 000042
7651 024702 012767 024752 000040
7652 024710 000167 000014
7653 024714 012767 024752
7654 024722 012767 024770 000022
7655 024730 006101 000020
7656
7657 024732 012737
7658 024734 000000
7659 024736 012736
7660 024740 000000
7661 024742 000137
7662 024744 000000

```

7663	024746	000137	
7664	024750	000000	
7665	024752	012702	000304
7666	024756	012742	000613
7667	024762	005242	
7668	024764	000000	
7669	024766	000000	
7670	024770	005303	
7671	024772	013705	177776
7672	024776	042705	177773
7673	025002	000165	025006
7674	025006	000167	177632
7675	025012	005367	177750
7676	025016	013705	177776
7677	025022	042705	177773
7678	025026	000165	025032
7679	025032	000167	177566

```

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

```

7680			
7681			
7682			
7683			
7684			
7685			
7686			
7687			
7688			
7689			
7690			
7691	025036	005212	
7692	025040	022712	
7693	025044	001052	000260
7694	025046	005000	
7695	025050	005001	
7696	025052	005002	
7697	025054	005003	
7698	025056	005004	
7699	025060	005005	
7700	025062	005006	
7701	025064	052700	000001
7702	025070	052701	000002
7703	025074	052702	000004
7704	025100	052703	000010
7705	025104	052704	000020
7706	025110	052705	000040
7707	025114	052706	000100
7708	025120	022706	000100
7709	025124	001022	
7710	025126	022705	000040
7711	025132	001017	
7712	025134	022704	000020
7713	025140	001014	
7714	025142	022703	000010
7715	025146	001011	
7716	025150	022702	000004
7717	025154	001006	
7718	025156	022701	000002
7719	025162	001003	
7720	025164	022700	000001
7721	025170	001404	
7722			
7723			
7724			
7725			
7726	025172		
7727	025172	012742	000614
7728	025176	005242	
7729	025200	000000	
7730	025202	012702	000304
7731	025206	012706	000500

```

; *****
; THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL
; REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET
; IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH
; REGISTER.
; *****
; TEST 260 DUAL REGISTER ADDRESSING TEST
; *****
*ST260: INC (R2) ; UPDATE TEST NUMBER
CMP #260,(R2) ; SEQUENCE ERROR?
BNE DAERR ; BR TO ERROR HALT ON SEQ ERROR
BITCLR: CLR R0 ; INITIALIZE ALL REGISTERS
CLR R1
CLR R2
CLR R3
CLR R4
CLR R5
CLR R6
BITSET: BIS #1,R0 ; SET R0=1
BIS #2,R1 ; R1=2
BIS #4,R2 ; R2=4
BIS #10,R3 ; R3=10
BIS #20,R4 ; R4=20
BIS #40,R5 ; R5=40
BIS #100,R6 ; R6=100
BITCHK: CMP #100,R6 ; TEST THAT NO DUAL ADDRESSING OCCURRED
BNE DAERR ; BR TO ERROR HALT IF ANY OTHER BITS ARE SET
CMP #40,R5
BNE DAERR
CMP #20,R4
BNE DAERR
CMP #10,R3
BNE DAERR
CMP #4,R2
BNE DAERR
CMP #2,R1
BNE DAERR
CMP #1,R0
BEQ BITCON
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 726 <====
DAERR: MOV #614,-(R2) ; MOVE TO MAILBOX # ***** 614 *****
INC ; SET MSGTYP TO FATAL ERROR
HALT ; DUAL ADDRESSING ERROR
BITCON: MOV #STESTN,R2 ; RESTORE POINTER
MOV #STBDT,R6 ; RESET STACK

```

```

7732 ;*****
7733 ; THIS TEST VERIFIES THAT THE UPPER BYTE OF THE PSW IS NOT AFFECTED
7734 ; WHEN THE PRIORITY LEVEL OR CC'S ARE CHANGED. ALL BITS ARE
7735 ; INITIALLY SET IN THE PSW, AND THE LOW BYTE IS CLEARED. A BIT
7736 ; INSTRUCTION VERIFIES THE DATA.
7737 ;*****
7738 ;*****
7739 ;*****
7740 ;*****
7741 ;*****
7742 025212 005212          000261          177776
7743 025214 022712          ;*****
7744 025220 001012          TST261: INC (R2)          ;UPDATE TEST NUMBER
7745 025222 052737          ;SEQUENCE ERROR?
7746 025230 105037          ;BR TO ERROR HALT ON SEQ ERROR
7747 025234 013700          ;SET ALL POSSIBLE BITS IN PSW
7748 025240 032700          ;CLR PR LEVEL AND CC'S
7749 025244 001006          ;COPY CONTENTS OF PSW
7750 025246 005037          ;TEST THAT UPPER BYTE IS UNAFFECTED
7751 025252 012742          ;CONTINUE IF OK
7752 025256 005242          BTERR: CLR @#PS          ;RETURN TO KERNEL MODE
7753 025260 000000          ;MOVE TO MAILBOX # ***** 615 *****
7754 025262 005037          ;SET MSGTYP TO FATAL ERROR
7755 ;*****
7756 ;*****
7757 ;*****
7758 ;*****
7759 ;*****
7760 ;*****
7761 ;*****
7762 ;*****
7763 ;*****
7764 025266 005212          000262
7765 025270 022712          ;*****
7766 025274 001010          ;*****
7767 025276 000277          ;*****
7768 025300 000252          ;*****
7769 025302 000167          ;*****
7770 025306 100403          ;*****
7771 025310 001002          ;*****
7772 025312 102401          ;*****
7773 025314 103404          ;*****
7774 ;*****
7775 ;*****
7776 ;*****
7777 ;*****
7778 025316 012742          000616
7779 025316 012742          ;*****
7780 025324 000000          ;*****
7781 025324 000000          ;*****
7782 ;*****

```

```

7783 ;*****
7784 ;*****
7785 ;*****
7786 ;*****
7787 ;*****
7788 ;*****
7789 ;*****
7790 ;*****
7791 ;*****
7792 ;*****
7793 ;*****
7794 ;*****
7795 ;*****
7796 ;*****
7797 ;*****
7798 ;*****
7799 ;*****
7800 025326 005212          000263
7801 025330 022712          ;*****
7802 025334 001002          ;*****
7803 025336 012767          000240 000024
7804 025344 012767          000017 000032
7805 025352 012767          000261 000102
7806 025360 012767          000001 000110
7807 025366 000277          ;*****
7808 025370 000000          ;*****
7809 025372 013704          177776
7810 025376 042704          177760
7811 025402 022704          ;*****
7812 025404 000000          ;*****
7813 025406 001404          ;*****
7814 ;*****
7815 ;*****
7816 ;*****
7817 ;*****
7818 025410 012742          000617
7819 025414 005242          ;*****
7820 025416 000000          ;*****
7821 025420 005367          177760
7822 025424 005267          177740
7823 025430 026727          000257
7824 025436 003753          ;*****
7825 025440 026727          177724 000260
7826 025446 001004          ;*****
7827 025450 012767          000017 177726
7828 025456 000743          ;*****
7829 025460 000257          ;*****
7830 025462 000000          ;*****
7831 025464 013704          177776
7832 025470 042704          177760
7833 025474 022704          ;*****
7834 025476 000000          ;*****
7835 025500 001404          ;*****
7836 ;*****
7837 ;*****
7838 ;*****

```


CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 172
 CFKAAC.P11 18-OCT-78 11:01 T264 END OF PASS SEQUENCE SEQ 0184

```

7904 026022 000207
7905
7906 026024 013700 000042 ACT: MOV @#42,R0 ;CHECK ACT
7907 026030 001405 BEQ GOAGIN ;KEEP GOING
7908 026032 000005 RESET
7909 026034 004710 $ENDAD: JSR PC,(R0) ;ACT HOOKS
7910 026036 000240 NOP
7911 026040 000240 NOP
7912 026042 000240 NOP
7913 026044 000167 152452 GOAGIN: JMP RESTRT ;DO NEXT PASS
7914 026050 EOP1:
7915 026054 012742 000621 MOV #621,-(R2) ;MOVE TO MAILBOX # ***** 621 *****
7916 026056 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7917 026060 000000 HALT ;SEQUENCE ERROR
7918 026062 177777 PASSPT: -1
7919 026064 000000 OCTPSS: -WORD 0 ;PASSCOUNT, OCTAL, STORED HERE
7920 026066 000000 ASCPSS: -WORD 0 ;PASSCOUNT, ASCII, BUILT HERE
7921 026068 005015 000000 000000 TITLE: -ASCII <15><12><0><0><0><0><0><0><0>.CFKAACO 11/34 BSC INST TST.<0><0><0><0><0>
7922 026074 000000 043103 040513
7923 026102 041501 020060 030461
7924 026110 031457 020064 051502
7925 026116 020103 047111 052123
7926 026124 052040 052123 000000
7927 026132 000000 000000 177400
7928
7929 026140 005015 000000 000000 MSG: -EVEN
7930 026146 000000 047105 020104 -ASCII <15><12><0><0><0><0><0><0>.END PASS .<0><0><0><0><0><377>
7931 026154 040520 051523 000000
7932 026162 000000 000000 177400
7933
7934 -EVEN
7935 ;*****
7936 ;THESE ARE A UNIT, INSERT NO CODE BETWEEN THEM *
7937 PSNUM: -WORD 0,0,0 ;
7938 -WORD 0,0,0 ;
7939 -WORD 0,0,0 ;
7940 BUFF: -WORD 0 ;
7941 ;*****

```

CFKAACO 11/34 BSC INST TST MACY11 30A(1052) 18-OCT-78 11:06 PAGE 173
 CFKAAC.P11 18-OCT-78 11:01 T264 END OF PASS SEQUENCE SEQ 0185

```

7942 026214 000402 BRTAB: BR +6
7943 026216 001002 BNE +6
7944 026220 001402 BEQ +6
7945 026222 002002 BGE +6
7946 026224 003402 BLT +6
7947 026226 003402 BBT +6
7948 026230 003402 BLE +6
7949 026232 100002 BPL +6
7950 026234 100402 BMI +6
7951 026236 101002 BHT +6
7952 026240 101402 BLOS +6
7953 026242 102002 BVC +6
7954 026244 102402 BVS +6
7955 026246 103002 BCC +6 ;SAME AS BHIS
7956 026250 103402 BCS +6 ;SAME AS BLO
7957
7958 -RADIX 2
7959 YNTAB: 1111111111111111 ;BR
1111000011110000 ;BNE: Z=0
0000111100001111 ;BEQ: Z=1
1100110000110011 ;BGE: N XOR V =0
0011001111001100 ;BLT: N XOR V =1
1100000000110000 ;BGT: Z+(N XOR V) =0
0011111111001111 ;BLE: Z+(N XOR V) =1
1111111100000000 ;BPL: N=0
0000000011111111 ;BMI: N=1
1010000010100000 ;BHT: C+Z=0
0101111101111111 ;BLOS: C+Z=1
1100110011001100 ;BVC: V=0
0011001100110011 ;BVS: V=1
1010101010101010 ;BCC: C=0
0101010101010101 ;BCS: C=1
7975 -RADIX 8
7976 026310 012737 026320 000024 PWRDN: MOV #PWRUP,@#24 ;SET UP FOR A POWER UP
7977 026316 000000 HALT
7978
7979 026320 012737 026310 000024 PWRUP: MOV #PWRDN,@#24 ;SET UP FOR A POWER FAIL
7980 026326 012706 000500 MOV #STBOT,R0 ;SET UP STACK POINTER
7981 026332 132767 000040 151761 BITB #40,SEVMV ;SHOULD PRINT?
7982 026340 001010 BNE PWR2 ;IF NOT: BR
7983 026342 012700 026366 MOV #PFMES,R0 ;GET POWER FAIL MESSG.
7984 026346 105737 177564 WATE: TSTB @TPS ;TPV READY?
7985 026348 100375 BPL WATE ;IF NOT: BR
7986 026354 112037 177566 MOVB (R0)+,@#TPB ;PRINT NEXT CHAR.
7987 026360 001372 BNE WATE ;IF NOT DONE: BR
7988 026362 000137 000500 PWR2: JMP @#START ;START PROGRAM AGAIN
7989
7990 026366 006412 047520 042527 PFMES: -ASCIIZ <12><15>.POWER FAILURE.<12><15>
7991 026374 040506 046111
7992 026402 051125 000015
7993
7994 026410 000006 -EVEN
7995 026424 .BLKW 6
7996 USTROT:
7997 ;*****
; THE FOLLOWING ARE SPECIAL CPU TRAP

```

```

7998 ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
7999 ;
8000 ;*****
8001 ;
8002 026424 000000 000622 T04: MOV #622,-(R2) ;MOVE TO MAILBOX # ***** 622 *****
8003 026424 012742 000622 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8004 026430 005242 000623 T010: HALT ;TRAPPED THRU LOC. 4
8005 026432 000000 000623 MOV #623,-(R2) ;MOVE TO MAILBOX # ***** 623 *****
8006 026434 012742 000623 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8007 026434 005242 000623 HALT ;TRAPPED THRU LOC. 10
8008 026440 005242 000624 T014: MOV #624,-(R2) ;MOVE TO MAILBOX # ***** 624 *****
8009 026442 000000 000624 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8010 026444 012742 000624 HALT ;TRAPPED THRU LOC. 14
8011 026444 012742 000624 MOV #624,-(R2) ;MOVE TO MAILBOX # ***** 624 *****
8012 026450 005242 000625 T030: INC -(R2) ;SET MSGTYP TO FATAL ERROR
8013 026452 000000 000625 HALT ;TRAPPED THRU LOC. 14
8014 026454 012742 000625 MOV #625,-(R2) ;MOVE TO MAILBOX # ***** 625 *****
8015 026454 005242 000625 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8016 026460 005242 000625 HALT ;TRAPPED THRU LOC. 30
8017 026462 000000 000626 T034: MOV #626,-(R2) ;MOVE TO MAILBOX # ***** 626 *****
8018 026464 012742 000626 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8019 026464 005242 000626 HALT ;TRAPPED THRU LOC. 34
8020 026470 005242 000627 T0114: MOV #627,-(R2) ;MOVE TO MAILBOX # ***** 627 *****
8021 026472 000000 000627 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8022 026474 012742 000627 HALT ;TRAPPED THRU LOC. 114
8023 026474 012742 000627 MOV #627,-(R2) ;MOVE TO MAILBOX # ***** 627 *****
8024 026500 005242 000630 T0244: INC -(R2) ;SET MSGTYP TO FATAL ERROR
8025 026502 000000 000630 HALT ;TRAPPED THRU LOC. 244
8026 026504 012742 000630 MOV #630,-(R2) ;MOVE TO MAILBOX # ***** 630 *****
8027 026504 005242 000630 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8028 026510 005242 000631 T0250: HALT ;TRAPPED THRU LOC. 250
8029 026512 000000 000631 MOV #631,-(R2) ;MOVE TO MAILBOX # ***** 631 *****
8030 026514 012742 000631 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8031 026520 005242 000631 HALT ;TRAPPED THRU LOC. 250
8032 026520 000000 000631 MOV #631,-(R2) ;MOVE TO MAILBOX # ***** 631 *****
8033 026522 000000 000631 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8034 026522 000000 000631 HALT ;TRAPPED THRU LOC. 250
.END

```

```

ABASE = 000000 28
ACDW1 = 000000 28
ACDW2 = 000000 28
ACPUOP = 000000 28
ACT1 = 026024 7858 43
ADC1 = 020064 6088 7860 7877 7906#
ADC2 = 020064 6090 6099 6095#
ADC3 = 020104 6103 6104#
ADC4 = 020114 6105 6114#
ADC5 = 020132 6117 6118 6119 6125#
ADDW0 = 000000 28
ADDW1 = 000000 28
ADDW10 = 000000 28
ADDW11 = 000000 28
ADDW12 = 000000 28
ADDW13 = 000000 28
ADDW14 = 000000 28
ADDW15 = 000000 28
ADDW2 = 000000 28
ADDW3 = 000000 28
ADDW4 = 000000 28
ADDW5 = 000000 28
ADDW6 = 000000 28
ADDW7 = 000000 28
ADDW8 = 000000 28
ADDW9 = 000000 28
ADD1 = 017670 6010 6011# 6017#
ADD2 = 017700 6012 6021#
ADD3 = 017714 6024 6025# 6031#
ADD4 = 017724 6026 6035#
ADD5 = 017742 6038 6039# 6045#
ADD6 = 017752 6040 6049#
ADD7 = 017764 6050 6051# 6057#
ADD8 = 017774 6052 6061#
ADD9 = 020014 6064 6065# 6066 6072#
ADEVCT = 000000 28
ADEVN = 000000 28
AENV = 000000 28
AENVM = 000000 28
AFATAL = 000000 28
AMADR1 = 000000 28
AMADR2 = 000000 28
AMADR3 = 000000 28
AMADR4 = 000000 28
AMAMS1 = 000000 28
AMAMS2 = 000000 28
AMAMS3 = 000000 28
AMAMS4 = 000000 28
AMSCAD = 000000 28
AMSGLG = 000000 28
AMSGTY = 000000 28
AMTYP1 = 000000 28
AMTYP2 = 000000 28
AMTYP3 = 000000 28
AMTYP4 = 000000 28
APASS = 000000 28

```

APRIOR=	000000	28							
AROUND	024730	7652	7655#						
ASCPSS	026064	7888*	7891*	7895*	7899*	7901	7920#		
ASL1	021370	6590	6592	6607	6613#				
ASL2	021406	6605	6602#						
ASL3	021426	6608	6617#						
ASL4	021442	6620	6621#	6627#					
ASL5	021452	6622	6631#						
ASL6	021476	6634	6635	6636	6637	6644#			
ASL7	021476	6634	6635	6636	6637	6644#			
ASR1	021530	6659	6660	6661	6667#				
ASR2	021550	6662	6671#	6677	6683#				
ASR3	021572	6675	6676	6677	6683#				
ASR4	021602	6678	6687#	6692	6698#				
ASR5	021616	6690	6691	6692	6698#				
ASR6	021656	6693	6704#	6708	6709	6716#			
ASR7	021656	6693	6704#	6708	6709	6716#			
ASWREG=	000000	28							
ARESTN=	000000	28							
AUNIT =	000000	28							
AUSWR =	000000	28							
AVECT1=	000000	28							
AVECT2=	000000	28							
BIC1	017020	5687	5688	5694#					
BIC2	017030	5689	5698#						
BIC3	017046	5701	5702	5708#					
BIS1	017110	5723	5724	5731#					
BIS2	017120	5730	5739#						
BIS3	017140	5738	5739	5740	5746#				
BITCHK	025120	7708#							
BITCLR	025046	7694#							
BITCON	025202	7721#	7730#						
BITSET	025064	7721#							
BIT1	016740	5650	5651	5657#					
BIT2	016740	5652	5662#						
BIT3	016756	5665	5666	5672#					
BRCT1	024766	7639*	7669#	7675*					
BRCT2	003040	1039	1045						
BRCT3	003050	1040	1044						
BRCT4	003060	1052	1058#						
BRH	024740	7640*	7660#						
BRN1	002720	945	951#						
BRN2	002730	946	956#						
BRN3	002740	945	964#						
BRV1A	002714	7853	7847#						
BRV1	002770	992	998#						
BRV2	003000	993	1003#						
BRV3	003010	1005	1011#						
BRZ1	002650	898	904#						
BRZ2	002660	899	909#						
BRZ3	002672	899	917#						
BR2	000602	134	140#						
BR3	000614	135	144#						
BR4	000622	145	153#						
BR5	000632	154	160#						
		155	164#						

BTCN	025262	7748	7753#						
BFERR	025246	7743	7749#						
BUFF	026412	7867	7939#						
BUILD	025734	7871	7884#	7658#					
CC	025734	7871	7884#						
CCERR	025502	7802	7840#						
CC1	025370	7803*	7808#	7822*	7823	7825			
CC2	025404	7804*	7812#	7821*	7821*				
CLRCN	025366	7807	7824	7828					
CLR1	017456	5903	5904	5905	5911#				
CMP1	020330	6209	6208	6214#					
CMP2	020330	6209	6218#						
CMP3	020352	6222	6223#	6229#					
CMP4	020362	6224	6233#						
CMP5	020406	6237	6238	6239	6245#				
CMP6	020436	6249	6249#						
CMP7	020436	6252	6253	6259#					
COM1	020476	6275	6276	6282#					
CON1	024770	7650	7654	7670#					
CON2	025420	7812	7821#						
CON3	025420	7812	7821#						
DAERR	025772	7893	7849	7711	7713	7715	7717	7719	7726#
DEC1	017316	5826	5827	5828	5834#				
DEC2	017326	5829	5838#						
DEC3	017342	5841	5842	5848#					
DEC4	017352	5844	5852#						
DEC5	017376	5857	5866#	5862#					
DEC6	017376	5857	5866#	5862#					
DEC7	017420	5870	5871	5872	5878#				
DNMB0A	010520	3459	3460	3461	3467#				
DNMB0B	010530	3463	3471#						
DNMB2A	010756	3569	3570	3571	3577#				
DNMB2B	010766	3572	3581#						
DNMB2C	011002	3582	3590#						
DNMB2D	011016	3592	3593	3599#					
DNMB2E	011026	3594	3603#						
DNMB2F	011044	3605	3613#						
DNMB3A	011126	3639	3640	3641	3647#				
DNMB3B	011136	3649	3649#						
DNMB3C	011154	3652	3660#						
DNMB3D	011172	3663	3664	3670#					
DNMB3E	011202	3665	3674#						
DNMB4A	011372	3743	3743#	3744	3750#				
DNMB4B	011402	3745	3754#						
DNMB4C	011420	3755	3763#						
DNMB4D	011430	3764	3770#						
DNMB4E	011440	3765	3774#						
DNMB4F	011454	3775	3783#						
DNM03A	007612	3069	3070	3070	3076#				
DNM03B	007622	3071	3080#						
DNM03C	007632	3081	3088#						
DNM1	007464	3002	3010#						
DNM1A	010576	3492	3493	3494	3500#				
DNM1B	010506	3495	3504#						
DNM2	007500	3519	3526#						
DNM2A	010654	3525	3526	3532#					

MDMSB	012772	4280	4289#							
MDMS	013070	4280	4288#							
MDMSD	013026	4299	4307#							
MDM5E	013054	4310	4318#							
MDM6A	013124	4347	4348#	4354#						
MDM6B	013134	4349	4358#							
MDM6C	013152	4358	4366#							
MDM6D	013222	4379	4387#							
MDM6E	013222	4379	4387#							
MDM7A	013276	4415	4416#	4422#						
MDM7B	013306	4417	4426#							
MDM7C	013324	4427	4435#							
MDM7D	013344	4436	4444#							
MDM7E	013370	4446	4454#							
MFP10	024442	7564	7569#							
MFP10A	024470	7570	7575#							
MFPS1	023206	7177	7186#							
MFPS2A	023276	7211	7212#	7213	7219#					
MFPS2B	023306	7214	7223#							
MFPS2C	023326	7224	7232#							
MFPS3A	023404	7253	7254#	7255	7261#					
MFPS3B	023414	7256	7265#							
MFPS3C	023434	7266	7274#							
MFPS4A	023512	7295	7296#	7297	7303#					
MFPS4B	023512	7295	7296#							
MFPS4C	023542	7308	7316#							
MFPS5A	023620	7337	7338#	7339	7345#					
MFPS5B	023630	7340	7349#							
MFPS5C	023650	7350	7358#							
MFPS6A	023740	7379	7380#	7381	7387#					
MFPS6B	023740	7382	7391#							
MFPS6C	023760	7392	7400#							
MFPS7A	024040	7421	7422#	7423	7429#					
MFPS7B	024050	7424	7433#							
MFPS7C	024070	7434	7442#							
MDV1	016640	5614	5614#	5620#						
MDV2	016650	5615	5625#							
MDV3	016666	5628	5629#	5635#						
MRK1	022246	6889	6896#	6898	6900	6907#				
MRK2	022270	6896	6897#							
MRK3	022300	6902	6911#							
MRK4	022322	6904	6916#							
MRK5	022334	6915	6920#							
MRK6	022350	6921	6928#							
MSG	026140	7865	7879#							
MTP10	024554	6953	6954#							
MTPS1	022440	6965	6966#	6967	6973#					
MTPS1A	022440	6965	6966#							
MTPS2	022514	6990	6998#							
MTPS3	022604	7021	7029#							
MTPS4	022672	7051	7059#							
MTPS5	023042	7081	7089#							
MTPS6	023042	7081	7089#							
MTPS7	023132	7141	7149#							
NBR	024744	7650*	7653#	7662#						
NEG00	004010	1501	1502	1503	1509#					

NEG01	004020	1504	1514#							
NEG02	004034	1515	1524#							
NEG03	004050	1526	1527#	1528	1534#					
NEG04	004060	1529	1538#							
NEG1	020174	6151	6152#	6153	6159#					
NEG10	004134	1559	1559#	1560	1566#					
NEG11	004152	1561	1561#							
NEG12	004152	1572	1580#							
NEG13	004166	1582	1583#	1584	1590#					
NEG14	004176	1585	1594#							
NEG2	020204	6154	6163#	1616	1622#					
NEG20	004454	1614	1615#							
NEG21	004454	1617	1626#							
NEG22	004302	1632	1640#							
NEG3	020226	6167	6168#	6169	6175#					
NEG30	004624	1818	1819#	1820	1826#					
NEG31	004634	1831	1831#							
NEG32	004650	1831	1839#							
NEG33	004674	1842	1850#							
NEG34	004710	1851	1859#							
NEG4	020236	6170	6179#							
NEG40	005266	2057	2058#	2059	2065#					
NEG41	005276	2060	2069#							
NEG42	005312	2070	2078#							
NEG5	020256	6183	6184#	6185	6191#					
NEG50	005366	2102	2103#	2104	2110#					
NEG51	005376	2105	2114#							
NEG52	005416	2113	2123#							
NEG60	005440	2147	2148#	2149	2155#					
NEG61	005500	2150	2159#							
NEG70	005550	2180	2181#	2182	2188#					
NEG71	005560	2183	2192#							
OCTPSS	026062	7887*	7889#	7893*	7897*	7902	7919#			
PASSPT	026060	7887*	7889#	7893*	7918*					
PS	177776	7983	7990#	810	827*	828	845*	846	863*	864
		9#	809*	7018*	7020	7048*	7050	7108*	7110	7138*
			7293*	7293*	7335*	7377*	7419*	7471	7484*	7500*
			7566*	7566*	7563*	7565*	7571*	7583*	7585*	7589
			7153*	7809	7831					
PSNUM	026170	7936#							6933#	6952
PUSRM	030000	13#							7144*	7188*
PWRDN	026310	119	7976#	7979					7174*	7209*
PWRUP	026320	7976	7979#						7251*	7251*
PR2	026362	7982	7988#						7299*	7318*
REG1	001530	505#	508						7523*	7538*
REG1A	001574	529#	532						7523*	7545*
REG1E	001542	506	514#						7744*	7745*
REG2	001644	553#	556							
REG2A	001672	549	549	557	566#					
REG2B	001720	577#	580							
REG2C	001752	587	588#							
REG3	002000	599#	602							
REG3A	002044	623#	626							
REG3E	002012	600	608#							
REG4	002114	648#	651							

TST136	011062	3563	3614	3628#
TST137	011222	3630	3676	3689#
TST14	001370	428	432	445#
TST140	011330	3691	3720	3733#
TST141	011472	3739	3784	3797#
TST143	011600	3843	3888	3894#
TST144	012020	3886	3914	3933#
TST145	012074	3935	3953	3972#
TST146	012166	3974	4002	4020#
TST147	012322	4021	4068	4086#
TST150	012500	4088	4134	4153#
TST151	012574	4155	4181	4202#
TST152	012730	4204	4249	4271#
TST153	013072	4273	4319	4340#
TST154	013242	4343	4388	4408#
TST155	013242	4343	4388	4408#
TST156	013474	4489	4518#	
TST157	013560	4527	4555#	
TST16	001454	4659	468	481#
TST160	013644	4598	4641	466#
TST161	013730	4617	4639	4652#
TST162	014024	4617	4639	4652#
TST163	014166	4664	4704	4725#
TST164	014344	4727	4773	4794#
TST165	014506	4796	4830	4852#
TST166	014566	4899	4861	4887#
TST167	001506	483	486	499#
TST170	014714	4923	4929	4950#
TST171	014772	4959	4982#	
TST172	015040	4988	4996	5016#
TST173	015162	5046	5073	5091#
TST174	015162	5046	5073	5091#
TST175	015224	5083	5087	5108#
TST176	015304	5110	5124	5147#
TST177	015374	5163	5189#	
TST20	001652	501	509	523#
TST200	015442	5196	5222#	
TST201	015514	5229	5277#	
TST202	016060	5388	5412	5433#
TST203	016536	5546	5568#	
TST204	016674	5619	5630	5650#
TST205	016742	5645	5667	5680#
TST206	016766	5688	5703	5716#
TST207	017056	572	5734	5747#
TST21	001622	578	581	586#
TST210	017129	582	584	589#
TST212	017430	5899	5906	5920#
TST213	017466	5922	5944	5957#
TST214	017552	5955	5981	6004#
TST215	017640	6006	6067	6081#
TST216	020024	6006	6067	6081#

TST217	020142	6083	6120	6144#
TST22	001676	571#		
TST220	020266	6146	6186	6200#
TST221	020446	6202	6254	6268#
TST222	020506	6270	6277	6302#
TST223	021066	6371	6436	6469#
TST224	021174	6450	6503	6516#
TST225	021336	6518	6570	6583#
TST226	021706	6585	6639	6652#
TST227	021766	6639	6693#	
TST228	022166	6754	6771	6788#
TST229	022174	6737	6757	6788#
TST230	022106	6790	6819	6839#
TST231	022174	6841	6863	6883#
TST232	022360	6885	6923	6946#
TST233	022532	698	6998	7014#
TST234	022622	7014	7030	7043#
TST24	002022	595	603	617#
TST240	022706	7045	7060	7073#
TST241	023070	707	7090	7103#
TST242	023060	7105	7120	7133#
TST243	023150	7135	7150	7171#
TST244	023242	7173	7191	7204#
TST245	023344	7206	7233	7246#
TST246	023452	7248	7275	7288#
TST247	023560	7290	7317	7330#
TST25	002072	619	619	647#
TST250	023666	7332	7359	7372#
TST251	023776	7374	7401	7414#
TST252	024106	7416	7443	7464#
TST253	024156	7466	7472	7491#
TST254	024276	7493	7516#	
TST255	024364	7556#		
TST256	024470	7558#	7580#	
TST257	024576	7582	7597	7634#
TST26	002136	644	652	666#
TST260	025036	7621#		
TST261	025242	7741#		
TST262	025266	7766	7773	7800#
TST263	025426	7766	7773	7800#
TST264	025532	7851#		
TST27	002206	666	677	692#
TST29	007700	194	198	211#
TST30	002252	718	727	741#
TST31	002322	743	751	765#
TST32	002366	767	776	805#
TST33	002436	807	811	824#
TST34	002436	826	847	860#
TST35	002572	862	865	892#
TST37	002630	213	217	230#
TST4	000736	894	912	939#
TST40	002700	941	959	986#
TST41	002750	941	959	986#

453#	471#	489#	512#	537#	606#	631#	655#	680#	705#	730#	754#	779#
1385#	1411#	1433#	1462#	1479#	1507#	1518#	1533#	1542#	1564#	1575#	1588#	1598#
1620#	1633#	1644#	1679#	1692#	1728#	1745#	1781#	1797#	1824#	1834#	1845#	1854#
1855#	1899#	1908#	1944#	1955#	1986#	1997#	2022#	2039#	2063#	2073#	2082#	2088#
2118#	2127#	2163#	2163#	2188#	2195#	2255#	2270#	2309#	2363#	2370#	2382#	2385#
2416#	2422#	2468#	2478#	2495#	2516#	2525#	2539#	2553#	2566#	2575#	2590#	2593#
2625#	2633#	2668#	2678#	2718#	2728#	2770#	2777#	2803#	2803#	2808#	2815#	2815#
2928#	2933#	2969#	2980#	3005#	3014#	3024#	3036#	3046#	3074#	3089#	3115#	3143#
3172#	3207#	3217#	3247#	3275#	3285#	3315#	3322#	3353#	3363#	3390#	3416#	3442#
3465#	3478#	3498#	3508#	3533#	3541#	3551#	3575#	3585#	3597#	3608#	3617#	3645#
3655#	3665#	3679#	3704#	3714#	3723#	3748#	3755#	3768#	3778#	3789#	3800#	3800#
4041#	4055#	4065#	4071#	4098#	4108#	4117#	4127#	4137#	4165#	4175#	4184#	4213#
4222#	4233#	4243#	4252#	4283#	4293#	4302#	4311#	4322#	4352#	4362#	4371#	4382#
4391#	4422#	4430#	4439#	4444#	4458#	4462#	4472#	4483#	4508#	4518#	4527#	4530#
4690#	4707#	4740#	4757#	4777#	4805#	4819#	4833#	4833#	4866#	4876#	4886#	4900#
4930#	4935#	5002#	5032#	5090#	5118#	5127#	5136#	5144#	5166#	5176#	5184#	5213#
5310#	5325#	5331#	5344#	5355#	5371#	5385#	5405#	5415#	5444#	5454#	5462#	5480#
5549#	5558#	5568#	5584#	5595#	5611#	5625#	5642#	5655#	5670#	5684#	5699#	5719#
6070#	6079#	6093#	6108#	6123#	6135#	6173#	6189#	6211#	6227#	6243#	6255#	6280#
6332#	6344#	6359#	6382#	6392#	6413#	6413#	6427#	6446#	6459#	6470#	6479#	6484#
6681#	6685#	6685#	6686#	66905#	66922#	66956#	66971#	66993#	70022#	70022#	7054#	7063#
7064#	7064#	7093#	7114#	7123#	7144#	7153#	7180#	7194#	7217#	7227#	7259#	7269#
7278#	7303#	7311#	7320#	7344#	7344#	7353#	7362#	7385#	7395#	7404#	7437#	7446#
7475#	7500#	7500#	7524#	7533#	7533#	7533#	7533#	7533#	7533#	7533#	7533#	7533#
814#	833#	850#	850#	850#	850#	850#	850#	850#	850#	850#	850#	850#
1120#	1127#	1157#	1174#	1199#	1209#	1236#	1244#	1276#	1290#	1322#	1337#	1370#
1385#	1411#	1433#	1462#	1479#	1507#	1518#	1533#	1542#	1564#	1575#	1588#	1598#
1620#	1633#	1644#	1679#	1692#	1728#	1745#	1781#	1797#	1824#	1834#	1845#	1854#
1855#	1899#	1908#	1944#	1955#	1986#	1997#	2022#	2039#	2063#	2073#	2082#	2088#
2118#	2127#	2163#	2163#	2188#	2195#	2255#	2270#	2309#	2363#	2370#	2382#	2385#
2416#	2422#	2468#	2478#	2495#	2516#	2525#	2539#	2553#	2566#	2575#	2590#	2593#
2625#	2633#	2668#	2678#	2718#	2728#	2770#	2777#	2803#	2803#	2808#	2815#	2815#
2928#	2933#	2969#	2980#	3005#	3014#	3024#	3036#	3046#	3074#	3089#	3115#	3143#
3172#	3207#	3217#	3247#	3275#	3285#	3315#	3322#	3353#	3363#	3390#	3416#	3442#
3465#	3478#	3498#	3508#	3533#	3541#	3551#	3575#	3585#	3597#	3608#	3617#	3645#
3655#	3665#	3679#	3704#	3714#	3723#	3748#	3755#	3768#	3778#	3789#	3800#	3800#
4041#	4055#	4065#	4071#	4098#	4108#	4117#	4127#	4137#	4165#	4175#	4184#	4213#
4222#	4233#	4243#	4252#	4283#	4293#	4302#	4311#	4322#	4352#	4362#	4371#	4382#
4391#	4422#	4430#	4439#	4444#	4458#	4462#	4472#	4483#	4508#	4518#	4527#	4530#
4690#	4707#	4740#	4757#	4777#	4805#	4819#	4833#	4833#	4866#	4876#	4886#	4900#
4930#	4935#	5002#	5032#	5090#	5118#	5127#	5136#	5144#	5166#	5176#	5184#	5213#
5310#	5325#	5331#	5344#	5355#	5371#	5385#	5405#	5415#	5444#	5454#	5462#	5480#
5549#	5558#	5568#	5584#	5595#	5611#	5625#	5642#	5655#	5670#	5684#	5699#	5719#
6070#	6079#	6093#	6108#	6123#	6135#	6173#	6189#	6211#	6227#	6243#	6255#	6280#
6332#	6344#	6359#	6382#	6392#	6413#	6413#	6427#	6446#	6459#	6470#	6479#	6484#
6681#	6685#	6685#	6686#	66905#	66922#	66956#	66971#	66993#	70022#	70022#	7054#	7063#
6822#	6850#	6850#	6866#	6866#	6905#	6905#	6926#	6926#	6926#	6926#	6926#	6926#

\$XXX = 000716

708#	7093#	7114#	7123#	7144#	7153#	7180#	7194#	7217#	7227#	7236#	7259#	7269#
7278#	7311#	7320#	7344#	7344#	7353#	7362#	7385#	7395#	7404#	7426#	7437#	7446#
7475#	7500#	7500#	7524#	7533#	7533#	7533#	7533#	7533#	7533#	7533#	7533#	7533#
814#	833#	850#	850#	850#	850#	850#	850#	850#	850#	850#	850#	850#
1120#	1127#	1157#	1174#	1199#	1209#	1236#	1244#	1276#	1290#	1322#	1337#	1370#
1385#	1411#	1433#	1462#	1479#	1507#	1518#	1533#	1542#	1564#	1575#	1588#	1598#
1620#	1633#	1644#	1679#	1692#	1728#	1745#	1781#	1797#	1824#	1834#	1845#	1854#
1855#	1899#	1908#	1944#	1955#	1986#	1997#	2022#	2039#	2063#	2073#	2082#	2088#
2118#	2127#	2163#	2163#	2188#	2195#	2255#	2270#	2309#	2363#	2370#	2382#	2385#
2416#	2422#	2468#	2478#	2495#	2516#	2525#	2539#	2553#	2566#	2575#	2590#	2593#
2625#	2633#	2668#	2678#	2718#	2728#	2770#	2777#	2803#	2803#	2808#	2815#	2815#
2928#	2933#	2969#	2980#	3005#	3014#	3024#	3036#	3046#	3074#	3089#	3115#	3143#
3172#	3207#	3217#	3247#	3275#	3285#	3315#	3322#	3353#	3363#	3390#	3416#	3442#
3465#	3478#	3498#	3508#	3533#	3541#	3551#	3575#	3585#	3597#	3608#	3617#	3645#
3655#	3665#	3679#	3704#	3714#	3723#	3748#	3755#	3768#	3778#	3789#	3800#	3800#
4041#	4055#	4065#	4071#	4098#	4108#	4117#	4127#	4137#	4165#	4175#	4184#	4213#
4222#	4233#	4243#	4252#	4283#	4293#	4302#	4311#	4322#	4352#	4362#	4371#	4382#
4391#	4422#	4430#	4439#	4444#	4458#	4462#	4472#	4483#	4508#	4518#	4527#	4530#
4690#	4707#	4740#	4757#	4777#	4805#	4819#	4833#	4833#	4866#	4876#	4886#	4900#
4930#	4935#	5002#	5032#	5090#	5118#	5127#	5136#	5144#	5166#	5176#	5184#	5213#
5310#	5325#	5331#	5344#	5355#	5371#	5385#	5405#	5415#	5444#	5454#	5462#	5480#
5549#	5558#	5568#	5584#	5595#	5611#	5625#	5642#	5655#	5670#	5684#	5699#	5719#
6070#	6079#	6093#	6108#	6123#	6135#	6173#	6189#	6211#	6227#	6243#	6255#	6280#
6332#	6344#	6359#	6382#	6392#	6413#	6413#	6427#	6446#	6459#	6470#	6479#	6484#
6681#	6685#	6685#	6686#	66905#	66922#	66956#	66971#	66993#	70022#	70022#	7054#	7063#
6822#	6850#	6850#	6866#	6866#	6905#	6905#	6926#	6926#	6926#	6926#	6926#	6926#

= 026524

.\$X = 000500

COMEN	1#	140	144	160	164	198	217	236	254	289	308	318	341	362	385
ENDCOM	1#	140	144	160	164	198	217	236	254	289	308	318	341	362	385
ERROR	1#	140	144	160	164	198	217	236	254	289	308	318	341	362	385
1#	1#	140	144	160	164	198	217	236	254	289	308	318	341	362	385
432	450	468	486	504	522	540	558	576	594	612	630	648	666	684	702
776	811	829	847	865	883	901	919	937	955	973	991	1009	1027	1045	1063
1124	1154	1184	1214	1244	1274	1304	1334	1364	1394	1424	1454	1484	1514	1544	1574
1725	1749	1773	1797	1821	1845	1869	1893	1917	1941	1965	1989	2013	2037	2061	2085
2025	2036	2047	2058	2069	2080	2091	2102	2113	2124	2135	2146	2157	2168	2179	2190
2299	2332	2364	2397	2430	2463	2496	2529	2562	2595	2628	2661	2694	2727	2760	2793
2829	2862	2895	2928	2961	2994	3027	3060	3093	3126	3159	3192	3225	3258	3291	3324
3275	3308	3341	3374	3407	3440	3473	3506	3539	3572	3605	3638	3671	3704	3737	3770
3814	3847	3880	3913	3946	3979	4012	4045	4078	4111	4144	4177	4210	4243	4276	4309
4352	4385	4418	4451	4484	4517	4550	4583	4616	4649	4682	4715	4748	4781	4814	4847
4890	4923	4956	4989	5022	5055	5088	5121	5154	5187	5220	5253	5286	5319	5352	5385
5428	5461	5494	5527	5560	5593	5626	5659	5692	5725	5758	5791	5824	5857	5890	5923
5956	5989	6022	6055	6088	6121	6154	6187	6220	6253	6286	6319	6352	6385	6418	6451
6484	6517	6550	6583	6616	6649	6682	6715	6748	6781	6814	6847	6880	6913	6946	6979
7012	7045	7078	7111	7144	7177	7210	7243	7276	7309	7342	7375	7408	7441	7474	7507
7540	7573	7606	7639	7672	7705	7738	7771	7804	7837	7870	7903	7936	7969	8002	8035
8068	8101	8134	8167	8200	8233	8266	8299	8332	8365	8398	8431	8464	8497	8530	8563
8596	8629	8662	8695	8728	8761	8794	8827	8860	8893	8926	8959	8992	9025	9058	9091
9124	9157	9190	9223	9256	9289	9322	9355	9388	9421	9454	9487	9520	9553	9586	9619
9652	9685	9718	9751	9784	9817	9850	9883	9916	9949	9982	10015	10048	10081	10114	10147
10180	10213	10246	10279	10312	10345	10378	10411	10444	10477	10510	10543	10576	10609	10642	10675
10708	10741	10774	10807	10840	10873	10906	10939	10972	11005	11038	11071	11104	11137	11170	11203
11236	11269	11302	11335	11368	11401	11434	11467	11500	11533	11566	11599	11632	11665	11698	11731
11764	11797	11830	11863	11896	11929	11962	11995	12028	12061	12094	12127	12160	12193	12226	12259
12292	12325	12358	12391	12424	12457	12490	12523	12556	12589	12622	12655	12688	12721	12754	12787
12820	12853	12886	12919	12952	12985	13018	13051	13084	13117	13150	13183	13216	13249	13282	13315
13348	13381	13414	13447	13480	13513	13546	13579	13612	13645	13678	13711	13744	13777	13810	13843
13876	13909	13942	13975	14008	14041	14074	14107	14140	14173	14206	14239	14272	14305	14338	14371
14404	14437	14470	14503	14536	14569	14602	14635	14668	14701	14734	14767	14800	14833	14866	14899
14932	14965	14998	15031	15064	15097	15130	15163	15196	15229	15262	15295	15328	15361	15394	15427
15460	15493	15526	15559	15592	15625	15658	15691	15724	15757	15790	15823	15856	15889	15922	15955
15988	16021	16054	16087	16120	16153	16186	16219	16252	16285	16318	16351	16384	16417	16450	16483
16516	16549	16582	16615	16648	16681	16714	16747	16780	16813	16846	16879	16912	16945	16978	17011
17044	17077	17110	17143	17176	17209	17242	17275	17308	17341	17374	17407	17440	17473	17506	17539
17572	17605	17638	17671	17704	17737	17770	17803	17836	17869	17902	17935	17968	18001	18034	18067
18100	18133	18166	18199	18232	18265	18298	18331	18364	18397	18430	18463	18496	18529	18562	18595
18628	18661	18694	18727	18760	18793	18826	18859	18892	18925	18958	18991	19024	19057	19090	19123
19156	19189	19222	19255	19288	19321	19354	19387	19420	19453	19486	19519	19552	19585	19618	19651
19684	19717	19750	19783	19816	19849	19882	19915	19948	19981	20014	20047	20080	20113	20146	20179
20212	20245	20278	20311	20344	20377	20410	20443	20476	20509	20542	20575	20608	20641	20674	20707
20740	20773	20806	20839	20872	20905	20938	20971	21004	21037	21070	21103	21136	21169	21202	21235
21268	21301	21334	21367	21400	21433	21466	21499	21532	21565	21598	21631	21664	21697	21730	21763
21796	21829	21862	21895	21928	21961	21994	22027	22060	22093	22126	22159	22192	22225	22258	22291
22324	22357	22390	22423	22456	22489	22522	22555	22588	22621	22654	22687	22720	22753	22786	22819
22852	22885	22918	22951	22984	23017	23050	23083	23116	23149	23182	23215	23248	23281	23314	23347
23380	23413	23446	23479	23512	23545	23578	23611	23644	23677	23710	23743	23776	23809	23842	23875
23908	23941	23974	24007	24040	24073	24106	24139	24172	24205	24238	24271	24304	24337	24370	24403
24436	24469	24502	24535	24568	24601	24634	24667	24700	24733	24766	24799	24832	24865	24898	24931
24964	24997	25030	25063	25096	25129	25162	25195	25228	25261	25294	25327	25360	25393	25426	25459
25492	25525	25558	25591	25624	25657	25690	25723	25756	25789	25822	25855	25888	25921	25954	25987
26020	26053	26086	26119	26152	26185	26218	26251	26284	26317	26350	26383	26416	26449	26482	26515
26548	26581	26614	26647	26680	26713	26746	26779	26812	26845	26878	26911	26944	26977	27010	27043
27076	27109	27142	27175	27208	27241	27274	27307	27340	27373	27406	27439	27472	27505	27538	27571
27604	27637	27670	27703	27736	27769	27802	27835	27868	27901	27934	27967	28000	28033	28066	28099
28132	28165	28198	28231	28264	28297	28330	28363	28396	28429	28462	28495	28528	28561	28594	28627
28660	28693	28726	28759	28792	28825	28858	28891	28924	28957	28990	29023	29056	29089	29122	29155
29188	29221	29254	29287	29320	29353	29386	29419	29452	29485	29518	29551	29584	29617	29650	29683
29716	29749	29782	29815	29848	29881	29914	29947	29980	30013	30046	30079	30112	30145	30178	30211
30244	30277	30310	30343	30376	30409	30442	30475	30508	30541	30574	30607	30640	30673	30706	30739
30772	30805	30838	30871	30904	30937	30970	31003	31036	31069	31102	31135	31168	31201	31234	31267
31300	31333	31366	31399	31432	31465	31498	31531	31564	31597	31630	31663	31696	31729	31762	31795
31828	31861	31894	31927	31960	31993	32026	32059	32092	32125	32158	32191	32224	32257	32290	32323
32356	32389	32422	32455	32488	32521	32554	32587	32620	32653	32686	32719	32752	32785	32818	32851
32884	32917	32950	32983	33016	33049	33082	33115	33148	33181	33214	33247	33280	33313	33346	33379
33412	33445	33478	33511	33544	33577	33610	33643	33676	33709	33742	33775	33808	33841	33874	33907

	7579	7608	7631	7633	7681	7688	7690	7732	7738	7740	7755	7761	7763	7783	7797
SWRSU	431	141	150	161	170	203	222	241	259	294	313	323	347	367	391
TYPBIN	782	455	473	491	515	540	563	585	609	634	658	683	708	733	757
TYPDEC	1131	816	834	852	870	905	918	952	965	999	1012	1046	1059	1111	1122
TYPNAM	1464	1159	1177	1200	1212	1238	1251	1278	1293	1324	1340	1372	1388	1420	1436
TYPNUM	1730	1482	1510	1520	1535	1544	1567	1577	1591	1600	1623	1637	1646	1681	1695
TYPDCS	2030	1748	1783	1800	1827	1836	1847	1856	1867	1895	1911	1946	1960	1988	2000
TYPDXT	2305	2042	2066	2079	2084	2111	2120	2130	2156	2165	2189	2201	2227	2240	2273
TYPXT	2635	2338	2370	2387	2409	2429	2462	2471	2487	2497	2497	2497	2518	2533	2603
\$\$SERCD	2930	2644	2678	2688	2721	2730	2763	2772	2799	2826	2856	2891	2904	2915	2955
	3219	2941	2972	2982	3007	3016	3026	3038	3048	3077	3089	3117	3145	3174	3209
	3533	3249	3277	3287	3317	3326	3355	3365	3392	3418	3444	3468	3477	3501	3510
	3751	3544	3553	3578	3587	3600	3610	3619	3648	3657	3671	3681	3707	3710	3749
	3958	3760	3771	3780	3789	3815	3824	3833	3858	3867	3876	3901	3910	3919	3949
	4177	3987	3998	4007	4034	4043	4055	4064	4073	4101	4110	4119	4129	4139	4168
	4384	4186	4215	4224	4236	4245	4254	4286	4295	4304	4315	4324	4355	4364	4373
	4710	4393	4423	4432	4441	4451	4460	4495	4533	4569	4600	4630	4645	4678	4693
	5065	4743	4760	4779	4808	4827	4836	4869	4905	4935	4965	4992	5001	5028	5056
	5373	5092	5120	5129	5159	5169	5202	5235	5289	5298	5312	5324	5333	5346	5359
	5673	5387	5427	5446	5465	5483	5501	5517	5533	5552	5577	5586	5621	5636	5658
	5972	5695	5709	5732	5747	5781	5797	5812	5835	5849	5879	5886	5912	5935	5950
	6246	5987	6018	6032	6046	6058	6073	6096	6111	6126	6176	6176	6192	6215	6230
	6532	6246	6283	6317	6332	6347	6362	6385	6400	6415	6430	6464	6479	6493	6509
	6862	6547	6586	6576	6599	6614	6628	6645	6668	6684	6699	6717	6752	6773	6807
	7086	6859	6899	6893	6908	6917	6928	6948	6968	6985	7004	7025	7035	7056	7085
	7313	7095	7116	7125	7146	7155	7182	7196	7220	7229	7238	7252	7271	7280	7304
	7541	7322	7346	7355	7364	7388	7397	7406	7430	7439	7448	7477	7501	7524	7534
	8015	7566	7572	7592	7602	7666	7727	7750	7779	7818	7841	7915	8003	8007	8011
\$\$SERNU	431	141	150	161	170	203	222	241	259	294	313	323	347	367	391
	782	455	473	491	515	540	563	585	609	634	658	683	708	733	757
	1131	816	834	852	870	905	918	952	965	999	1012	1046	1059	1111	1122
	1464	1159	1177	1200	1212	1238	1251	1278	1293	1324	1340	1372	1388	1420	1436
	1730	1482	1510	1520	1535	1544	1567	1577	1591	1600	1623	1637	1646	1681	1695
	2030	1748	1783	1800	1827	1836	1847	1856	1867	1895	1911	1946	1960	1988	2000
	2305	2042	2066	2079	2084	2111	2120	2130	2156	2165	2189	2201	2227	2240	2273
	2635	2338	2370	2387	2409	2429	2462	2471	2487	2497	2497	2497	2518	2533	2603
	2930	2644	2678	2688	2721	2730	2763	2772	2799	2826	2856	2891	2904	2915	2955
	3219	2941	2972	2982	3007	3016	3026	3038	3048	3077	3089	3117	3145	3174	3209
	3533	3249	3277	3287	3317	3326	3355	3365	3392	3418	3444	3468	3477	3501	3510
	3751	3544	3553	3578	3587	3600	3610	3619	3648	3657	3671	3681	3707	3710	3749
	3958	3760	3771	3780	3789	3815	3824	3833	3858	3867	3876	3901	3910	3919	3949
	4177	3987	3998	4007	4034	4043	4055	4064	4073	4101	4110	4119	4129	4139	4168
	4384	4186	4215	4224	4236	4245	4254	4286	4295	4304	4315	4324	4355	4364	4373
	4710	4393	4423	4432	4441	4451	4460	4495	4533	4569	4600	4630	4645	4678	4693
	5065	4743	4760	4779	4808	4827	4836	4869	4905	4935	4965	4992	5001	5028	5056
	5373	5092	5120	5129	5159	5169	5202	5235	5289	5298	5312	5324	5333	5346	5359

	5373	5387	5427	5446	5465	5483	5501	5517	5533	5552	5577	5586	5621	5636	5658
	5673	5695	5709	5732	5747	5781	5797	5812	5835	5849	5879	5886	5912	5935	5950
	5972	5987	6018	6032	6046	6058	6073	6096	6111	6126	6176	6176	6192	6215	6230
	6246	6260	6283	6317	6332	6347	6362	6385	6400	6415	6430	6464	6479	6493	6509
	6532	6547	6586	6576	6599	6614	6628	6645	6668	6684	6699	6717	6752	6773	6807
	7086	7095	7116	7125	7146	7155	7182	7196	7220	7229	7238	7252	7271	7280	7304
	7313	7322	7346	7355	7364	7388	7397	7406	7430	7439	7448	7477	7501	7524	7534
	7541	7566	7572	7592	7602	7666	7727	7750	7779	7818	7841	7915	8003	8007	8011
\$\$ERRO	486	509	534	603	628	652	677	702	727	751	776	801	829	847	865
	912	959	1006	1053	1126	1171	1206	1245	1287	1334	1382	1430	1476	1539	1595
	1641	1689	1742	1794	1862	1909	1954	1994	2036	2079	2125	2160	2196	2234	2267
	2397	2432	2383	2424	2492	2536	2579	2634	2683	2725	2767	2794	2821	2860	2936
	2977	3043	3083	3112	3140	3174	3214	3244	3284	3321	3360	3387	3413	3439	3477
	3505	3548	3634	3676	3720	3784	3811	3841	3874	3923	3960	4002	4044	4081	4117
	4319	4388	4455	4489	4527	4564	4595	4639	4704	4773	4830	4863	4899	4959	4959
	4996	5023	5060	5087	5124	5163	5196	5229	5382	5546	5581	5630	5667	5703	5741
	5806	5873	5906	5944	5981	6067	6120	6186	6254	6277	6356	6424	6503	6570	6639
	6711	6767	6819	6863	6923	6968	6999	7030	7060	7090	7120	7150	7191	7233	7275
\$\$ESCA	471	148	168	201	220	239	257	292	311	321	344	365	388	435	453
\$\$LOOP	489	489	512	537	606	631	655	680	705	730	754	779	814	832	850
	868	902	915	949	962	996	1009	1043	1056	1109	1120	1129	1157	1174	1198
	1209	1236	1248	1278	1290	1327	1337	1370	1385	1418	1433	1462	1479	1507	1518
	1532	1547	1568	1576	1598	1629	1635	1644	1644	1697	1692	1728	1728	1781	1797
	1824	1834	1845	1854	1865	1893	1908	1944	1957	1986	1997	2028	2039	2063	2073
	2082	2108	2118	2128	2153	2163	2186	2199	2225	2237	2270	2302	2335	2367	2384
	2416	2427	2459	2469	2484	2495	2528	2539	2575	2590	2601	2632	2642	2675	2686
	2718	2758	2764	2770	2797	2824	2853	2889	2902	2902	2913	2928	2939	2969	2980
	3008	3014	3024	3036	3046	3074	3085	3115	3145	3172	3207	3217	3247	3275	3285
	3315	3334	3353	3363	3390	3416	3485	3495	3495	3498	3508	3530	3541		

4199	4268	4337	4405	4477	4515	4552	4583	4612	4659	4722	4791	4849	4884	4918
4947	4979	5013	5041	5078	5105	5144	5186	5219	5274	5409	5565	5604	5640	5677
5713	5763	5813	5894	5917	5954	6001	6078	6141	6197	6265	6299	6366	6445	6513
6580	6649	6732	6785	6836	6880	6943	6979	7009	7040	7070	7100	7130	7168	7201
7243	7285	7327	7369	7411	7461	7488	7513	7553	7577	7631	7688	7738	7761	7797
7848														

\$SSKIP
 \$EQUAT
 \$HEAD
 \$HTI
 \$SETUP
 \$SWRHI
 \$ACT1
 \$APT8
 \$APTH
 \$APTY
 \$ASTA
 \$CATC
 \$CMTA
 \$DB2D
 \$DB2D
 \$DIV
 \$EOP
 \$ERRR
 \$ERRT
 \$HULT
 \$POWE
 \$GRAND
 \$RDDE
 \$RDOC
 \$READ
 \$RAZ
 \$SAVE
 \$SB2D
 \$SB2D
 \$SCOP
 \$SITE
 \$SOPR
 \$TRAP
 \$TVPB
 \$TVPD
 \$TVE
 \$TVPD
 \$40CA
 \$1170

14#
14#
52

. ABS. 026524 000

ERRORS DETECTED: 0
 CFKAAC.BIN,CFKAAC.LST/CRF/SOL=CFKAAC.SML,CFKAAC.P11
 RUN-TIME: 30 40 3 SECONDS
 RUN-TIME RATIO: 170/74=2.2
 CORE USED: 33K (65 PAGES)