

LSI-11

MOVE/STRING INSTR TEST
MD-11-DVKAI-A

EP-DVKAI-A DL A
COPYRIGHT 1977
FICHE 1 OF 1

MAR 1977
digital
MADE IN USA

The main body of the document consists of a grid of 10 columns and 10 rows of small, illegible test data tables. Each table appears to be a structured list of data points, possibly representing test results for different memory addresses or instructions. The text is too small to be read, but the layout is consistent across the grid.

.REM !

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVKAI-A-D

PRODUCT NAME: DIS MOVE & STRING INSTRUCTION TEST

DATE CREATED: FEBRUARY 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: D.J. CASALETTO

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

00010000
22-DEC-76 10:47
MAC222427(1006) 22-DEC-76 10:49 PAGE 0010VKAIASEG
00010000
773224
SEG 0002

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURES

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES

2.2 STARTING PROCEDURE

LOAD THE SWITCH REGISTER WITH THE DESIRED SETTING
 (SOFTWARE SWITCH REGISTER LOCATION = 176)

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.
 STARTING AT 200, THE PROGRAM CLEARS ALL PROGRAM PARAMETERS AND
 THEN PRINTS ITS MAINDEC IDENTIFICATION. "END PASS" IS PRINTED
 AT THE END OF EACH FULL PASS OF THE DIAGNOSTIC.

2.3 OPERATING PROCEDURES

2.3.1 OPERATIONAL SWITCH REGISTER

LOCATION 176 IS USED FOR THE SOFTWARE SWITCH REGISTER AND
 THE FOLLOWING OPTIONS MAY BE SELECTED BY INSERTING A 1 IN THEIR
 RESPECTIVE BIT POSITIONS.

BIT15	- HALT ON ERROR
BIT14	+ SCOPE LOOP
BIT13	- INHIBIT ERROR TYPEOUT
BIT12	- INHIBIT TRACE TRAP
BIT11	- UNUSED
BIT10	- UNUSED
BIT09	- LOOP ON ERROR
BIT08	- LOOP ON TEST IN SWR<05:00>
BIT07	- INHIBIT INTERRUPTABILITY TESTS

NOTE: ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BITS OF BYTE SENVM
 HIGH.

2.3.2 RUNNING UNDER APT

 THE APT MAILBOX-E-TABLE IS LOCATED AT LOCATION 566.

USING THE CONSOLE INTERFACE AS THE INTERRUPTING DEVICE,
 THE INTERRUPTABILITY TESTS WILL BE RUN ON ONLY THE FIRST PASS TO
 AVOID INTERFERENCE WITH THE APT INTERFACE. IF INTERRUPTABILITY
 TESTS ARE DESIRED ON ALL PASSES, ANOTHER SLL MUST BE SUPPLIED
 AND ITS RECEIVER STATUS REGISTER ADDRESS & ITS INTERRUPT VECTOR MUST
 BE PLACED IN THE APT E-TABLE AT LOCATIONS "\$BASE" & "\$VECT1" RESPECTIVELY.

2.3.3 RUN WITH ALTERNATE CONSOLE ADDRESS

 TO USE A CONSOLE ADDRESS OTHER THAN 177560, THE OPERATOR
 MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM
 AT THE LOCATIONS LABELED:

\$TKS: RCSR ADDRESS
 \$TKB: RBLF ADDRESS
 \$TPS: TCSR ADDRESS
 \$TPB: TBUF ADDRESS

2.3.4 RUN INTERRUPT TESTS WITH ALTERNATE SLU

 TO USE A SERIAL LINE INTERFACE ADDRESS OTHER THAN THE STANDARD
 CONSOLE ADDRESS (177560), THE OPERATOR MUST SUPPLY THE CORRECT ADDRESS
 AND INTERRUPT VECTOR BY INSERTING THEM IN THE LOCATIONS LABELED:

\$BASE: *RCSR ADDRESS*
 \$VECT1: *RECEIVER INTERRUPT VECTOR*

177560
 177561
 177562
 177563
 177564
 177565
 177566
 177567
 177568
 177569
 177570
 177571
 177572
 177573
 177574
 177575
 177576
 177577
 177578
 177579
 177580
 177581
 177582
 177583
 177584
 177585
 177586
 177587
 177588
 177589
 177590
 177591
 177592
 177593
 177594
 177595
 177596
 177597
 177598
 177599
 177600
 177601
 177602
 177603
 177604
 177605
 177606
 177607
 177608
 177609
 177610
 177611
 177612
 177613
 177614
 177615
 177616
 177617
 177618
 177619
 177620
 177621
 177622
 177623
 177624
 177625
 177626
 177627
 177628
 177629
 177630
 177631
 177632
 177633
 177634
 177635
 177636
 177637
 177638
 177639
 177640
 177641
 177642
 177643
 177644
 177645
 177646
 177647
 177648
 177649
 177650
 177651
 177652
 177653
 177654
 177655
 177656
 177657
 177658
 177659
 177660
 177661
 177662
 177663
 177664
 177665
 177666
 177667
 177668
 177669
 177670
 177671
 177672
 177673
 177674
 177675
 177676
 177677
 177678
 177679
 177680
 177681
 177682
 177683
 177684
 177685
 177686
 177687
 177688
 177689
 177690
 177691
 177692
 177693
 177694
 177695
 177696
 177697
 177698
 177699
 177700
 177701
 177702
 177703
 177704
 177705
 177706
 177707
 177708
 177709
 177710
 177711
 177712
 177713
 177714
 177715
 177716
 177717
 177718
 177719
 177720
 177721
 177722
 177723
 177724
 177725
 177726
 177727
 177728
 177729
 177730
 177731
 177732
 177733
 177734
 177735
 177736
 177737
 177738
 177739
 177740
 177741
 177742
 177743
 177744
 177745
 177746
 177747
 177748
 177749
 177750
 177751
 177752
 177753
 177754
 177755
 177756
 177757
 177758
 177759
 177760
 177761
 177762
 177763
 177764
 177765
 177766
 177767
 177768
 177769
 177770
 177771
 177772
 177773
 177774
 177775
 177776
 177777
 177778
 177779
 177780
 177781
 177782
 177783
 177784
 177785
 177786
 177787
 177788
 177789
 177790
 177791
 177792
 177793
 177794
 177795
 177796
 177797
 177798
 177799
 177800
 177801
 177802
 177803
 177804
 177805
 177806
 177807
 177808
 177809
 177810
 177811
 177812
 177813
 177814
 177815
 177816
 177817
 177818
 177819
 177820
 177821
 177822
 177823
 177824
 177825
 177826
 177827
 177828
 177829
 177830
 177831
 177832
 177833
 177834
 177835
 177836
 177837
 177838
 177839
 177840
 177841
 177842
 177843
 177844
 177845
 177846
 177847
 177848
 177849
 177850
 177851
 177852
 177853
 177854
 177855
 177856
 177857
 177858
 177859
 177860
 177861
 177862
 177863
 177864
 177865
 177866
 177867
 177868
 177869
 177870
 177871
 177872
 177873
 177874
 177875
 177876
 177877
 177878
 177879
 177880
 177881
 177882
 177883
 177884
 177885
 177886
 177887
 177888
 177889
 177890
 177891
 177892
 177893
 177894
 177895
 177896
 177897
 177898
 177899
 177900
 177901
 177902
 177903
 177904
 177905
 177906
 177907
 177908
 177909
 177910
 177911
 177912
 177913
 177914
 177915
 177916
 177917
 177918
 177919
 177920
 177921
 177922
 177923
 177924
 177925
 177926
 177927
 177928
 177929
 177930
 177931
 177932
 177933
 177934
 177935
 177936
 177937
 177938
 177939
 177940
 177941
 177942
 177943
 177944
 177945
 177946
 177947
 177948
 177949
 177950
 177951
 177952
 177953
 177954
 177955
 177956
 177957
 177958
 177959
 177960
 177961
 177962
 177963
 177964
 177965
 177966
 177967
 177968
 177969
 177970
 177971
 177972
 177973
 177974
 177975
 177976
 177977
 177978
 177979
 177980
 177981
 177982
 177983
 177984
 177985
 177986
 177987
 177988
 177989
 177990
 177991
 177992
 177993
 177994
 177995
 177996
 177997
 177998
 177999
 178000

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
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

- 4.5 NPREP

THIS ROUTINE IS USED TO STORE A COPY OF THE INSTRUCTION TEST ARGUMENTS TO BE STORED IN RC-->R5.
- 4.6 GENR

THIS ROUTINE IS USED TO TRANSFER INSTRUCTION TEST ARGUMENTS TO THE GENERAL REGISTERS AND TO COPY THE STACK POINTER BEFORE THE TEST INSTRUCTION EXECUTION.
- 4.7 XPSW

THIS ROUTINE IS USED TO STORED THE EXPECTED PSW OF THE INSTRUCTION TEST AND TO SET THE T-BIT IN THE EXPECTED PSW ON PASSES USING THE TRACE TRAP.
- 4.9 INTR

THIS ROUTINE IS USED TO DETECT WHEN THE TEST INSTRUCTION HAS BEEN INTERRUPTED AND TO CONTINUE THE INTERRUPT STREAM UNTIL THE INSTRUCTION IS INTERRUPTED.

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200
000201
000202
000203
000204
000205
000206
000207
000208
000209
000210
000211
000212
000213
000214
000215
000216
000217
000218
000219
000220
000221
000222
000223
000224
000225
000226
000227
000228
000229
000230
000231
000232
000233
000234
000235
000236
000237
000238
000239
000240
000241
000242
000243
000244
000245
000246
000247
000248
000249
000250
000251
000252
000253
000254
000255
000256
000257
000258
000259
000260
000261
000262
000263
000264
000265
000266
000267
000268
000269
000270
000271
000272
000273
000274
000275
000276
000277
000278
000279
000280
000281
000282
000283
000284
000285
000286
000287
000288
000289
000290
000291
000292
000293
000294
000295
000296
000297
000298
000299
000300
000301
000302
000303
000304
000305
000306
000307
000308
000309
000310
000311
000312
000313
000314
000315
000316
000317
000318
000319
000320
000321
000322
000323
000324
000325
000326
000327
000328
000329
000330
000331
000332
000333
000334
000335
000336
000337
000338
000339
000340
000341
000342
000343
000344
000345
000346
000347
000348
000349
000350
000351
000352
000353
000354
000355
000356
000357
000358
000359
000360
000361
000362
000363
000364
000365
000366
000367
000368
000369
000370
000371
000372
000373
000374
000375
000376
000377
000378
000379
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
000408
000409
000410
000411
000412
000413
000414
000415
000416
000417
000418
000419
000420
000421
000422
000423
000424
000425
000426
000427
000428
000429
000430
000431
000432
000433
000434
000435
000436
000437
000438
000439
000440
000441
000442
000443
000444
000445
000446
000447
000448
000449
000450
000451
000452
000453
000454
000455
000456
000457
000458
000459
000460
000461
000462
000463
000464
000465
000466
000467
000468
000469
000470
000471
000472
000473
000474
000475
000476
000477
000478
000479
000480
000481
000482
000483
000484
000485
000486
000487
000488
000489
000490
000491
000492
000493
000494
000495
000496
000497
000498
000499
000500
000501
000502
000503
000504
000505
000506
000507
000508
000509
000510
000511
000512
000513
000514
000515
000516
000517
000518
000519
000520
000521
000522
000523
000524
000525
000526
000527
000528
000529
000530
000531
000532
000533
000534
000535
000536
000537
000538
000539
000540
000541
000542
000543
000544
000545
000546
000547
000548
000549
000550
000551
000552
000553
000554
000555
000556
000557
000558
000559
000560
000561
000562
000563
000564
000565
000566
000567
000568
000569
000570
000571
000572
000573
000574
000575
000576
000577
000578
000579
000580
000581
000582
000583
000584
000585
000586
000587
000588
000589
000590
000591
000592
000593
000594
000595
000596
000597
000598
000599
000600
000601
000602
000603
000604
000605
000606
000607
000608
000609
000610
000611
000612
000613
000614
000615
000616
000617
000618
000619
000620
000621
000622
000623
000624
000625
000626
000627
000628
000629
000630
000631
000632
000633
000634
000635
000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660
000661
000662
000663
000664
000665
000666
000667
000668
000669
000670
000671
000672
000673
000674
000675
000676
000677
000678
000679
000680
000681
000682
000683
000684
000685
000686
000687
000688
000689
000690
000691
000692
000693
000694
000695
000696
000697
000698
000699
000700
000701
000702
000703
000704
000705
000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
000716
000717
000718
000719
000720
000721
000722
000723
000724
000725
000726
000727
000728
000729
000730
000731
000732
000733
000734
000735
000736
000737
000738
000739
000740
000741
000742
000743
000744
000745
000746
000747
000748
000749
000750
000751
000752
000753
000754
000755
000756
000757
000758
000759
000760
000761
000762
000763
000764
000765
000766
000767
000768
000769
000770
000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880
000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990
000991
000992
000993
000994
000995
000996
000997
000998
000999
001000

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

001100

STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS

000011
000012
000015
000200
177776

177774
177772
177570
177570

HT= 11 ;;CODE FOR HORIZONTAL TAB
LF= 12 ;;CODE FOR LINE FEED
CR= 15 ;;CODE FOR CARRIAGE RETURN
CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776 ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570 ;;HARDWARE SWITCH REGISTER
DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

R0= %0 ;;GENERAL REGISTER
R1= %1 ;;GENERAL REGISTER
R2= %2 ;;GENERAL REGISTER
R3= %3 ;;GENERAL REGISTER
R4= %4 ;;GENERAL REGISTER
R5= %5 ;;GENERAL REGISTER
R6= %6 ;;GENERAL REGISTER
R7= %7 ;;GENERAL REGISTER
SP= %6 ;;STACK POINTER
PC= %7 ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS

000000
000040
000100
000140
000200
000240
000300
000340

PR0= 0 ;;PRIORITY LEVEL 0
PR1= 40 ;;PRIORITY LEVEL 1
PR2= 100 ;;PRIORITY LEVEL 2
PR3= 140 ;;PRIORITY LEVEL 3
PR4= 200 ;;PRIORITY LEVEL 4
PR5= 240 ;;PRIORITY LEVEL 5
PR6= 300 ;;PRIORITY LEVEL 6
PR7= 340 ;;PRIORITY LEVEL 7

;*SWITCH REGISTER SWITCH DEFINITIONS

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010

SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20

00000000
00000001
00000010
00000011
00000100
00000101
00000110
00000111
00001000
00001001
00001010
00001011
00001100
00001101
00001110
00001111
00010000
00010001
00010010
00010011
00010100
00010101
00010110
00010111
00011000
00011001
00011010
00011011
00011100
00011101
00011110
00011111
00100000
00100001
00100010
00100011
00100100
00100101
00100110
00100111
00101000
00101001
00101010
00101011
00101100
00101101
00101110
00101111
00110000
00110001
00110010
00110011
00110100
00110101
00110110
00110111
00111000
00111001
00111010
00111011
00111100
00111101
00111110
00111111
01000000
01000001
01000010
01000011
01000100
01000101
01000110
01000111
01001000
01001001
01001010
01001011
01001100
01001101
01001110
01001111
01010000
01010001
01010010
01010011
01010100
01010101
01010110
01010111
01011000
01011001
01011010
01011011
01011100
01011101
01011110
01011111
01100000
01100001
01100010
01100011
01100100
01100101
01100110
01100111
01101000
01101001
01101010
01101011
01101100
01101101
01101110
01101111
01110000
01110001
01110010
01110011
01110100
01110101
01110110
01110111
01111000
01111001
01111010
01111011
01111100
01111101
01111110
01111111
10000000
10000001
10000010
10000011
10000100
10000101
10000110
10000111
10001000
10001001
10001010
10001011
10001100
10001101
10001110
10001111
10010000
10010001
10010010
10010011
10010100
10010101
10010110
10010111
10011000
10011001
10011010
10011011
10011100
10011101
10011110
10011111
10100000
10100001
10100010
10100011
10100100
10100101
10100110
10100111
10101000
10101001
10101010
10101011
10101100
10101101
10101110
10101111
10110000
10110001
10110010
10110011
10110100
10110101
10110110
10110111
10111000
10111001
10111010
10111011
10111100
10111101
10111110
10111111
11000000
11000001
11000010
11000011
11000100
11000101
11000110
11000111
11001000
11001001
11001010
11001011
11001100
11001101
11001110
11001111
11010000
11010001
11010010
11010011
11010100
11010101
11010110
11010111
11011000
11011001
11011010
11011011
11011100
11011101
11011110
11011111
11100000
11100001
11100010
11100011
11100100
11100101
11100110
11100111
11101000
11101001
11101010
11101011
11101100
11101101
11101110
11101111
11110000
11110001
11110010
11110011
11110100
11110101
11110110
11110111
11111000
11111001
11111010
11111011
11111100
11111101
11111110
11111111

00000000
00000001
00000010
00000011
00000100
00000101
00000110
00000111
00001000
00001001
00001010
00001011
00001100
00001101
00001110
00001111
00010000
00010001
00010010
00010011
00010100
00010101
00010110
00010111
00011000
00011001
00011010
00011011
00011100
00011101
00011110
00011111
00100000
00100001
00100010
00100011
00100100
00100101
00100110
00100111
00101000
00101001
00101010
00101011
00101100
00101101
00101110
00101111
00110000
00110001
00110010
00110011
00110100
00110101
00110110
00110111
00111000
00111001
00111010
00111011
00111100
00111101
00111110
00111111
01000000
01000001
01000010
01000011
01000100
01000101
01000110
01000111
01001000
01001001
01001010
01001011
01001100
01001101
01001110
01001111
01010000
01010001
01010010
01010011
01010100
01010101
01010110
01010111
01011000
01011001
01011010
01011011
01011100
01011101
01011110
01011111
01100000
01100001
01100010
01100011
01100100
01100101
01100110
01100111
01101000
01101001
01101010
01101011
01101100
01101101
01101110
01101111
01110000
01110001
01110010
01110011
01110100
01110101
01110110
01110111
01111000
01111001
01111010
01111011
01111100
01111101
01111110
01111111
10000000
10000001
10000010
10000011
10000100
10000101
10000110
10000111
10001000
10001001
10001010
10001011
10001100
10001101
10001110
10001111
10010000
10010001
10010010
10010011
10010100
10010101
10010110
10010111
10011000
10011001
10011010
10011011
10011100
10011101
10011110
10011111
10100000
10100001
10100010
10100011
10100100
10100101
10100110
10100111
10101000
10101001
10101010
10101011
10101100
10101101
10101110
10101111
10110000
10110001
10110010
10110011
10110100
10110101
10110110
10110111
10111000
10111001
10111010
10111011
10111100
10111101
10111110
10111111
11000000
11000001
11000010
11000011
11000100
11000101
11000110
11000111
11001000
11001001
11001010
11001011
11001100
11001101
11001110
11001111
11010000
11010001
11010010
11010011
11010100
11010101
11010110
11010111
11011000
11011001
11011010
11011011
11011100
11011101
11011110
11011111
11100000
11100001
11100010
11100011
11100100
11100101
11100110
11100111
11101000
11101001
11101010
11101011
11101100
11101101
11101110
11101111
11110000
11110001
11110010
11110011
11110100
11110101
11110110
11110111
11111000
11111001
11111010
11111011
11111100
11111101
11111110
11111111

SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC=14 ;: "T" BIT
TRTVEC= 14 ;: TRACE TRAP
BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWRVEC= 24 ;: POWER FAIL
EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
TRAPVEC=34 ;: "TRAP" TRAP
TKVEC= 60 ;: TTY KEYBOARD VECTOR
TPVEC= 64 ;: TTY PRINTER VECTOR
PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR

MO1

SEQ 0013

000200
000001
000100
000040
171400
000300
000020
000001
000001
177777
177560
000060
000000

APTSIZE= 200
APTENV= 001
APTPOOL= 100
APCSUP= 040
\$SWR= 171400
\$SWRMK= 300
TBIT= 20
\$TN= 1
N= 1
NXM= 177777
REASE= 177560
AVECT1= 60
.=C

```

480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
    .SBTTL TRAP CATCHER

    000000
    ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
    ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
    ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

    000174 000000    ;=174
    DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
    SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER

    000000 015106    ;=0
    TZERO          ;SET LOCATIONS 0,4,6 TO ERROR REPORT
    340
    TIMTRP
    340
    ILLTRP
    340

    000100 000102    ;=100
    000106 000102    .WORD 102      ;HANDLE EVENT LINE INTERRUPTS
    000102 000002    .WORD 2

    000200 000167 000512    ;=200
    JMP START          ;STARTING ADDRESS OF PROGRAM
    
```

508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542

000400

.SBTTL .=400
ACT11 HOOKS

HOOKS REQUIRED BY ACT11

000400
000046
015010
000052
000000
000400

\$SVPC=. ;SAVE PC
.=46
\$ENDAD ;:1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .SEOF
.=52
.WORD 0 ;:2)SET LOC.52 TO ZERO
.= \$SVPC ;: RESTORE PC

.SBTTL APT PARAMETER BLOCK

SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT

000400
000024
000200
000044
000400
000400

.\$X=. ;:SAVE CURRENT LOCATION
.=24 ;:SET POWER FAIL TO POINT TO START OF PROGRAM
200 ;:FOR APT START UP
.=44 ;:POINT TO APT INDIRECT ADDRESS PNTR.
\$APTHDR ;:POINT TO APT HEADER BLOCK
.= \$X ;:RESET LOCATION COUNTER

SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
INTERFACE SPEC.

\$APTHD:
\$HIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
\$MBADR: .WORD \$MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
\$TSTM: .WORD 30 ;:RUN TIM OF LONGEST TEST
\$PASTM: .WORD 40 ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
\$UNITM: .WORD ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
\$.WORD \$END-\$MAIL/2 ;:LENGTH MAILBOX-ETABLE.WORDS

CO2

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 15
CVKATA.P11 22-DEC-76 10:47 APT PARAMETER BLOCK

SEQ 0016

543

544
545
546
547
548
549
550
551 000500
552 000500 000000
553 000502 000
554 000503 000
555 000504 000000
556 000506 000000
557 000510 000000
558 000512 000000
559 000514 000
560 000515 001
561 000516 000000
562 000520 000000
563 000522 000000
564 000524 000000
565 000526 000000
566 000530 000000
567 000532 000000
568 000534 000
569 000535 000
570 000536 000000
571 000540 177570
572 000542 177570
573 000544 177560
574 000546 177562
575 000550 177564
576 000552 177566
577 000554 000
578 000555 002
579 000556 012
580 000557 000
581 000560 000000
582 000562 077
583 000563 015
584 000564 000012
585
586
587
588
589
590 000566
591 000566 000000
592 000570 000000
593 000572 000000
594 000574 000000
595 000576 000000
596 000600 000000
597 000602 000000
598 000604 000000
599 000606

.SBTTL COMMON TAGS

::*****
:*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
:*USED IN THE PROGRAM.

SCMTAG: . =500
:: START OF COMMON TAGS
\$TSTNM: .WORD 0
:: CONTAINS THE TEST NUMBER
\$ERFLG: .BYTE 0
:: CONTAINS ERROR FLAG
\$ICNT: .WORD 0
:: CONTAINS SUBTEST ITERATION COUNT
\$LPADR: .WORD 0
:: CONTAINS SCOPE LOOP ADDRESS
\$LPERR: .WORD 0
:: CONTAINS SCOPE RETURN FOR ERRORS
\$ERTTL: .WORD 0
:: CONTAINS TOTAL ERRORS DETECTED
\$ITEMB: .BYTE 0
:: CONTAINS ITEM CONTROL BYTE
\$ERMAX: .BYTE 1
:: CONTAINS MAX. ERRORS PER TEST
\$ERRPC: .WORD 0
:: CONTAINS PC OF LAST ERROR INSTRUCTION
\$GDADR: .WORD 0
:: CONTAINS ADDRESS OF 'GOOD' DATA
\$BDADR: .WORD 0
:: CONTAINS ADDRESS OF 'BAD' DATA
\$GDDAT: .WORD 0
:: CONTAINS 'GOOD' DATA
\$BDDAT: .WORD 0
:: CONTAINS 'BAD' DATA
:: RESERVED--NOT TO BE USED
\$AUTOB: .BYTE 0
:: AUTOMATIC MODE INDICATOR
\$INTAG: .BYTE 0
:: INTERRUPT MODE INDICATOR
\$SWR: .WORD DSWR
:: ADDRESS OF SWITCH REGISTER
\$DISPLAY: .WORD DDISP
:: ADDRESS OF DISPLAY REGISTER
\$TKS: 177560
:: TTY KBD STATUS
\$TKB: 177562
:: TTY KBD BUFFER
\$TPS: 177564
:: TTY PRINTER STATUS REG. ADDRESS
\$TPB: 177566
:: TTY PRINTER BUFFER REG. ADDRESS
\$NULL: .BYTE 0
:: CONTAINS NULL CHARACTER FOR FILLS
\$FILLS: .BYTE 2
:: CONTAINS # OF FILLER CHARACTERS REQUIRED
\$FILLC: .BYTE 12
:: INSERT FILL CHARS. AFTER A "LINE FEED"
\$TPFLG: .BYTE 0
:: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
\$ESCAPE: 0
:: ESCAPE ON ERROR ADDRESS
\$QUES: .ASCII /?/
:: QUESTION MARK
\$CRLF: .ASCII <15>
:: CARRIAGE RETURN
\$LF: .ASCII <12>
:: LINE FEED

::*****
.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
\$PASS: .WORD APASS :: PASS COUNT
\$DEVCT: .WORD ADEVCT :: DEVICE COUNT
\$UNIT: .WORD AUNIT :: I/O UNIT NUMBER
\$MSGAD: .WORD AMSGAD :: MESSAGE ADDRESS
\$MSGLG: .WORD AMSLG :: MESSAGE LENGTH
\$E*TABLE: :: APT ENVIRONMENT TABLE

600	000606	000	\$ENV:	.BYTE	AENV	:: ENVIRONMENT BYTE
601	000607	000	\$ENVM:	.BYTE	AENVM	:: ENVIRONMENT MODE BITS
602	000610	000000	\$SWREG:	.WORD	ASWREG	:: APT SWITCH REGISTER
603	000612	000000	\$USWR:	.WORD	AUSWR	:: USER SWITCHES
604	000614	000000	\$CPUOP:	.WORD	ACPUOP	:: CPU TYPE, OPTIONS
605			.*			BITS 15-11=CPU TYPE
606			.*			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
607			.*			11/70=06, PDQ=07, Q=10
608			.*			BIT 10=REAL TIME CLOCK
609			.*			BIT 9=FLOATING POINT PROCESSOR
610			.*			BIT 8=MEMORY MANAGEMENT
611	000616	000	\$MAMS1:	.BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
612	000617	000	\$MTYP1:	.BYTE	AMTYP1	:: MEM. TYPE, BLK#1
613			.*			MEM. TYPE BYTE -- (HIGH BYTE)
614			.*			900 NSEC CORE=001
615			.*			300 NSEC BIPOLAR=002
616			.*			500 NSEC MOS=003
617	000620	000000	\$MADR1:	.WORD	AMADR1	:: HIGH ADDRESS, BLK#1
618			.*			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
619	000622	000	\$MAMS2:	.BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
620	000623	000	\$MTYP2:	.BYTE	AMTYP2	:: MEM. TYPE, BLK#2
621	000624	000000	\$MADR2:	.WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
622	000626	000	\$MAMS3:	.BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
623	000627	000	\$MTYP3:	.BYTE	AMTYP3	:: MEM. TYPE, BLK#3
624	000630	000000	\$MADR3:	.WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
625	000632	000	\$MAMS4:	.BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
626	000633	000	\$MTYP4:	.BYTE	AMTYP4	:: MEM. TYPE, BLK#4
627	000634	000000	\$MADR4:	.WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
628	000636	000000	\$VECT1:	.WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
629	000640	000000	\$VECT2:	.WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
630	000642	177560	\$BASE:	.WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
631	000644		\$*END:			
632			.MEXIT			

633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671

.SBTTL ERROR POINTER TABLE

:*THIS TABLE CONTAINS THE INFORMATIC FOR EACH ERROR THAT CAN OCCUR.
:*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
:*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
:*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
:*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

:* EM ;;POINTS TO THE ERROR MESSAGE
:* DH ;;POINTS TO THE DATA HEADER
:* DT ;;POINTS TO THE DATA
:* DF ;;POINTS TO THE DATA FORMAT

000644

\$ERRTB:

SRCLN: .WORD 0 ; SOURCE LENGTH
SRCAD: .WORD ; SOURCE ADDRESS
DSTLN: .WORD ; DESTINATION LENGTH
DSTAD: .WORD ; DESTINATION ADDRESS
FILL: .WORD ; FILL CHARACTER
TABLE: .WORD 177777 ; TRANSLATION TABLE ADDRESS
TCSR: 177564 ; TCSR ADDRESS OF SLU USED FOR INTERRUPTS
TBUF: 177566 ; TBUF ADDRESS
TVECT: 64 ; TRANSMIT INTERRUPT VECTOR
TPSW: 66 ; AND PSW LOCATION
PCI: 0 ; ADDRESS OF TEST INSTRUCTION TO INTERRUPT
CCODES: 0 ; CONDITION CODES AFTER TEST INSTRUCTION EXECUTION
EXPPSW: 0 ; EXPECTED CONDITION CODES
SAVR6: 0 ; STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
BADR6: 0 ; BAD STACK POINTER VALUE
OLDPC: 0 ; PC WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED
TEMP: 0
TEMP1: 0
TEMP2: 0
ONES: .WORD -1
ONEBYT: 377

000644 000000
000646 000000
000650 000000
000652 000000
000654 000000
000656 177777
000660 177564
000662 177566
000664 000064
000666 000066
000670 000000
000672 000000
000674 000000
000676 000000
000700 000000
000702 000000
000704 000000
000706 000000
000710 000000
000712 177777
000714 000377

728	001250	005720		TST	(R0)+	:ADJUST TO TBUF ADDRESS
729	001252	010037	000662	MOV	RC, @#TBUF	:STORE TBUF ADDRESS
730	001256	013700	000636	MOV	@#\$VECT1, R0	:GET SLU INTERRUPT VECTOR
731	001262	062700	000004	ADD	#4, R0	:ADJUST TO TRANSMIT INTERRUPT VECTOR
732	001266	010037	000664	MOV	R0, @#TVECT	:STORE TRANSMIT INTERRUPT VECTOR
733	001272	005720		TST	(R0)+	:ADJUST TO TRANSMIT INTERRUPT PSW
734	001274	010037	000666	MOV	R0, @#TPSW	:STORE TRANSMIT INTERRUPT PSW LOCATION
735	001300	012701	016434	PATGEN: MOV	#BUF1, R1	:GENERATE TEST PATTERN IN BUF1
736	001304	012700	000001	MOV	#1, R0	: THE PATTERN IS A BINARY COUNT ALTERNATED
737	001310	110021		1\$: MCVB	RC, (R1)+	: WITH ITS COMPLEMENT
738	001312	105100		COMB	R0	: 001,376,002,375,003,374.....177,200,200,177
739	001314	110021		MOVB	R0, (R1)+	
740	001316	105100		COMB	R0	
741	001320	105200		INCB	R0	: INCREMENT PATTERN
742	001322	020027	000200	CMP	RC, #200	: IS PATTERN FINISHED?
743	001326	003770		BLE	1\$: BR IF NOT
744						
745	001330	106427	000200	BEGIN: MTPS	#200	: SET PRIORITY TO 7

746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801

001334 000004
001336 004567 01365C
001342 000000
001344 177777
001346 000005
001350 017035
001352 000377
001354 004767 013660
001360 004567 013722
001364 000211
001366 004767 013554
001372 000251
001374 000266
001376 076030
001400 106767 177266
001404 042767 177400 177260
001412 023767 000674 177252
001420 001401
001422 104001
001424 023706 000676
001430 001403
001432 010637 000700
001436 104002
001440 005700
001442 001401
001444 104003
001446 005701
001450 001401
001452 104004
001454 005702
001456 001401
001460 104005

```

:*****
:*TEST 1 TEST "MOVC" INSTRUCTION WITH ZERO SOURCE LENGTH
:*SUCCESSFUL COMPLETION OF THIS TEST RESULTS IN
:*"FILL" ("377") BYTES WRITTEN THROUGHOUT THE DESTINATION
:*FIELD AND CONDITION CODES-->N,C=1 & Z,V=0
:*****
+ST1: SCOPE
      JSR    RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
      0          ;SOURCE LENGTH
      NXM       ;SOURCE ADDRESS
      5          ;DESTINATION LENGTH
      BUF2+1    ;DESTINATION ADDRESS
      377       ;FILL CHARACTER
      JSR    PC,CLDST    ;CLEAR DESTINATION
      JSR    RS,XPSW     ;STORE EXPECTED PSW VALUE
      .WORD    211
      JSR    PC,GENR     ;SET UP GENERAL REGISTERS
      +CLN!CLC          ;CLEAR CONDITION CODES N & C
      +SEZ!SEV          ;SET CONDITION CODES Z & V
                        ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MOVC
                        ;CHECK RESULTS
      MFPS    CCODES     ;STORE THE PSW
      BIC    #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP    @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ    64$         ;BR, IF EQUAL
      ERROR  1          ;*****TEST 1 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT "EXPPSW"
                        ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP    @#SAVR6,SP  ;VERIFY STACK POINTER IS RESTORED
      BEQ    65$         ;BR IF OK
      MOV    SP,@#BADR6 ;STORE BAD SP VALUE
      ERROR  2          ;*****TEST 1 - ERROR 2*****
                        ;STACK POINTER NOT RESTORED BY INSTRUCTION
                        ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                        ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      TST    R0          ;CHECK R0 FOR ZERO
      BEQ    1$         ;BR, IF ZERO
      ERROR  3          ;*****TEST 1 - ERROR 3*****
                        ;R0 SHOULD BE ZERO
      1$:
      TST    R1          ;CHECK STATE OF OTHER GENERAL REGISTERS
      BEQ    66$         ;TEST R1
      ERROR  4          ;BR, IF ZERO
                        ;*****TEST 1 - ERROR 4*****
                        ;R1 SHOULD BE ZERO
      66$:
      TST    R2          ;TEST R2
      BEQ    67$         ;BR IF ZERO
      ERROR  5          ;*****TEST 1 - ERROR 5*****
                        ;R2 SHOULD BE ZERO

```

```

802 001462 005703      67S:  TST      R3      ;TEST R3
803 001464 001401      BEQ      68S      ;BR, IF ZERO
804 001466 104006      ERROR    6        ;*****TEST 1 - ERROR 5*****
805                                     ;R3 SHOULD BE ZERO
806 001470                                     68S:
807 001470 026704 177150  CMP      FILL,R4    ;CHECK R4 UNCHANGED
808 001474 001401      BEQ      69S      ;BR IF OK
809 001476 104007      ERROR    7        ;*****TEST 1 - ERROR 7*****
810                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
811 001500 026705 177152 69S:  CMP      TABLE,R5 ;CHECK R5 UNCHANGED
812 001504 001401      BEQ      70S      ;BR IF OK
813 001506 104010      ERROR    10       ;*****TEST 1 - ERROR 10*****
814                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
815 001510      70S:
816                                     ;VERIFY DESTINATION CONTENTS
817 001510 012700 017034  MOV      #BJF2,R0   ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
818 001514 105720      TSTB    (R0)+      ;TEST CONTENTS OF BOUNDARY
819 001516 001401      BEQ      71S      ;BR, IF STILL ZERO
820 001520 104011      ERROR    11       ;*****TEST 1 - ERROR 11*****
821                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
822                                     ; SHOULD STILL EQUAL ZERO
823 001522      71S:
824 001522 016701 177122  MOV      DSTLN,R1  ;STORE TRANSFER BYTE COUNT IN R1
825 001526 122067 177122  TIE12: CMPB    (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
826 001532 001401      BEQ      +4        ;BR IF OK
827 001534 104012      ERROR    12       ;*****TEST 1 - ERROR 12*****
828                                     ;COMPARE ERROR IN DESTINATION
829                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
830
831 001536 005301      DEC      R1        ;DECREMENT BYTE COUNT
832 001540 001372      BNE     TIE12     ;BR, IF NOT FINISHED CHECKING
833 001542 105720      TSTB    (R0)+      ;TEST CONTENTS OF DEST. UPPER BOUNDARY
834 001544 001401      BEQ      ENDT1    ;BR, IF STILL ZERO
835 001546 104013      ERROR    13       ;*****TEST 1 - ERROR 13*****
836                                     ;UPPER BOUNDARY OF DEST. CHANGED
837                                     ; SHOULD STILL EQUAL ZERO
838 001550      ENDT1:
    
```

039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094

001550 000004
001552 012767 000712 177066
001560 004567 013426
001564 000001
001566 177777
001570 000000
001572 017035
001574 000377

001576 012700 017034
001602 005020
001604 005020
001606 004567 013474
001612 000200
001614 004767 013326
001620 000277

001622 076030

001624 106767 177042
001630 042767 177400 177034
001636 023767 000674 177026
001644 001401
001646 104001

001650
001650 023706 000676
001654 001403
001656 010637 000700
001662 104002

001664
001664 020067 176754
001670 001401
001672 104003

001674
001674 005701
001676 001401
001700 104004

```
*****  
:TEST 2 TEST "MOVC" INSTRUCTION WITH ZERO DESTINATION LENGTH  
:*****  
:SUCCESSFUL COMPLETION OF THIS TEST RESULTS IN  
:NO CHARACTERS TRANSFERED TO THE DESTINATION  
:AREA AND ALL CONDITION CODES CLEAR  
:*****  
:*****  
TST2: SCOPE  
MOV #ONES, SRCAD ;SET SOURCE ADDRESS  
JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS  
1 ;SOURCE LENGTH  
NXM ;SOURCE ADDRESS  
0 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
;CLEAR DESTINATION  
MOV #BUF2, R0 ;POINT R0 TO LOWER BYTE BOUNDARY  
CLR (R0)+ ;CLEAR BOUNDARY & DEST. BYTES  
CLR (R0)+ ;CLEAR UPPER BYTE BOUNDARY  
JSR R5, XPSW ;STORE EXPECTED PSW VALUE  
.WORD 200  
JSR PC, GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES  
;EXECUTE "MOVE CHARACTER" INSTRUCTION  
  
;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #177400, CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 2 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
  
64$:  
CMP @#SAVR6, SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP, @#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 2 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
  
65$:  
CMP R0, SRCLN ;CHECK R0 EQUAL TO SOURCE LENGTH  
BEQ 1$ ;BR, IF EQUAL  
ERROR 3 ;*****TEST 2 - ERROR 3*****  
;R0 SHOULD EQUAL SOURCE LENGTH  
;CHECK OTHER GENERAL REGISTERS  
  
1$:  
TST R1 ;TEST R1  
BEQ 66$ ;BR, IF ZERO  
ERROR 4 ;*****TEST 2 - ERROR 4*****  
;R1 SHOULD BE ZERO
```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 24
 CVKARA.P11 22-DEC-76 10:47 T2

TEST "MOVC" INSTRUCTION WITH ZERO DESTINATION LENGTH

SEQ 0025

```

895 001702 005702      66$:  TST      R2      ;TEST R2
896 001704 001401      BEQ      67$      ;BR IF ZERO
897 001706 104005      ERROR    5        ;*****TEST 2 - ERROR 5*****
898                                     ;R2 SHOULD BE ZERO
899 001710 005703      67$:  TST      R3      ;TEST R3
900 001712 001401      BEQ      68$      ;BR, IF ZERO
901 001714 104006      ERROR    6        ;*****TEST 2 - ERROR 6*****
902                                     ;R3 SHOULD BE ZERO
903
904 001716 026704 176732 68$:  CMP      FILL,R4   ;CHECK R4 UNCHANGED
905 001722 001401      BEQ      69$      ;BR IF OK
906 001724 104007      ERROR    7        ;*****TEST 2 - ERROR 7*****
907                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
908 001726 026705 176724 69$:  CMP      TABLE,R5 ;CHECK R5 UNCHANGED
909 001732 001401      BEQ      70$      ;BR IF OK
910 001734 104010      ERROR    10       ;*****TEST 2 - ERROR 10*****
911                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
912 001736
913
914 001736 012700 017034 MOV      #BUF2,R0   ;VERIFY DESTINATION UNCHANGED
915 001742 105720      TSTB    (R0)+      ;PCINT R0 TO DESTINATION LOWER BYTE BOUNDARY
916 001744 001401      BEQ      71$      ;TEST CONTENTS OF BOUNDARY
917 001746 104011      ERROR    11       ;BR, IF STILL ZERO
918                                     ;*****TEST 2 - ERROR 11*****
919                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
920                                     ; SHOULD STILL EQUAL ZERO
921 001750
922 001750 105720      71$:  TSTB    (R0)+      ;TEST CONTENTS OF DESTINATION BYTE
923 001752 001401      BEQ      2$        ;BR, IF STILL ZERO
924 001754 104012      ERROR    12       ;*****TEST 2 - ERROR 12*****
925                                     ;DESTINATION WAS CHANGED
926 001756
927 001756 105720      2$:  TSTB    (R0)+      ;TEST CONTENTS OF DEST. UPPER BOUNDARY
928 001760 001401      BEQ      ENDT2    ;BR, IF STILL ZERO
929 001762 104013      ERROR    13       ;*****TEST 2 - ERROR 13*****
930                                     ;UPPER BOUNDARY OF DEST. CHANGED
931 001764      ENDT2: ; SHOULD STILL EQUAL ZERO
932

```

M02

933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988

001764 000004
 001766 004567 013220
 001772 000020
 001774 016434
 001776 000011
 002000 017035
 002002 000377
 002004 004767 013230
 002010 004567 013272
 002014 000200
 002016 004767 013124
 002022 000277
 002024 076030
 002026 106767 176640
 002032 042767 177400 176632
 002040 023767 000674 176624
 002046 001401
 002050 104001
 002052
 002052 023706 000676
 002056 001403
 002060 010637 000700
 002064 104002
 002066
 002066 026704 176522
 002072 001401
 002074 104003
 002076 026705 176554
 002102 001401
 002104 104004
 002106
 002106 016705 176532

```

:*****
:TEST 3      TEST "MOVC" INSTRUCTION WITH SRCAD .LT. DSTAD, SL .GT. DL
:*****
:PROPER TERMINATION FOR THIS INSTRUCTION TEST
:IS A TRUNCATED SOURCE STORED IN THE DESTINATION
:(LEAST SIGNIFICANT BYTES NOT MOVED), RD EQUALS THE
:NUMBER OF UNMOVED SOURCE BYTES (SRCLN-DSTLN),
:R1-->R3 EQUAL TO ZERO, AND ALL CONDITION
:CODES CLEAR
:*****
:*****
TST3:  SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      20          ;SOURCE LENGTH
      BUF1       ;SOURCE ADDRESS
      11          ;DESTINATION LENGTH
      BUF2+1     ;DESTINATION ADDRESS
      377        ;FILL CHARACTER
      JSR      PC,CLDST    ;CLEAR DESTINATION
      JSR      R5,XPSW     ;STORE EXPECTED PSW VALUE
      .WORD    200
      JSR      PC,GENR    ;SET UP GENERAL REGISTERS
      SCC          ;SET ALL CONDITION CODES
                   ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MOVC
                   ;CHECK RESULTS
      MFPS      CCODES    ;STORE THE PSW
      BIC      *177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      *EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$        ;BR, IF EQUAL
      ERROR    1          ;*****TEST-2 - ERROR 1*****
                   ;PSW ERROR
                   ;EXPECTED PSW IS STORED AT "EXPPSW"
                   ;ACTUAL PSW IS STORED AT "CCODES"
64$:  CMP      *SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$        ;BR IF OK
      MOV      SP,*BADR6  ;STORE BAD SP VALUE
      ERROR    2          ;*****TEST 3 - ERROR 2*****
                   ;STACK POINTER NOT RESTORED BY INSTRUCTION
                   ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                   ;ERRONEOUS VALUE IS AT "BADR6"
65$:  CMP      FILL,R4    ;CHECK R4 UNCHANGED
      BEQ      66$        ;BR IF OK
      ERROR    3          ;*****TEST 3 - ERROR 3*****
                   ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                   ;CHECK R5 UNCHANGED
66$:  CMP      TABLE,R5 ;CHECK R5 UNCHANGED
      BEQ      67$        ;BR IF OK
      ERROR    4          ;*****TEST 3 - ERROR 4*****
                   ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:  MOV      SRCLN,R5   ;CHECK RD=SOURCE LENGTH MINUS
    
```

NO2

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 26
 CVKRIA.P11 22-DEC-76 10:47 T3

TEST "MOVC" INSTRUCTION WITH SRCAD .LT. DSTAD, SL .GT. DL

SEQ 0027

```

989 002112 166705 176532 SUB DSTLN,R5 ;DESTINATION LENGTH
990 002116 020005 CMP RO,R5
991 002120 001401 BEQ 2$ ;BR IF OK
992 002122 104005 ERROR 5 ;*****TEST 3 - ERROR 5*****
993 ;RO NOT EQUAL TO (SRCLN-DSTLN)
994 002124 2$: ;CHECK OTHER GENERAL REGISTERS
995 002124 005701 TST R1 ;TEST R1
996 002126 001401 BEQ 68$ ;BR, IF ZERO
997 002130 104005 ERROR 6 ;*****TEST 3 - ERROR 6*****
998 ;R1 SHOULD BE ZERO
999 002132 005702 68$: TST R2 ;TEST R2
1000 002134 001401 BEQ 69$ ;BR IF ZERO
1001 002136 104005 ERROR 7 ;*****TEST 3 - ERROR 7*****
1002 ;R2 SHOULD BE ZERO
1003 002140 005703 69$: TST R3 ;TEST R3
1004 002142 001401 BEQ 70$ ;BR, IF ZERO
1005 002144 104005 ERROR 10 ;*****TEST 3 - ERROR 10*****
1006 ;R3 SHOULD BE ZERO
1007 002146 70$:
1008 ;VERIFY DESTINATION AREA
1009 002146 012700 017034 MOV #BUF2,RO ;POINT RO TO DESTINATION LOWER BYTE BOUNDARY
1010 002152 105720 TSTB (RO)+ ;TEST CONTENTS OF BOUNDARY
1011 002154 001401 BEQ 71$ ;BR, IF STILL ZERO
1012 002156 104011 ERROR 11 ;*****TEST 3 - ERROR 11*****
1013 ;LOWER BOUNDARY OF DESTINATION CHANGED
1014 ; SHOULD STILL EQUAL ZERO
1015 002160 71$:
1016 002160 016702 176462 MOV SRCAD,R2 ;POINT R2 TO SOURCE STRING
1017 002164 016701 176460 MOV DSTLN,R1 ;STORE TRANSFER BYTE COUNT IN R1
1018 002170 122022 T3E12: CMPB (RO)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
1019 002172 001401 BEQ +4 ;BR IF OK
1020 002174 104012 ERROR 12 ;*****TEST 3 - ERROR 12*****
1021 ;COMPARE ERROR IN DESTINATION
1022 ;RO CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1023
1024 002176 005301 DEC R1 ;DECREMENT BYTE COUNT
1025 002200 001373 BNE T3E12 ;BR, IF NOT FINISHED CHECKING
1026 002202 105720 TSTB (RO)+ ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1027 002204 001401 BEQ ENCT3 ;BR, IF STILL ZERO
1028 002206 104013 ERROR 13 ;*****TEST 3 - ERROR 13*****
1029 ;UPPER BOUNDARY OF DEST. CHANGED
1030 ; SHOULD STILL EQUAL ZERO
1031 002210 ENCT3:
  
```

1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087

002210 000004
002212 004567 012774
002216 000374
002220 016434
002222 000376
002224 017035
002226 000377
002230 004767 013004
002234 004567 013046
002240 000211
002242 004767 012700
002246 000251
002250 000266
002252 076030
002254 106767 176412
002260 042767 177400 176404
002266 023767 000674 176376
002274 001401
002276 104001
002300
002300 023706 000676
002304 001403
002306 010637 000700
002312 104002
002314
002314 005700
002316 001401
002320 104003
002322
002322 005701
002324 001401
002326 104004
002330 005702

```
*****  
*TEST 4 TEST "MOVC" INSTRUCTION WITH SRCAD .LT. DSTAD, DL .GT.SL  
*****  
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS  
*A TRANSFER OF ALL BYTES FROM SOURCE TO DESTINATION  
*AND "FILL" CHARACTERS IN THE LSB OF THE DESTINATION.  
*R0-->R3 EQUAL TO ZERO, AND CONDITION CODES N,C=1  
*AND Z,V=0  
*****  
*****  
TST4: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
374 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
376 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR DESTINATION AREA & SET DEST. ADDRESS  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 211  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLN!CLC ;CLEAR CONDITION CODES N&C  
+SEV!SEZ ;SET CONDITION CODES V&Z  
MOVC ;EXECUTE "MOVE CHARACTER" INSTRUCTION  
;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 4 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$: CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 4 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
65$: TST R0 ;CHECK R0=ZERO  
BEQ 1$ ;BR, IF ZERO  
ERROR 3 ;*****TEST 4 - ERROR 3*****  
;R0 SHOULD BE ZERO  
1$: TST R1 ;CHECK OTHER GENERAL REGISTERS  
BEQ 66$ ;TEST R1  
ERROR 4 ;BR, IF ZERO  
;*****TEST 4 - ERROR 4*****  
66$: TST R2 ;R1 SHOULD BE ZERO  
;TEST R2
```

```

1088 002332 001401          BEQ     67$      ;BR IF ZERO
1089 002334 104005          ERROR    5      ;*****TEST 4 - ERROR 5*****
1090                                     ;R2 SHOULD BE ZERO
1091 002336 005703          67$:  TST     R3      ;TEST R3
1092 002340 001401          BFQ     68$      ;BR, IF ZERO
1093 002342 104006          ERROR    6      ;*****TEST 4 - ERROR 6*****
1094                                     ;R3 SHOULD BE ZERO
1095 002344                                     68$:
1096 002344 026704 176304          CMP     FILL,R4   ;CHECK R4 UNCHANGED
1097 002350 001401          BEQ     69$      ;BR IF OK
1098 002352 104007          ERROR    7      ;*****TEST 4 - ERROR 7*****
1099                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1100 002354 026705 176276          69$:  CMP     TABLE,R5 ;CHECK R5 UNCHANGED
1101 002360 001401          BEQ     70$      ;BR IF OK
1102 002362 104010          ERROR    10     ;*****TEST 4 - ERROR 10*****
1103                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1104 002364                                     70$:
1105                                     ;VERIFY DESTINATION
1106 002364 012700 017034          MOV     #BUF2,R0  ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1107 002370 105720          TSTB   (R0)+     ;TEST CONTENTS OF BOUNDARY
1108 002372 001401          BEQ     71$      ;BR, IF STILL ZERO
1109 002374 104011          ERROR    11     ;*****TEST 4 - ERROR 11*****
1110                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1111                                     ; SHOULD STILL EQUAL ZERO
1112 002376                                     71$:
1113 002376 016702 176244          2$:  MOV     SRCAD,R2  ;POINT R2 TO SOURCE ADDRESS
1114 002402 016701 176236          MOV     SRCLN,R1 ;STORE TRANSFER BYTE COUNT IN R1
1115 002406 122022          T4E12: CMPB   (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
1116 002410 001401          BEQ     .+4      ;BR IF OK
1117 002412 104012          ERROR    12     ;*****TEST 4 - ERROR 12*****
1118                                     ;COMPARE ERROR IN DESTINATION
1119                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1120
1121 002414 005301          DEC     R1        ;DECREMENT BYTE COUNT
1122 002416 001373          BNE     T4E12    ;BR, IF NOT FINISHED CHECKING
1123 002420 016705 176224          MOV     DSTLN,R5 ;CALCULATE THE NUMBER OF "FILL"
1124 002424 166705 176214          SUB     SRCLN,R5 ;CHARACTERS THAT SHOULD APPEAR IN DEST.
1125 002430 010501          .OV     R5,R1    ;STORE TRANSFER BYTE COUNT IN R1
1126 002432 122067 176216          T4E13: CMPB   (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
1127 002436 001401          BEQ     .+4      ;BR IF OK
1128 002440 104013          ERROR    13     ;*****TEST 4 - ERROR 13*****
1129                                     ;COMPARE ERROR IN DESTINATION
1130                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1131
1132 002442 005301          DEC     R1        ;DECREMENT BYTE COUNT
1133 002444 001372          BNE     T4E13    ;BR, IF NOT FINISHED CHECKING
1134 002446 105720          TSTB   (R0)+     ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1135 002450 001401          BEQ     ENCT4    ;BR, IF STILL ZERO
1136 002452 104014          ERROR    14     ;*****TEST 4 - ERROR 14*****
1137                                     ;UPPER BOUNDARY OF DEST. CHANGED
1138                                     ; SHOULD STILL EQUAL ZERO
1139
1140 002454          ENCT4:
    
```

```

1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154 002454 000004
1155 002456 004567 012533
1156 002462 000010
1157 002464 017035
1158 002466 000015
1159 002470 017035
1160 002472 000377
1161 002474 004767 012540
1162 002500 004567 012562
1163 002504 000010
1164 002506 017035
1165 002510 004567 012572
1166 002514 000211
1167 002516 004767 012424
1168 002522 000245
1169 002524 000272
1170
1171 002526 076030
1172
1173 002530 106767 176136
1174 002534 042767 177400 176130
1175 002542 023767 000674 176122
1176 002550 001401
1177 002552 104001
1178
1179
1180
1181 002554
1182 002554 023706 000676
1183 002560 001403
1184 002562 010637 000700
1185 002566 104002
1186
1187
1188
1189 002570
1190 002570 005700
1191 002572 001401
1192 002574 104003
1193
1194 002576
1195 002576 005701

```

```

*****
*TEST 5 TEST "MOVC" INSTRUCTION WITH SRCAD=DSTAD, SL .LT. DL
*****
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS
*NO CHARACTERS TRANSFERED TO THE DESTINATION,
*RO-->R3 EQUAL TO ZERO, AND CONDITION CODES
*N,V=1 AND C,Z=0.
*BOTH SOURCE AND DESTINATION ADDRESSES SHOULD
*BE DON'T CARE VALUES, THEREFORE A PROBABLE
*NON-EXISTANT MEMORY LOCATION WILL BE USED FOR
*BOTH ADDRESSES
*****
TSTS: SCOPE
      JSR    R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
          10                ;SOURCE LENGTH
          BUF2+1            ;SOURCE ADDRESS
          15                ;DESTINATION LENGTH
          BUF2+1            ;DESTINATION ADDRESS
          377               ;FILL CHARACTER
      JSR    PC,CLDST
      JSR    R5,GENSRC       ;GENERATE A SOURCE STRING
          .WORD 10
          .WORD BUF2+1
      JSR    R5,XPSW        ;STORE EXPECTED PSW VALUE
          .WORD 211
      JSR    PC,GENR        ;SET UP GENERAL REGISTERS
          +CLZ!CLC          ;CLEAR CONDITION CODES Z & C
          +SEV!SEN         ;SET CONDITION CODES V & N
                          ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MOVC
                          ;CHECK RESULTS
      MFPS  CCODES         ;STORE THE PSW
      BIC  #177400,CCODES  ;CLEAR ALL UNUSED BITS
      CMP  @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ  64$             ;BR, IF EQUAL
                          ;*****TEST 5 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "EXPPSW"
                          ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP  @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
      BEQ  65$             ;BR IF OK
      MOV  SP,@#BADR6     ;STORE BAD SP VALUE
      ERROR 2              ;*****TEST 5 - ERROR 2*****
                          ;STACK POINTER NOT RESTORED BY INSTRUCTION
                          ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                          ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      TST  R0              ;CHECK R0 FOR ZERO
      BEQ  1$             ;BR, IF EQUAL
      ERROR 3              ;*****TEST 5 - ERROR 3*****
                          ;R0 SHOULD BE ZERO
      1$:
      TST  R1              ;CHECK OTHER GENERAL REGISTERS
                          ;TEST R1

```

E03

```

1196 002600 001401      BEQ      66$      ;BR, IF ZERO
1197 002602 104004      ERROR    4        ;*****TEST 5 - ERROR 4*****
1198                                ;R1 SHOULD BE ZERO
1199 002604 005702      56$:    TST      R2        ;TEST R2
1200 002606 001401      BEQ      67$      ;BR IF ZERO
1201 002610 104005      ERROR    5        ;*****TEST 5 - ERROR 5*****
1202                                ;R2 SHOULD BE ZERO
1203 002612 005703      67$:    TST      R3        ;TEST R3
1204 002614 001401      BEQ      68$      ;BR, IF ZERO
1205 002616 104006      ERROR    6        ;*****TEST 5 - ERROR 6*****
1206                                ;R3 SHOULD BE ZERO
1207 002620                                68$:
1208 002620 026704 176030  CMP      FILL,R4    ;CHECK R4 UNCHANGED
1209 002624 001401      BEQ      69$      ;BR IF OK
1210 002626 104007      ERROR    7        ;*****TEST 5 - ERROR 7*****
1211                                ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1212 002630 026705 176022  59$:    CMP      TABLE,R5 ;CHECK R5 UNCHANGED
1213 002634 001401      BEQ      70$      ;BR IF OK
1214 002636 104010      ERROR   10       ;*****TEST 5 - ERROR 10*****
1215                                ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1216 002640                                70$:
1217 002640 012700 017034  MOV      #BUF2,R0    ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1218 002644 105720      TSTB    (R0)+       ;TEST CONTENTS OF BOUNDARY
1219 002646 001401      BEQ      71$      ;BR, IF STILL ZERO
1220 002650 104011      ERROR   11       ;*****TEST 5 - ERROR 11*****
1221                                ;LOWER BOUNDARY OF DESTINATION CHANGED
1222                                ; SHOULD STILL EQUAL ZERO
1223 002652                                71$:
1224 002652 012702 017035  MOV      #BUF2+1,R2
1225 002656 016701 175762  MOV      SRCLN,R1    ;STORE TRANSFER BYTE COUNT IN R1
1226 002662 122022      TSE12:  CMPB    (R0)+(R2)+ ;CHECK CHARACTERS IN DESTINATION
1227 002664 001401      BEQ      +4        ;BR IF OK
1228 002666 104012      ERROR   12       ;*****TEST 5 - ERROR 12*****
1229                                ;COMPARE ERROR IN DESTINATION
1230                                ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1231
1232 002670 005301      DEC      R1        ;DECREMENT BYTE COUNT
1233 002672 001373      BNE     TSE12      ;BR, IF NOT FINISHED CHECKING
1234 002674 016702 175750  MOV      DSTLN,R2    ;CALCULATE FILL LENGTH
1235 002700 166702 175740  SUB      SRCLN,R2
1236 002704 010201      MOV      R2,R1      ;STORE TRANSFER BYTE COUNT IN R1
1237 002706 122067 175742  TSE13:  CMPB    (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
1238 002712 001401      BEQ      +4        ;BR IF OK
1239 002714 104013      ERROR   13       ;*****TEST 5 - ERROR 13*****
1240                                ;COMPARE ERROR IN DESTINATION
1241                                ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1242
1243 002716 005301      DEC      R1        ;DECREMENT BYTE COUNT
1244 002720 001372      BNE     TSE13      ;BR, IF NOT FINISHED CHECKING
1245 002722 105720      TSTB    (R0)+       ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1246 002724 001401      BEQ      ENDS      ;BR, IF STILL ZERO
1247 002726 104014      ERROR   14       ;*****TEST 5 - ERROR 14*****
1248                                ;UPPER BOUNDARY OF DEST. CHANGED
1249                                ; SHOULD STILL EQUAL ZERO
1250 002730      ENDS:

```

F03

```

1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264 002730 000004
1265 002732 004567 012254
1266 002736 000020
1267 002740 017034
1268 002742 000007
1269 002744 017037
1270 002746 000377
1271 002750 004567 012312
1272 002754 000020
1273 002756 017034
1274 002760 004567 012322
1275 002764 000200
1276 002766 004767 012154
1277
1278
1279 002772 076030
1280
1281 002774 106767 175672
1282 003000 042767 177400 175664
1283 003006 023767 000674 175656
1284 003014 001401
1285 003016 104001
1286
1287
1288
1289 003020
1290 003020 023706 000676
1291 003024 001403
1292 003026 010637 000700
1293 003032 104002
1294
1295
1296
1297 003034
1298 003034 026704 175614
1299 003040 001401
1300 003042 104003
1301
1302 003044 026705 175606
1303 003050 001401
1304 003052 104004
1305
1306 003054

```

```

:*****
:*TEST 6 TEST "MOVC" WITH SRCAD .LT. DSTAD, SRCLN .GT. DSTLN
:*****
:*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN
:*THE DESTINATION AREA IS CONTAINED WITHIN THE SOURCE AREA.
:*THEREFORE, SRCAD<DSTAD AND SRCLN>DSTLN
:*THE RESULT IS A TRUNCATED (LSB OF SOURCE NOT MOVED) SOURCE
:*IN THE DESTINATION, RD EQUALS (SRCLN-DSTLN), AND ALL CONDITION
:*CODES CLEAR
:*****
*ST6: SCOPE
      JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
              20              ;SOURCE LENGTH
              BUF2            ;SOURCE ADDRESS
              7              ;DESTINATION LENGTH
              BUF2+3          ;DESTINATION ADDRESS
              377            ;FILL CHARACTER
              R5,GENSRC       ;GENERATE SOURCE STRING
      .WORD   20
      .WORD   BUF2
      JSR      R5,XPSW         ;STORE EXPECTED PSW VALUE
      .WORD   200
      JSR      PC,GENR        ;SET UP GENERAL REGISTERS
                              ;SET ALL CONDITION CODES
                              ;EXECUTE "MOVE CHARACTER" INSTRUCTION
                              ;CHECK RESULTS
                              ;STORE THE PSW
      MFPS    CCODES          ;CLEAR ALL UNUSED BITS
      BIC     #177400,CCODES
      CMP     3#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ    64$
      ERROR  1
      *****TEST 6 - ERROR 1*****
      PSW ERROR
      EXPECTED PSW IS STORED AT "EXPPSW"
      ACTUAL PSW IS STORED AT "CCODES"
64$:
      CMP     2#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
      BEQ    65$
      MOV     SP,3#BADP6     ;STORE BAD SP VALUE
      ERROR  2
      *****TEST 6 - ERROR 2*****
      STACK POINTER NOT RESTORED BY INSTRUCTION
      EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ERRONEOUS VALUE IS AT "BADP6"
65$:
      CMP     FILL,R4        ;CHECK R4 UNCHANGED
      BEQ    66$
      ERROR  3
      *****TEST 6 - ERROR 3*****
      R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      CHECK R5 UNCHANGED
      BEQ    67$
      ERROR  4
      *****TEST 6 - ERROR 4*****
      R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:

```


1307	003054	016705	175564	MOV	SRCLN,R5	:CHECK R0=SOURCE LENGTH MINUS
1308	003060	166705	175564	SUB	DSTLN,R5	:DESTINATION LENGTH
1309	003064	020005		CMP	R0,R5	
1310	003066	001401		BEG	3\$	
1311	003070	104005		ERROR	5	:*****TEST 6 - ERROR 5*****
1312						:R0 NOT EQUAL TO (SRCLN-DSTLN)
1313	003072		2\$:	TST	R1	:TEST R1
1314	003072	005701		BEG	6\$:BR, IF ZERO
1315	003074	001401		ERROR	6	:*****TEST 6 - ERROR 6*****
1316	003076	104006				:R1 SHOULD BE ZERO
1317				TST	R2	:TEST R2
1318	003100	005702	5\$:	BEG	6\$:BR IF ZERO
1319	003102	001401		ERROR	7	:*****TEST 6 - ERROR 7*****
1320	003104	104007				:R2 SHOULD BE ZERO
1321				TST	R3	:TEST R3
1322	003106	005703	6\$:	BEG	70\$:BR, IF ZERO
1323	003110	001401		ERROR	10	:*****TEST 6 - ERROR 10*****
1324	003112	104010				:R3 SHOULD BE ZERO
1325						
1326	003114		70\$:			:VERIFY DESTINATION & NON-OVERLAP SOURCE
1327						:CHECK BEGINNING OF OLD SOURCE AREAS
1328	003114	012700	016434	MOV	#BUF1,R0	
1329	003120	012701	017034	MOV	#BUF2,R1	
1330	003124	122021		CMPB	(R0)+,(R1)+	
1331	003126	001401	3\$:	BEG	4\$	
1332	003130	104011		ERROR	11	:*****TEST 6 - ERROR 11*****
1333						:NON-OVERLAP SOURCE AREA CHANGED
1334						:R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1335	003132	020167	175514	CMP	R1,DSTAD	:REACHED START OF DESTINATION?
1336	003136	002772		BLT	3\$:BR, IF NO
1337	003140	012702	016434	MOV	#BUF1,R2	:CHECK DESTINATION AREA
1338	003144	016703	175500	MOV	DSTLN,R3	
1339	003150	122221		CMPB	(R2)+,(R1)+	
1340	003152	001401	5\$:	BEG	6\$	
1341	003154	104012		ERROR	12	:*****TEST 6 - ERROR 12*****
1342						:COMPARE ERROR IN DESTINATION
1343						:R1 CONTAINS PC+1 OF BAD DESTINATION CHARACTER
1344	003156	005200	6\$:	INC	R0	:UPDATE POINTER TO OLD SOURCE
1345	003160	005303		DEC	R3	:DECREMENT DEST. BYTE COUNT
1346	003162	001372		BNE	5\$:BR, IF NOT FINISHED
1347	003164	016703	175456	MOV	SRCAD,R3	:CHECK LOWER PORTION OF NON-OVERLAP SOURCE
1348	003170	066703	175450	ADD	SRCLN,R3	
1349	003174	122021		CMPB	(R0)+,(R1)+	
1350	003176	001401	7\$:	BEG	10\$	
1351	003200	104013		ERROR	13	:*****TEST 6 - ERROR 13*****
1352						:NON-OVERLAP SOURCE AREA CHANGED
1353						:R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1354	003202	020103	10\$:	CMP	R1,R3	:FINISHED CHECK?
1355	003204	002773		BLT	7\$:BR, IF NO

H03

1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411

003206 000004
003210 004567 011776
003214 000007
003216 017037
003220 000017
003222 017035
003224 000377
003226 004767 012006
003232 004567 012030
003236 000007
003240 017037
003242 004567 012040
003246 000211
003250 004767 011572
003254 000251
003256 000266

003260 076030

003262 106767 175404
003266 042767 177400 175376
003274 023767 000674 175370
003302 001401
003304 104001

003306
003306 023706 000676
003312 001403
003314 010637 000700
003320 104002

003322
003322 005700
003324 001401
003326 104003

003330
003330 005701

```
*****  
*TEST 7 TEST "MOVC" WITH DSTAD .GT. SRCAD, SRCLN .LT. DSTLN  
*****  
*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN  
*THE SOURCE AREA IS CONTAINED WITHIN THE DESTINATION AREA.  
*THEREFORE, SRCAD>DSTAD AND DSTLN>SRCLN.  
*THE RESULT IS A FULL TRANSFER OF THE SOURCE TO THE  
*DESTINATION AND "FILL" CHARACTERS IN THE LSB OF THE  
*DESTINATION; R0-->R3 EQUAL ZERO, AND CONDITION  
*CODES N,C=1 AND Z,V=0  
*****  
↑ST7: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
7 ;SOURCE LENGTH  
BUF2+3 ;SOURCE ADDRESS  
1 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR DESTINATION  
JSR R5,GENSRC ;GENERATE A SOURCE STRING  
.WORD 7  
BUF2+3  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 211  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLN!CLC ;CLEAR CONDITION CODES N & C  
+SEV!SEZ ;SET CONDITION CODES V & Z  
;EXECUTE "MOVE CHARACTER" INSTRUCTION  
MOVC  
;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 7 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$:  
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR, IF OK  
MOV SP,#BADR6 ;STORE BAD SP VALUE  
ERRCR 2 ;*****TEST 7 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
65$:  
TST R0 ;CHECK R0 FOR ZERO  
BEQ 1$ ;BR, IF ZERO  
ERROR 3 ;*****TEST 7 - ERROR 3*****  
;R0 SHOULD BE ZERO  
1$:  
TST R1 ;TEST R1
```

```

1412 003332 001401      BEQ      66$      ;BR, IF ZERO
1413 003334 104004      ERROR    4        ;*****TEST 7 - ERROR 4*****
1414                                     ;R1 SHOULD BE ZERO
1415 003336 005702      66$:    TST      R2        ;TEST R2
1416 003340 001401      BEQ      67$      ;BR IF ZERO
1417 003342 104005      ERROR    5        ;*****TEST 7 - ERROR 5*****
1418                                     ;R2 SHOULD BE ZERO
1419 003344 005703      67$:    TST      R3        ;TEST R3
1420 003346 001401      BEQ      68$      ;BR, IF ZERO
1421 003350 104006      ERROR    6        ;*****TEST 7 - ERROR 6*****
1422                                     ;R3 SHOULD BE ZERO
1423 003352                                     68$:
1424 003352 026704 175276    CMP      FILL,R4    ;CHECK R4 UNCHANGED
1425 003356 001401      BEQ      69$      ;BR IF OK
1426 003360 104007      ERROR    7        ;*****TEST 7 - ERROR 7*****
1427                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1428 003362 026705 175270    69$:    CMP      TABLE,R5 ;CHECK R5 UNCHANGED
1429 003366 001401      BEQ      70$      ;BR IF OK
1430 003370 104010      ERROR    10       ;*****TEST 7 - ERROR 10*****
1431                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1432 003372                                     70$:
1433                                     ;VERIFY DESTINATION
1434 003372 012700 017034    MOV      #BUF2,R0   ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1435 003376 105720      TSTB    (R0)+       ;TEST CONTENTS OF BOUNDARY
1436 003400 001401      BEQ      71$      ;BR, IF STILL ZERO
1437 003402 104011      ERROR    11       ;*****TEST 7 - ERROR 11*****
1438                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1439                                     ; SHOULD STILL EQUAL ZERO
1440 003404                                     71$:
1441 003404 012702 016434    2$:    MOV      #BUF1,R2   ;POINT R2 TO ORIGINAL SOURCE BYTES
1442 003410 016701 175230    MOV      SRCLN,R1   ;STORE TRANSFER BYTE COUNT IN R1
1443 003414 122022      77E12: CMPB    (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
1444 003416 001401      BEQ      12$      ;BR IF OK
1445 003420 104012      ERROR    12       ;*****TEST 7 - ERROR 12*****
1446                                     ;COMPARE ERROR IN DESTINATION
1447                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1448
1449 003422 005301      DEC      R1        ;DECREMENT BYTE COUNT
1450 003424 001373      BNE     T7E12     ;BR, IF NOT FINISHED CHECKING
1451 003426 016705 175216    MOV      DSTLN,R5   ;CHECK FILL
1452 003432 166705 175206    SUB      SRCLN,R5
1453 003436 122067 175212    3$:    CMPB    (R0)+,FILL
1454 003442 001401      BEQ      4$      ;BR IF OK
1455 003444 104013      ERROR    13       ;*****TEST 7 - ERROR 13*****
1456                                     ;"FILL" ERROR IN DESTINATION
1457 003446 105305      4$:    DECB   R5        ;
1458 003450 001372      BNE     3$      ;
1459 003452 105720      TSTB   (R0)+     ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1460 003454 001401      BEQ     END7     ;BR, IF STILL ZERO
1461 003456 104014      ERROR   14       ;*****TEST 7 - ERROR 14*****
1462                                     ;UPPER BOUNDARY OF DEST. CHANGED
1463                                     ; SHOULD STILL EQUAL ZERO
1464 003460      END7:
  
```

1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520

003460 000004
003462 004567 011524
003466 000020
003470 017054
003472 000007
003474 017035
003476 000377
003500 004767 011534
003504 004567 011556
003510 000020
003512 017054
003514 004567 011566
003520 000200
003522 004767 011420
003526 000277
003530 076030
003532 106767 175134
003536 042767 177400 175126
003544 023767 000674 175120
003552 001401
003554 104001
003556
003556 023706 000676
003562 001403
003564 010E37 000700
003570 104002
003572
003572 026704 17505E
003576 001401
003600 104003
003602 026705 17505C
003606 001401
003610 104004

```
*****  
*TEST 10 TEST "MOVC" WITH DSTAC .LT. SRCAD & SRCLN .GT. DSTLN. TRUNCATION.  
*****  
*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN THE  
*SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION AND THE TRANSFER  
*IS TRUNCATED (SRCLN > DSTLN).  
*THE RESULT IS A TRUNCATED SOURCE IN THE DESTINATION, R0-->R3  
*EQUAL ZERO, AND ALL CONDITION CODES CLEAR.  
*****  
*ST10: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
20 ;SOURCE LENGTH  
BUF2+20 ;SOURCE ADDRESS  
7 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR DESTINATION AREA  
JSR R5,GENSRC ;GENERATE SOURCE STRING  
.WORD 20  
.WORD BUF2+20  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 200  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES  
MOVC ;EXECUTE "MOVE CHARACTER" INSTRUCTION  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 10 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$:  
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,#BADP6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 10 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADP6"  
65$:  
CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 10 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
66$:  
CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 10 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
```

K03

TEST "MOVC" WITH DSTAD .LT. SRCAD & SRCLN .GT. DSTLN. (TRUNCATION)

SEQ 0037

```

1521 003612          67$:
1522 003612 016705 175026      MOV      SRCLN,R5      ;CHECK R0=SOURCE LENGTH MINUS
1523 003616 166705 175026      SUB      DSTLN,R5      ; DESTINATION LENGTH
1524 003622 020005          CMP      R0,R5
1525 003624 001401          BEQ     68$
1526 003626 104005          ERROR   5      ;*****TEST 10 - ERROR 5*****
1527                                ;R0 NOT EQUAL TO (SRCLN-DSTLN)
1529 003630          69$:      ;CHECK OTHER GENERAL REGISTERS
1529 003630 005701          TST     R1      ;TEST R1
1530 003632 001401          BEQ     69$     ;BR, IF ZERO
1531 003634 104006          ERROR   6      ;*****TEST 10 - ERROR 6*****
1532                                ;R1 SHOULD BE ZERO
1533 003636 005702          69$:      TST     R2      ;TEST R2
1534 003640 001401          BEQ     70$     ;BR IF ZERO
1535 003642 104007          ERROR   7      ;*****TEST 10 - ERROR 7*****
1536                                ;R2 SHOULD BE ZERO
1537 003644 005703          70$:      TST     R3      ;TEST R3
1538 003646 001401          BEQ     71$     ;BR, IF ZERO
1539 003650 104010          ERROR   10     ;*****TEST 10 - ERROR 10*****
1540                                ;R3 SHOULD BE ZERO
1541 003652          71$:
1542                                ;VERIFY DESTINATION AREA
1543 003652 012700 017034      MOV     #BUF2,R0     ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1544 003656 105720          TSTB   (R0)+        ;TEST CONTENTS OF BOUNDARY
1545 003660 001401          BEQ     72$     ;BR, IF STILL ZERO
1546 003662 104011          ERROR   11     ;*****TEST 10 - ERROR 11*****
1547                                ;LOWER BOUNDARY OF DESTINATION CHANGED
1548                                ; SHOULD STILL EQUAL ZERO
1549 003664          72$:
1550 003664 016702 174756      MOV     SRCAD,R2
1551 003670 016701 174754      MOV     DSTLN,R1
1552 003674 122022          TICE12: CMPB   (R0)+,(R2)+ ;STORE TRANSFER BYTE COUNT IN R1
1553 003676 001401          BEQ     +4        ;CHECK CHARACTERS IN DESTINATION
1554 003700 104012          ERROR   12     ;BR IF OK
1555                                ;*****TEST 10 - ERROR 12*****
1556                                ;COMPARE ERROR IN DESTINATION
1557                                ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1558 003702 005301          DEC     R1      ;DECREMENT BYTE COUNT
1559 003704 001373          BNE    TICE12    ;BR, IF NOT FINISHED CHECKING
1560 003706 105720          TSTB   (R0)+        ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1561 003710 001401          BEQ     END-10    ;BR, IF STILL ZERO
1562 003712 104013          ERROR   13     ;*****TEST 10 - ERROR 13*****
1563                                ;UPPER BOUNDARY OF DEST. CHANGED
1564                                ; SHOULD STILL EQUAL ZERO
1565 003714          ENDTIC:
1566

```

```

1567
1568
1569
1570
1571
1572
1573
1574
1575
1576 003714 000004
1577 003716 004567 01127C
1578 003722 016434
1579 003724 000010
1580 003726 017034
1581 003730 000010
1582 003732 000377
1583 003734 012737 076032 003762
1584 003742 013767 003310 174736
1585 003750 012737 003772 000010
1586 003755 004767 011164
1587
1588 003762 076032
1589
1590 003764 016700 177772
1591 003770 104001
1592
1593
1594
1595 003772 012626
1596 003774 005267 177762
1597 004000 022767 076040 177754
1598 004006 001363
1599 004010 016737 174672 000010
1600

```

```

;*****
;*TEST 11      TEST THAT BAD "MOVE" OPCODES TRAP
;*****
;*THIS TEST VERIFIES THAT OPCODES 076032-->076037 TRAP TO
;*LOCATION 10
;*****
;*****
TST11:  SCOPE
        JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
        JSR      BUF1        ;SOURCE LENGTH
        JSR      10          ;SOURCE ADDRESS
        JSR      BUF2        ;DESTINATION LENGTH
        JSR      10          ;DESTINATION ADDRESS
        JSR      377         ;FILL CHARACTER
        MOV      #076032,2#BD11 ;STORE THE FIRST BAD MOVE OPCODE
        MOV      2#10,TEMP1   ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
        MOV      #T11CONT,2#10 ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST
REF11:  JSR      PC,GENR      ;SET UP GENERAL REGISTERS
BD11:   .WORD    076032      ;EXECUTE BAD MOVE INSTRUCTION
        MOV      BD11,RO      ;STORE BAD OPCODE THAT DID NOT TRAP
        ERROR    1           ;*****TEST 11 - ERROR 1*****
        ;BAD MOVE OPCODE DID NOT TRAP
        ;RO CONTAINS THE BAD OPCODE
T11CONT:MOV  (SP)+,(SP)+    ;RESTORE THE STACK POINTER AFTER THE TRAP
        INC     BD11        ;INCREMENT INSTRUCTION OPCODE
        CMP     #076040,BD11 ;FINISHED WITH BAD MOVE OPCODES?
        BNE    REF11       ;BR IF NOT
        MOV     TEMP1,2#10  ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR

```

```

1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615 004016 000004
1616 004020 105777 174514
1617 004024 100570
1618 004026 026767 174516 174624
1619 004034 001007
1620 004036 032767 000001 174542
1621 004044 001403
1622 004046 005767 174522
1623 004052 001155
1624 004054
1625 004054 004567 011132
1626 004060 000374
1627 004052 016434
1628 004064 000376
1629 004066 017035
1630 004070 000377
1631 004072 004767 011142
1632 004076 012767 004160 174564
1633 004104 012777 015356 174552
1634 004112 005077 174550
1635 004116 004767 011210
1636 004122 013777 000554 174532
1637 004130 004567 011152
1638 004134 000011
1639 004136 006427 000000
1640 004142 052777 000100 174510
1641 004150 004767 010772
1642 004154 000251
1643 004156 000266
1644
1645 004160 076030
1646
1647 004162 106767 174504
1648 004166 032777 000100 174464
1649 004174 001365
1650 004176 042767 177400 17446E
1651 004204 023767 000674 174460
1652 004212 001401
1653 004214 104001
1654
1655
1656

```

```

*****
*TEST 12 TEST INTERRUPTABILITY OF "MOVC" INSTRUCTION
*****
*THIS TEST INTERRUPTS THE EXECUTION OF THE "MOVC"
*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
*INTERRUPT, AND VERIFIES THE RESULTS.
*THE PROPER RESULT IS ALL BYTES MOVED
*TO THE DESTINATION, R0-->R3 EQUAL ZERO
*AND CONDITION CODES N,C=1 AND Z,V=0
*****
TST12: SCOPE
      TSTB @SWR ;TEST BIT 7 OF SWR
      BMI TST13 ;SKIP TO NEXT TEST IF SET
      CMP @TPS,@TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
      BNE T12CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
      BIT @BIT0,@ENV ;IF YES, CHECK IF ON APT
      BEQ T12CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
      TST @PASS ;CHECK IF ON FIRST PASS
      BNE TST13 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T12CONT: JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
          ;SOURCE LENGTH
          BUF1 ;SOURCE ADDRESS
          ;DESTINATION LENGTH
          BUF2+1 ;DESTINATION ADDRESS
          ;FILL CHARACTER
          ;CLEAR DESTINATION
          JSR PC,CLDST
          MOV @MC,@PCI ;STORE PC OF TEST INSTRUCTION
          MOV @INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
          CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
          JSR PC,TDONE
          MOV @@NULL,@TBUF ;SEND CARRIAGE RETURN
          JSR R5,XPSW ;STORE EXPECTED PSW VALUE
          .WORD 11
          MTPS @0 ;SET PSW TO ALLOW INTERRUPTS
          BIS @100,@TCSR ;ENABLE TTY INTERRUPTS
          JSR PC,GENR ;SET UP GENERAL REGISTERS
          +CLN!CLC
          +SEV!SEZ

MC: MOVC ;EXECUTE "MOVE CHARACTER" INSTRUCTION

MFPS CCODES ;STORE THE PSW
BIT @100,@TCSR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
BNE REPMC ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
BIC @177400,CCODES ;CLEAR ALL UNUSED BITS
CMP @@EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 12 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "CCODES"

```

1657	004216			64\$:					
1658									
1659	004216	023706	000676		CMP	2#SAVR6,SP			:CHECK RESULTS
1660	004222	001403			BEQ	65\$:VERIFY STACK POINTER IS RESTORED
1661	004224	010637	000700		MOV	SP,2#BADR6			:BR IF OK
1662	004230	104002			ERROR	2			:STORE BAD SP VALUE
1663									:*****TEST 12 - ERROR 2*****
1664									:STACK POINTER NOT RESTORED BY INSTRUCTION
1665									:EXPECTED VALUE OF SP IS STORED AT "SAVRE"
1666	004232			65\$:					:ERRONEOUS VALUE IS AT "BADR6"
1667	004232	005700			TST	R0			:CHECK R0=ZERO
1668	004234	001401			BEQ	1\$			
1669	004236	104003			ERROR	3			:*****TEST 12 - ERROR 3*****
1670									:R0 SHOULD BE ZERO
1671	004240			1\$:					:CHECK OTHER GENERAL REGISTERS
1672	004240	005701			TST	R1			:TEST R1
1673	004242	001401			BEQ	66\$:BR, IF ZERO
1674	004244	104004			ERROR	4			:*****TEST 12 - ERROR 4*****
1675									:R1 SHOULD BE ZERO
1676	004246	005702		66\$:	TST	R2			:TEST R2
1677	004250	001401			BEQ	67\$:BR IF ZERO
1678	004252	104005			ERROR	5			:*****TEST 12 - ERROR 5*****
1679									:R2 SHOULD BE ZERO
1680	004254	005703		67\$:	TST	R3			:TEST R3
1681	004256	001401			BEQ	68\$:BR, IF ZERO
1682	004260	104006			ERROR	6			:*****TEST 12 - ERROR 6*****
1683									:R3 SHOULD BE ZERO
1684	004262			68\$:					
1685	004262	026704	174366		CMP	FILL,R4			:CHECK R4 UNCHANGED
1686	004266	001401			BEQ	69\$:BR IF OK
1687	004270	104007			ERROR	7			:*****TEST 12 - ERROR 7*****
1688									:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1689	004272	026705	174360	69\$:	CMP	TABLE,R5			:CHECK R5 UNCHANGED
1690	004276	001401			BEQ	70\$:BR IF OK
1691	004300	104010			ERROR	10			:*****TEST 12 - ERROR 10*****
1692									:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1693	004302			70\$:					
1694	004302	012700	017034		MOV	#BUF2,R0			:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1695	004306	012700	017034		MOV	#BUF2,R0			:TEST CONTENTS OF BOUNDARY
1696	004312	105720			TSTB	(R0)+			:BR, IF STILL ZERO
1697	004314	001401			BEQ	71\$:*****TEST 12 - ERROR 11*****
1698	004316	104011			ERROR	11			:LOWER BOUNDARY OF DESTINATION CHANGED
1699									: SHOULD STILL EQUAL ZERO
1700									
1701	004320			71\$:					
1702	004320	016702	174322	4\$:	MOV	SRCAD,R2			:POINT R2 TO THE SOURCE ADDRESS
1703	004324	016701	174314		MOV	SRCLN,R1			:STORE TRANSFER BYTE COUNT IN R1
1704	004330	122022		T12E12:	CMPB	(R0)+,(R2)+			:CHECK CHARACTERS IN DESTINATION
1705	004332	001401			BEQ	+4			:BR IF OK
1706	004334	104012			ERROR	12			:*****TEST 12 - ERROR 12*****
1707									:COMPARE ERROR IN DESTINATION
1708									:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1709									
1710	004336	005301			DEC	R1			:DECREMENT BYTE COUNT
1711	004340	001373			BNE	T12E12			:BR, IF NOT FINISHED CHECKING
1712	004342	016701	174302		MOV	DSTLN,R1			:CALCULATE THE NUMBER OF "FILL"

TEST INTERRUPTABILITY OF "MOVC" INSTRUCTION

SEQ 2041

1713	004346	166701	174272		SUB	SRCLN,R1	:CHARACTERS THAT SHOULD APPEAR IN DEST.
1714	004352	122067	174276	7\$:	CMPB	(R0)+,FILL	:CHECK LSB'S OF DEST. FOR FILLS
1715	004356	001401			BEQ	6\$:BR, IF EQUAL
1716	004360	104013			ERROR	13	:*****TEST 12 - ERROR 13*****
1717							: "FILL" ERROR IN DESTINATION
1718	004362	005301		6\$:	DEC	R1	:DECREMENT FILL COUNT
1719	004364	001372			BNE	5\$:BR, IF NOT FINISHED
1720	004366	105720			TSTB	(R0)+	:TEST CONTENTS OF DEST. UPPER BOUNDARY
1721	004370	001401			BEQ	ENDT12	:BR, IF STILL ZERO
1722	004372	104014			ERROR	14	:*****TEST 12 - ERROR 14*****
1723							:UPPER BOUNDARY OF DEST. CHANGED
1724							: SHOULD STILL EQUAL ZERO
1725	004374			ENDT12:			
1726							
1727							
1728	004374	106427	000200	40\$:	MTPS	#200	:RESTORE PSW TO PRIORITY 7
1729	004400	016777	174262 174256		MOV	TPSW,@TVECT	:RESTORE TRAP CATCHER

1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785

004406 000004
004410 004567 010576
004414 000020
004416 016434
004420 000011
004422 017035
004424 000377
004426 004767 010606
004432 004567 010650
004436 000200
004440 004767 010502
004444 000277
004446 076031
004450 106767 174216
004454 042767 177400 174210
004462 023767 000674 174202
004470 001401
004472 104001
004474
004474 023706 000676
004500 001403
004502 010637 000700
004506 104002
004510
004510 026704 174140
004514 001401
004516 104003
004520 026705 174132
004524 001401
004526 104004
004530

```
*****  
:TEST 13 TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), SL .GT. DL  
:*****  
:PROPER TERMINATION FOR THIS INSTRUCTION TEST IS A  
:TRUNCATED SOURCE REVERSE (HIGH ADDRESS) JUSTIFIED  
:(WITH MOST SIGNIFICANT BYTES NOT MOVED) IN THE  
:DESTINATION. R0 EQUALS THE NUMBER OF UNMOVED SOURCE  
:BYTES, R1-->R3 EQUAL ZERO AND ALL CONDITION CODES  
:CLEAR  
:*****  
:*****  
:*****  
TST13: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
20 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
11 ;DESTINATION LENGTH  
BJF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLOST ;CLEAR DESTINATION  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 200  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES  
MOVRC ;EXECUTE "MOVE REVERSE CHARACTER"  
MFP5 ;CHECK RESULTS  
CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 13 - ERROR 1*****  
PSW ERROR  
EXPECTED PSW IS STORED AT "EXPPSW"  
ACTUAL PSW IS STORED AT "CCODES"  
64$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 13 - ERROR 2*****  
STACK POINTER NOT RESTORED BY INSTRUCTION  
EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
ERRONEOUS VALUE IS AT "BADR6"  
65$: CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 13 - ERROR 3*****  
R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 13 - ERROR 4*****  
R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"  
67$:
```

1786	004530	016705	174110	MOV	SRCLN,R5	:CHECK R0=SOURCE LENGTH MINUS
1787	004534	166705	174110	SUB	DSTLN,R5	: DESTINATION LENGTH
1788	004540	020005		CMP	R0,R5	
1789	004542	001401		BEQ	68\$	
1790	004544	104005		ERROR	5	:*****TEST 13 - ERROR 5*****
1791						:R0 NOT EQUAL TO (SRCLN-DSTLN)
1792	004546			68\$:		:CHECK OTHER GENERAL REGISTERS
1793	004546	005701		TST	R1	:TEST R1
1794	004550	001401		BEQ	69\$:BR, IF ZERO
1795	004552	104006		ERROR	6	:*****TEST 13 - ERROR 6*****
1796						:R1 SHOULD BE ZERO
1797	004554	005702		69\$:	TST	R2
1798	004556	001401		BEQ	70\$:TEST R2
1799	004560	104007		ERROR	7	:BR IF ZERO
1800						:*****TEST 13 - ERROR 7*****
1801	004562	005703		70\$:	TST	R3
1802	004564	001401		BEQ	71\$:TEST R3
1803	004566	104010		ERROR	10	:BR, IF ZERO
1804						:*****TEST 13 - ERROR 10*****
1805	004570			71\$:		:R3 SHOULD BE ZERO
1806						:VERIFY DESTINATION AREA
1807	004570	012700	017034	MOV	#BUF2,R0	:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1808	004574	105720		TSTB	(R0)+	:TEST CONTENTS OF BOUNDARY
1809	004576	001401		BEQ	72\$:BR, IF STILL ZERO
1810	004600	104011		ERROR	11	:*****TEST 13 - ERROR 11*****
1811						:LOWER BOUNDARY OF DESTINATION CHANGED
1812						: SHOULD STILL EQUAL ZERO
1813	004602			72\$:		
1814	004602	016702	174036	MOV	SRCLN,R2	:CALCULATE ADDRESS OF MSB OF THE
1815	004606	166702	174036	SUB	DSTLN,R2	: SOURCE TO BE MOVED
1816	004612	066702	174030	ADD	SRCAD,R2	
1817	004616	016701	174026	MOV	DSTLN,R1	:STORE TRANSFER BYTE COUNT IN R1
1818	004622	122022		T13E12:	CMPB	(R0)+,(R2)+
1819	004624	001401		BEQ	+4	:CHECK CHARACTERS IN DESTINATION
1820	004626	104012		ERROR	12	:BR IF OK
1821						:*****TEST 13 - ERROR 12*****
1822						:COMPARE ERROR IN DESTINATION
1823						:R0 CONTAIN THE PC+1 OF THE BAD DESTINATION CHARACTER
1824	004630	005301		DEC	R1	:DECREMENT BYTE COUNT
1825	004632	001373		BNE	T13E12	:BR, IF NOT FINISHED CHECKING
1826	004634	105720		TSTB	(R0)+	:TEST CONTENTS OF DEST. UPPER BOUNDARY
1827	004636	001401		BEQ	ENDT13	:BR, IF STILL ZERO
1828	004640	104013		ERROR	13	:*****TEST 13 - ERROR 13*****
1829						:UPPER BOUNDARY OF DEST. CHANGED
1830						: SHOULD STILL EQUAL ZERO
1831	004642			ENDT13:		

1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887

004642 000004
004644 004567 010342
004650 000011
004652 016434
004654 000020
004656 017035
004660 000377
004662 004767 010352
004666 004567 010414
004672 000211
004674 004767 010246
004700 000251
004702 000266
004704 076031
004706 1067E 173760
004712 042767 177400 173752
004720 023767 000674 173744
004726 001401
004730 104001
004732
004732 023706 000676
004736 001403
004740 010637 000700
004744 104002
004746
004746 005700
004750 001401
004752 104003
004754
004754 005701
004756 001401
004760 104004

```
*****  
*TEST 14 TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL  
*****  
*PROPER TERMINATION FOR THIS INSTRUCTION TEST IS A  
*TRANSFER OF ALL BYTES FROM SOURCE TO DESTINATION  
*AND "FILL" CHARACTERS IN THE MSB OF THE DESTINATION.  
*(R0-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1 AND  
*Z,V=0  
*****  
TST14: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
11 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
20 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR DESTINATION  
JSR S,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 211  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLN:CLC ;CLEAR CONDITION CODES N&C  
+SEV:SEZ ;SET CONDITION CODES V&Z  
;EXECUTE "MOVE REVERSE CHARACTER"  
MOVRC ;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #17400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 14 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$:  
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 14 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
65$:  
TST R0 ;CHECK R0=ZERO  
BEQ 1$ ;BR, IF ZERO  
ERROR 3 ;*****TEST 14 - ERROR 3*****  
;R0 SHOULD BE ZERO  
;CHECK OTHER GENERAL REGISTERS  
1$:  
TST R1 ;TEST R1  
BEQ 66$ ;BR, IF ZERO  
ERROR 4 ;*****TEST 14 - ERROR 4*****  
;R1 SHOULD BE ZERO
```

```

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 44
DVKAIA.P11 22-DEC-76 10:47 T14 TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL      SEQ 0045

1888 004762 005702          65$: TST      R2          ;TEST R2
1889 004764 001401          BEQ      67$          ;BR IF ZERO
1890 004766 104005          ERROR    5           ;*****TEST 14 - ERROR 5*****
1891                                     ;R2 SHOULD BE ZERO
1892 004770 005703          67$: TST      R3          ;TEST R3
1893 004772 001401          BEQ      69$          ;BR, IF ZERO
1894 004774 104006          ERROR    6           ;*****TEST 14 - ERROR 6*****
1895                                     ;R3 SHOULD BE ZERO
1896 004776                                     69$:
1897 004776 026704 173652    CMP      FILL,R4      ;CHECK R4 UNCHANGED
1898 005002 001401          BEQ      69$          ;BR IF OK
1899 005004 104007          ERROR    7           ;*****TEST 14 - ERROR 7*****
1900                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1901 005006 026705 173644    69$: CMP      TABLE,R5 ;CHECK R5 UNCHANGED
1902 005012 001401          BEQ      73$          ;BR IF OK
1903 005014 104010          ERROR    10          ;*****TEST 14 - ERROR 10*****
1904                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1905 005016                                     73$:
1906 005016 012700 017034    MOV      #BUF2,R0     ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1907 005022 105720          TSTB    (R0)+         ;TEST CONTENTS OF BOUNDARY
1908 005024 001401          BEQ      71$          ;BR, IF STILL ZERO
1909 005026 104011          ERROR    11          ;*****TEST 14 - ERROR 11*****
1910                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1911                                     ; SHOULD STILL EQUAL ZERO
1912 005030                                     71$:
1913 005030 016702 173614    MOV      DSTLN,R2     ;CALCULATE THE NUMBER OF "FILL"
1914 005034 166702 173604    SUB      SRCLN,R2     ;  CHARACTERS THAT SHOULD APPEAR IN DEST.
1915 005040 010201          MOV      R2,R1       ;STORE TRANSFER BYTE COUNT IN R1
1916 005042 122067 173606    114E12: CMPB   (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
1917 005046 001401          BEQ      +4          ;BR IF OK
1918 005050 104012          ERROR    12          ;*****TEST 14 - ERROR 12*****
1919                                     ;COMPARE ERROR IN DESTINATION
1920                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1921
1922 005052 005301          DEC      R1          ;DECREMENT BYTE COUNT
1923 005054 001372          BNE     T14E12       ;BR, IF NOT FINISHED CHECKING
1924 005056 016702 173564    MOV      SRCAD,R2     ;POINT R2 TO SOURCE STRING
1925 005062 016701 173556    MOV      SRCLN,R1     ;STORE TRANSFER BYTE COUNT IN R1
1926 005066 122022          114E13: CMPB   (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
1927 005070 001401          BEQ      +4          ;BR IF OK
1928 005072 104013          ERROR    13          ;*****TEST 14 - ERROR 13*****
1929                                     ;COMPARE ERROR IN DESTINATION
1930                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1931
1932 005074 005301          DEC      R1          ;DECREMENT BYTE COUNT
1933 005076 001373          BNE     T14E13       ;BR, IF NOT FINISHED CHECKING
1934 005100 105720          TSTB    (R0)+         ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1935 005102 001401          BEQ      ENDT14      ;BR, IF STILL ZERO
1936 005104 104014          ERROR    14          ;*****TEST 14 - ERROR 14*****
1937                                     ;UPPER BOUNDARY OF DEST. CHANGED
1938                                     ; SHOULD STILL EQUAL ZERO
1939
1940 005106          ENDT14:

```

CO-PROCESSOR UNIT WORKS...
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995

005106 000004
005110 004567 010076
005114 000020
005116 017054
005120 000007
005122 017035
005124 000377
005126 004767 010136
005132 004567 010130
005136 000020
005140 017054
005142 004567 010140
005146 000200
005150 004767 007772
005154 000277

005156 076031

005160 106767 173506
005164 042767 177400 173500
005172 023767 000574 173472
005200 001401
005202 104001

005204 026706 000676
005210 001403
005212 010637 000700
005216 104002

005220 026704 173430
005224 001401
005226 104003

005230 026705 173422
005234 001401
005236 104004

*TEST 15 TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB), SL .GT. DL. TRUNCATED.

*THIS TEST CHECKS "MOVE REVERSE" WHEN THE LEAST SIGNIFICANT
*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
*AND THE TRANSFER IS TRUNCATED.
*THE RESULT IS THE MOST SIGNIFICANT BYTES ARE TRUNCATED, R0 = THE
*NUMBER OF BYTES TRUNCATED. R1-->R3 EQUAL ZERO, AND ALL
*CONDITION CODES CLEAR.

*STIS: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
20 ;SOURCE LENGTH
BUF2+20 ;SOURCE ADDRESS
7 ;DESTINATION LENGTH
BUF2+1 ;DESTINATION ADDRESS
377 ;FILL CHARACTER
JSR PC,CLDST ;CLEAR DESTINATION AREA
JSR R5,GENSRC ;GENERATE A SOURCE STRING
.WORD 20
.WORD BUF2+27
JSR R5,XPS.. ;STORE EXPECTED PSW VALUE
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET ALL CONDITION CODES

MOVRC ;EXECUTE "MOVE REVERSE CHARACTER" INSTRUCTION

MFPS CCODES ;STORE THE PSW
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64\$;BR, IF EQUAL
ERROR 1 ;*****TEST 15 - ERROR 1*****
PSW ERROR
EXPECTED PSW IS STORED AT "EXPPSW"
ACTUAL PSW IS STORED AT "CCODES"

64\$:
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65\$;BR IF OK
MOV SP,#BADR6 ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 15 - ERROR 2*****
STACK POINTER NOT RESTORED BY INSTRUCTION
EXPECTED VALUE OF SP IS STORED AT "SAVR6"
ERRONEOUS VALUE IS AT "BADR6"

65\$:
CIV FILL,R4 ;CHECK R4 UNCHANGED
BEQ 66\$;BR IF OK
ERROR 3 ;*****TEST 15 - ERROR 3*****
R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"

66\$:
CMP TABLE,R5 ;CHECK R5 UNCHANGED
BEQ 67\$;BR IF OK
ERROR 4 ;*****TEST 15 - ERROR 4*****
R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

H04

TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB). SL .GT. DL. TRUNCATED

SEG 0047

```

1996 005240          67$:
1997 005240 016705 173400      MOV      SRCLN,R5      ;CHECK RO=SOURCE LENGTH MINUS
1998 005244 166705 173400      SLB      DSTLN,R5      ;  DESTINATION LENGTH
1999 005250 020005      CMP      RO,R5
2000 005252 001401      BEQ      68$
2001 005254 104005      ERROR    5      ;*****TEST 17 - ERROR 5*****
2002                                     ;RO NOT EQUAL TO (SRCLN-DSTLN)
2003 005256          68$:      ;CHECK OTHER GENERAL REGISTERS
2004 005256 005701      TST     R1      ;TEST R1
2005 005260 001401      BEQ     69$      ;BR, IF ZERO
2006 005262 104006      ERROR    6      ;*****TEST 15 - ERROR 5*****
2007                                     ;R1 SHOULD BE ZERO
2008 005264 005702          69$:      TST     R2      ;TEST R2
2009 005266 001401      BEQ     70$      ;BR IF ZERO
2010 005270 104007      ERROR    7      ;*****TEST 15 - ERROR 7*****
2011                                     ;R2 SHOULD BE ZERO
2012 005272 005703          70$:      TST     R3      ;TEST R3
2013 005274 001401      BEQ     71$      ;BR, IF ZERO
2014 005276 104010      ERROR    10     ;*****TEST 15 - ERROR 10*****
2015                                     ;R3 SHOULD BE ZERO
2016 005300          71$:
2017
2018 005300 012700 017034      MOV     #BUF2,RO      ;VERIFY DESTINATION AREA
2019 005304 105720      TSTB   (RO)+         ;POINT RO TO DESTINATION LOWER BYTE BOUNDARY
2020 005306 001401      BEQ     72$         ;TEST CONTENTS OF BOUNDARY
2021 005310 104011      ERROR    11        ;BR, IF STILL ZERO
2022                                     ;*****TEST 15 - ERROR 11*****
2023                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
2024                                     ; SHOULD STILL EQUAL ZERO
2025 005312          72$:
2026 005312 016702 173326      MOV     SRCLN,R2      ;CALCULATE ADDRESS OF MSB OF
2027 005316 166702 173326      SUB     DSTLN,R2      ;  THE SOURCE TO BE MOVED
2028 005322 066702 173320      ADD     SRCAD,R2
2029 005326 016701 173316      MOV     DSTLN,R1
2030 005332 122022          T15E12:  CMPB   (RO)+,(R2)+   ;STORE TRANSFER BYTE COUNT IN R1
2031 005334 001401      BEQ     +4           ;CHECK CHARACTERS IN DESTINATION
2032 005336 104012      ERROR    12        ;BR IF OK
2033                                     ;*****TEST 15 - ERROR 12*****
2034                                     ;COMPARE ERROR IN DESTINATION
2035                                     ;RO CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2036 005340 005301          DEC     R1           ;DECREMENT BYTE COUNT
2037 005342 001373          SNE     T15E12      ;BR, IF NOT FINISHED CHECKING
2038 005344 105720      TSTB   (RO)+         ;TEST CONTENTS OF DEST. UPPER BOUNDARY
2039 005346 001401      BEQ     ENDT15     ;BR, IF STILL ZERO
2040 005350 104013      ERROR    13        ;*****TEST 15 - ERROR 13*****
2041                                     ;UPPER BOUNDARY OF DEST. CHANGED
2042                                     ; SHOULD STILL EQUAL ZERO
2043 005352          ENDT15:

```

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

```

*****
*TEST 16 TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLEN .GT. DESTLEN. FILL
*****
*THIS TEST CHECKS "MOVE REVERSE" WHEN THE LEAST SIGNIFICANT
*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
*AND ALL BYTES ARE TRANSFERED.
*THE RESULT IS THE MSB OF THE DESTINATION ARE FILLS, R0-->R3 EQUAL
*ZERO, AND CONDITION CODES N,C=1 & Z,V=0.
*****
TS*16: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
10 ;SOURCE LENGTH
BUF2+20 ;SOURCE ADDRESS
15 ;DESTINATION LENGTH
BUF2+1 ;DESTINATION ADDRESS
377 ;FILL CHARACTER
JSR PC,COLDST ;CLEAR DESTINATION AREA
JSR R5,GENSRC ;GENERATE SOURCE STRING
.WORD 10
.WORD BUF2+20
JSR R5,XPSW ;STORE EXPECTED PSW VALUE
.WORD 211
JSR PC,GENR ;SET UP GENERAL REGISTERS
+CLN!CLC
+SEV!SEZ

MOVRC ;EXECUTE "MOVE REVERSE CHARACTER"

MFPS CCODES ;STORE THE PSW
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 16 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "CCODES"

64$:
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65$ ;BR IF OK
MOV SP,#BADSP ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 16 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
;ERRONEOUS VALUE IS AT "BADSP"

65$:
TST R0
BEQ 3$
ERROR 3 ;*****TEST 16 - ERROR 3*****
;R0 SHOULD BE ZERO

3$:
TST R1
BEQ 66$
ERROR 4 ;*****TEST 16 - ERROR 4*****

```

005352 000004
005354 004567 007632
005360 000010
005362 017054
005364 000015
005366 017035
005370 000377
005372 004767 007642
005376 004567 007664
005402 000010
005404 017054
005406 004567 007674
005412 000211
005414 004767 007526
005420 000251
005422 000266
005424 076031
005426 106767 173240
005432 042767 177400 173232
005440 023767 000674 173224
005446 001401
005450 104001
005452
005452 023706 000676
005456 001403
005460 010637 000700
005464 104002
005466
005466 005700
005470 001401
005472 104003
005474
005474 005701
005476 001401
005500 104004

J04

```

005502      005702      66$:      TST      R2
005504      001401          BEQ      67$
005506      104005          ERROR     5
                        :R1 SHOULD BE ZERO
                        :TEST R2
                        :BR IF ZERO
                        :*****TEST 16 - ERROR 5*****
005510      005703      67$:      TST      R3
005512      001401          BEQ      68$
005514      104006          ERROR     6
                        :R2 SHOULD BE ZERO
                        :TEST R3
                        :BR IF ZERO
                        :*****TEST 16 - ERROR 6*****
005516      005704      68$:      CMP      FILL,R4
005518      001401          BEQ      69$
005522      104007          ERROR     7
                        :CHECK R4 UNCHANGED
                        :BR IF OK
                        :*****TEST 16 - ERROR 7*****
005526      005705      69$:      CMP      TABLE,R5
005528      001401          BEQ      70$
005534      104010          ERROR     10
                        :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                        :CHECK R5 UNCHANGED
                        :BR IF OK
                        :*****TEST 16 - ERROR 10*****
                        :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
005536      012700      70$:      MOV      #BLF2,RD
005538      105720          TSTB     (RD)+
005544      001401          BEQ      71$
005546      104011          ERROR     11
                        :POINT RD TO DESTINATION LOWER BYTE BOUNDARY
                        :TEST CONTENTS OF BOUNDARY
                        :BR IF STILL ZERO
                        :*****TEST 16 - ERROR 11*****
                        :LOWER BOUNDARY OF DESTINATION CHANGED
                        : SHOULD STILL EQUAL ZERO
005550      016702      71$:      MOV      DESTN,R2
005552      166702      173074  SUB      SRCAL,R2
005554      166702      173064  MOV      R2,R1
005556      010201          :STORE TRANSFER BYTE COUNT IN R1
005558      122067      173066 716E12: CMPB     (RD)+,FILL
005566      001401          BEQ      +4
005570      104012          ERROR     12
                        :CHECK CHARACTERS IN DESTINATION
                        :BR IF OK
                        :*****TEST 16 - ERROR 12*****
                        :COMPARE ERROR IN DESTINATION
                        :RD CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
                        :DECREMENT BYTE COUNT
005572      005301          DEC      R1
005574      001372          BNE     T16E12
005576      016702      173044  MOV      SRCAL,R2
005582      016701      173036  MOV      SRCAL,R1
005586      122022      716E13: CMPB     (RD)+,(R2)+
005590      001401          BEQ      +4
005592      104013          ERROR     13
                        :STORE TRANSFER BYTE COUNT IN R1
                        :CHECK CHARACTERS IN DESTINATION
                        :BR IF OK
                        :*****TEST 16 - ERROR 13*****
                        :COMPARE ERROR IN DESTINATION
                        :RD CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
                        :DECREMENT BYTE COUNT
005614      005301          DEC      R1
005616      001373          BNE     T16E13
005620      105720          TSTB     (RD)+
005622      001401          BEQ      ENCT16
005624      104014          ERROR     14
                        :BR IF STILL ZERO
                        :*****TEST 16 - ERROR 14*****
                        :UPPER BOUNDARY OF DEST. CHANGED
                        : SHOULD STILL EQUAL ZERO
005626      ENCT16:

```

2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209

005626	000004		
005630	105777	172704	
005634	100571		
005636	026767	172706	173014
005644	001007		
005646	032767	000001	172732
005654	001403		
005656	005767	172712	
005662	001156		
005664			
005664	004567	007322	
005670	000374		
005672	016434		
005674	000376		
005676	017035		
005700	000377		
005702	004767	007332	
005706	012767	005774	172754
005714	012777	015356	172742
005722	005077	172740	
005726	004767	007400	
005732	013777	000554	172722
005740	004567	007342	
005744	000011		
005746	106427	000000	
005752	052777	000100	172700
005760	004767	007162	
005764	010637	000676	
005770	000251		
005772	000266		
005774	076031		
005776	106767	172670	
006002	032777	000100	172650
006010	001363		
006012	042767	177400	172652
006020	023767	000674	172644
006026	001401		
006030	104001		

```

*****
*TEST 17      TEST INTERRUPTABILITY OF "MOVRC" INSTRUCTION
*****
*THIS TEST INTERRUPTS THE EXECUTION OF THE "MOVRC"
*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
*INTERRUPT, AND VERIFIES THE RESULTS. THE PROPER
*RESULT IS ALL BYTES MOVED TO THE DESTINATION,
*RD-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1
*AND Z,V=0.
*****
TST17:  SCOPE
        TSTB      @SWR      ;TEST BIT 7 OF SWR
        BMI      TST20     ;SKIP TO NEXT TEST IF SET
        CMP      $TPS,TCSP ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
        BNE      T17CONT   ;BR. IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT      @BIT0,$ENV ;IF YES, CHECK IF ON APT
        BEQ      T17CONT   ;BR IF NOT UNDER APT, AND DO THIS TEST
        TST      $PASS     ;CHECK IF ON FIRST PASS
        BNE      TST20     ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T17CONT: JSR      R5,PREP   ;SET UP INSTRUCTION ARGUMENTS
        JSR      374      ;SOURCE LENGTH
        JSR      BUF1     ;SOURCE ADDRESS
        JSR      376      ;DESTINATION LENGTH
        JSR      BUF2+1   ;DESTINATION ADDRESS
        JSR      377      ;FILL CHARACTER
        JSR      PC,CLDST ;CLEAR DESTINATION
        MOV      @MRC,PCI  ;STORE PC OF TEST INSTRUCTION
        MOV      @INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
        CLR      @TPSW    ;ALLOW INTERRUPTS AFTER TTY INTERRUPT

        JSR      PC,TDONE  ;SEND CARRIAGE RETURN
        MOV      @M$NULL,@TBUF ;STORE EXPECTED PSW VALUE
        JSR      R5,XPSW

        .WORD    11
        MTPS    #0        ;SET PSW TO ALLOW INTERRUPTS
        BIS      #100,@TCSR ;ENABLE TTY INTERRUPTS
        JSR      PC,GENR   ;SET UP GENERAL REGISTERS
        MOV      SP,@$SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
        +CLN!CLC          ;CLEAR CONDITION CODES N&C
        +SEV!SEZ          ;CLEAR CONDITION CODES V&Z

MRC:    MOVRC           ;EXECUTE "MOVE REVERSE CHARACTER"

MFRS    CCODES         ;STORE THE PSW
        BIT      #100,@TCSR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BNE      REPMRC   ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
        BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP      @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$      ;BR, IF EQUAL
        ERROR    !       ;*****TEST 17 - ERROR 1*****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT "EXPPSW"

```

```

2210                                     ;ACTUAL PSW IS STORED AT "CCODES"
2211 006032 64$:
2212                                     ;CHECK RESULTS
2213 006032 023706 000676 CMP      @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2214 006036 001403 BEQ      65$ ;BR IF OK
2215 006040 010637 000700 MOV      SP,@#BADRE ;STORE BAD SP VALUE
2216 005044 104002 ERROR   2 ;*****TEST 17 - ERROR 2*****
2217                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2218                                     ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2219                                     ;ERRONEOUS VALUE IS AT "BADRE"
2220 006046 65$:
2221 006046 005700 TST      R0 ;CHECK R0=ZERO
2222 006050 001401 BEQ      1$
2223 006052 104003 ERROR   3 ;*****TEST 17 - ERROR 3*****
2224                                     ;R0 SHOULD BE ZERO
2225 006054 1$:
2226 006054 005701 TST      R1 ;CHECK OTHER GENERAL REGISTERS
2227 006056 001401 BEQ      66$ ;TEST R1
2228 006060 104004 ERROR   4 ;BR, IF ZERO
2229                                     ;*****TEST 17 - ERROR 4*****
2230 006062 005702 TST      R2 ;R1 SHOULD BE ZERO
2231 006064 001401 BEQ      67$ ;TEST R2
2232 006066 104005 ERROR   5 ;BR IF ZERO
2233                                     ;*****TEST 17 - ERROR 5*****
2234 006070 005703 TST      R3 ;R2 SHOULD BE ZERO
2235 006072 001401 BEQ      68$ ;TEST R3
2236 006074 104006 ERROR   6 ;BR, IF ZERO
2237                                     ;*****TEST 17 - ERROR 6*****
2238 006076 68$:
2239 006076 026704 172552 CMP      FILL,R4 ;CHECK R4 UNCHANGED
2240 006102 001401 BEQ      69$ ;BR IF OK
2241 006104 104007 ERROR   7 ;*****TEST 17 - ERROR 7*****
2242                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2243 006106 026705 172544 CMP      TABLE,R5 ;CHECK R5 UNCHANGED
2244 006112 001401 BEQ      70$ ;BR IF OK
2245 006114 104010 ERROR   10 ;*****TEST 17 - ERROR 10*****
2246                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2247 006116 70$:
2248 006116 012700 017034 MOV      #BUF2,R0 ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2249 006122 105720 TSTB   (R0)+ ;TEST CONTENTS OF BOUNDARY
2250 006124 001401 BEQ      71$ ;BR, IF STILL ZERO
2251 006126 104011 ERROR   11 ;*****TEST 17 - ERROR 11*****
2252                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
2253                                     ; SHOULD STILL EQUAL ZERO
2254 006130 71$:
2255 006130 016702 172514 MOV      DSTLN,R2 ;CALCULATE THE NUMBER OF FILL
2256 006134 166702 172504 SUB      SRCLN,R2 ;CHARACTERS THAT SHOULD APPEAR IN DEST.
2257 006140 010201 MOV      R2,R1 ;STORE TRANSFER BYTE COUNT IN R1
2258 006142 122067 172506 T17E12: CMPB   (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
2259 006146 001401 BEQ      +4 ;BR IF OK
2260 006150 104012 ERROR   12 ;*****TEST 17 - ERROR 12*****
2261                                     ;COMPARE ERROR IN DESTINATION
2262                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2263
2264 006152 005301 DEC      R1 ;DECREMENT BYTE COUNT
2265 006154 001372 BNE     T17E12 ;BR, IF NOT FINISHED CHECKING

```

2266	006156	016702	172464		MOV	SRCAD,R2		:POINT R2 TO SOURCE STRING
2267	006162	016701	172456		MOV	SRCLN,R1		:STORE TRANSFER BYTE COUNT IN R1
2268	006166	122022		TE13:	CMPB	(R0)+,(R2)+		:CHECK CHARACTERS IN DESTINATION
2269	006170	001401			BEQ	.+4		:BR IF OK
2270	006172	104013			ERROR	13		:*****TEST 17 - ERROR 13*****
2271								:COMPARE ERROR IN DESTINATION
2272								:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2273								
2274	006174	005301			DEC	R1		:DECREMENT BYTE COUNT
2275	006176	001373			BNE	TE13		:BR IF NOT FINISHED CHECKING
2276	006200	105720			TSTB	(R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
2277	006202	001401			BEQ	ENDT17		:BR IF STILL ZERO
2278	006204	104014			ERROR	14		:*****TEST 17 - ERROR 14*****
2279								:UPPER BOUNDARY OF DEST. CHANGED
2280								: SHOULD STILL EQUAL ZERO
2281	006206			ENDT17:				
2282	006206	106427	000200	40\$:	MTPS	#200		:RESTORE PSW TO PRIORITY 7
2283	006212	016777	172450	172444	MOV	TPSW,0TVECT		:RESTORE TRAP CATCHER

```

2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296 006220 000004
2297 006222 004567 006764
2298 006226 000000
2299 006230 177777
2300 006232 000000
2301 006234 177777
2302 006236 000377
2303 006240 004567 007042
2304 006244 000204
2305 006246 004767 006674
2306 006252 000277
2307 006254 000244
2308
2309 006256 076044
2310
2311 006260 106767 172406
2312 006264 042767 177400 172400
2313 006272 023767 000674 172372
2314 006300 0C14C1
2315 006302 104001
2316
2317
2318
2319 006304
2320 006304 023706 000676
2321 006310 001403
2322 006312 010637 000700
2323 006316 104002
2324
2325
2326
2327 006320
2328 006320 026700 172320
2329 006324 001401
2330 006326 104003
2331
2332 006330 026701 172312
2333 006334 001401
2334 006336 104004
2335
2336 006340 026702 172304
2337 006344 001401
2338 006346 104005
2339

```

```

*****
*TEST 20 TEST "CMPC" WITH SOURCE1 & SOURCE2 LENGTHS = 0
*****
*THIS TEST VERIFIES THAT "COMPARE CHARACTER" INDICATES EQUAL
*STRINGS WITH SOURCE LENGTHS EQUAL TO ZERO.
*THE RESULT IS R0-->R3 ARE UNCHANGED, AND ALL CONDITION
*CODES CLEAR EXCEPT Z=1.
*****
↑ST20: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      0          ;SOURCE1 LENGTH
      NXM       ;SOURCE1 ADDRESS
      0          ;SOURCE2 LENGTH
      NXM       ;SOURCE2 ADDRESS
      377       ;FILL CHARACTER
      JSR      R5,XPSW
      .WORD    204
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC          ;SET ALL CONDITION CODES, EXCEPT
      CLZ          ;CLEAR Z
      CMPC       ;EXECUTE "CMPC" INSTRUCTION
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ERROR    1          ;*****TEST 20 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT "EXPPSW"
                        ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @#SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      MOV      SP,@#BADR6   ;STORE BAD SP VALUE
      ERROR    2          ;*****TEST 20 - ERROR 2*****
                        ;STACK POINTER NOT RESTORED BY INSTRUCTION
                        ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                        ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      CMP      SRCLN,R0     ;CHECK R0 UNCHANGED
      BEQ      1$          ;
      ERROR    3          ;*****TEST 20 - ERROR 3*****
                        ;SRC1 LENGTH ERROR
                        ;CHECK R1 UNCHANGED
      1$:
      CMP      SRCAD,R1     ;
      BEQ      2$          ;
      ERROR    4          ;*****TEST 20 - ERROR 4*****
                        ;SRC1 ADDRESS ERROR
                        ;CHECK R2 UNCHANGED
      2$:
      CMP      CSTLN,R2     ;
      BEQ      3$          ;
      ERROR    5          ;*****TEST 20 - ERROR 5*****
                        ;SRC2 LENGTH ERROR

```

2340	006350	026703	172276	3\$:	CMP	DSTAD,R3	:CHECK R3 UNCHANGED
2341	006354	001401			BEQ	4\$	
2342	006356	104006			ERROR	6	:*****TEST 20 - ERROR 5*****
2343							:SRC2 ADDRESS ERROR
2344	006360			4\$:			
2345	006360	026704	172270		CMP	FILL,R4	:CHECK R4 UNCHANGED
2346	006364	001401			BEQ	66\$:BR IF OK
2347	006366	104007			ERROR	7	:*****TEST 20 - ERROR 7*****
2348							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2349	006370	026705	172262	66\$:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
2350	006374	001401			BEQ	67\$:BR IF OK
2351	006376	104010			ERROR	10	:*****TEST 20 - ERROR 10*****
2352							:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2353	006400			67\$:			

2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409

006400 000004
006402 004567 006604
006406 000000
006410 177777
006412 000004
006414 017034
006416 000377
006420 012700 017034
006424 012720 177777
006430 005020
006432 004567 006650
006436 000210
006440 004767 006502
006444 000277
006446 076044
006450 106767 172216
006454 042767 177400 172210
006462 023767 000674 172202
006470 001401
006472 104001
006474
006474 023706 000676
006500 001403
006502 010637 000700
006506 104002
006510
006510 005700
006512 001401
006514 104003
006516 026701 172124
006522 001401
006524 104004

```
*****  
*TEST 21 TEST "CMPC" WITH SOURCE1 LENGTH=0 & A NON-FILL CHARACTER IN SRC2  
*****  
*THIS TEST VERIFIES THAT "CMPC" COMPARES SOURCE2 WITH THE FILL  
*CHARACTER WHEN SOURCE1 LENGTH IS ZERO.  
*THE RESULT IS R0=0, R1 IS UNCHANGED, R2 EQUALS THE LENGTH  
*OF SOURCE2 SUB-STRING, R3 EQUALS THE POINTER TO SOURCE2  
*SUB-STRING, AND ALL CONDITION CODES CLEAR EXCEPT N=1.  
*****  
*ST21: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
0 ;SOURCE1 LENGTH  
NXM ;SOURCE1 ADDRESS  
4 ;SOURCE2 LENGTH  
BJF2 ;SOURCE2 ADDRESS  
377 ;FILL CHARACTER  
MOV #BUF2,R0 ;GENERATE SOURCE2 STRING STARTING  
MOV #177777,(R0)+ ;WITH TWO FILL BYTES  
CLR (R0)+ ;THEN TWO NON-FILL BYTES  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 210  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES  
CMPC ;EXECUTE "CMPC" INSTRUCTION  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 21 - ERROR 1*****  
 ;PSW ERROR  
 ;EXPECTED PSW IS STORED AT "EXPPSW"  
 ;ACTUAL PSW IS STORED AT "CCODES"  
64$:  
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 21 - ERROR 2*****  
 ;STACK POINTER NOT RESTORED BY INSTRUCTION  
 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
 ;ERRONEOUS VALUE IS AT "BADR6"  
55$:  
TST R0 ;CHECK R0=0  
BEQ 1$  
ERROR 3 ;*****TEST 21 - ERROR 3*****  
 ;SRC1 LENGTH ERROR  
1$:  
CMP SRCAD,R1 ;CHECK R1 UNCHANGED  
BEQ 2$  
ERROR 4 ;*****TEST 21 - ERROR 4*****  
 ;SRC1 ADDRESS ERROR
```

2410	006526	022702	000002	2\$:	CMP	#2,R2	:CHECK R2=2
2411	006532	001401			BEQ	3\$	
2412	006534	104005			ERROR	5	:*****TEST 21 - ERROR 5*****
2413							:SRC2 LENGTH ERROR
2414	006536	022703	017036	3\$:	CMP	#BUF2+2,R3	:CHECK R3=BUF2+2
2415	006542	001401			BEQ	4\$	
2416	006544	104006			ERROR	6	:*****TEST 21 - ERROR 6*****
2417							:SRC2 ADDRESS ERROR
2418	006546			4\$:			
2419	006546	026704	172102		CMP	FILL,R4	:CHECK R4 UNCHANGED
2420	006552	001401			BEQ	66\$:BR IF OK
2421	006554	104007			ERROR	7	:*****TEST 21 - ERROR 7*****
2422							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2423	006556	026705	172074	66\$:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
2424	006562	001401			BEQ	67\$:BR IF OK
2425	006564	104010			ERROR	10	:*****TEST 21 - ERROR 10*****
2426							:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2427	006566			67\$:			

2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

006566 000004
006570 004567 006416
006574 000007
006576 016434
006600 000012
006602 017034
006604 000377
006606 004767 006426
006612 004567 006450
006616 000012
006620 017034
006622 105037 017041
006626 004567 006454
006632 000210
006634 004767 006306
006640 000277

006642 076044

006644 106767 172022
006650 042767 177400 172014
006656 023767 000674 172006
006664 001401
006666 104001

006670
006670 023706 000676
006674 001403
006676 010637 000700
006702 104002

006704
006704 026704 171744
006710 001401
006712 104003

006714 026704 171736
006720 001401
006722 104004

```
*****  
*TEST 22 TEST "CMPC" WITH S1L .LT. S2L AND SRC1.GT.SRC2, NON-COMPARE IN SOURCE  
*****  
*THIS TEST VERIFIES THAT "CMPC" RETURNS THE CORRECT SUBSTRING IDENTIFIERS  
*WHEN SOURCE1 STRING AND SOURCE2 STRING DO NOT COMPARE.  
*THE RESULT IS R0 EQUALS SRC1 SUBSTRING LENGTH, R1 EQUALS  
*SRC1 SUBSTRING POINTER, R2 EQUALS SRC2 SUBSTRING LENGTH, R3 EQUALS  
*SRC2 SUBSTRING POINTER, AND ALL CONDITION CODES CLEAR EXCEPT N=1.  
*****  
*ST22: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
7 ;SOURCE1 LENGTH  
BUF1 ;SOURCE1 ADDRESS  
12 ;SOURCE2 LENGTH  
BUF2 ;SOURCE2 ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR SOURCE2 AREA  
JSR R5,GENSRC ;GENERATE SOURCE2 STRING  
.WORD 12  
.WORD BUF2  
CLRB Q#BUF2-5 ;CREATE A NON-COMPARE CHARACTER IN SRC2 STRING  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 210  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES  
  
CMPC ;EXECUTE "CMPC" INSTRUCTION  
  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP Q#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;-*****TEST 22 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
  
64$:  
CMP Q#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MVC SP,Q#BADR6 ;STORE BAD SP VALUE  
ERRCR 2 ;*****TEST 22 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
  
65$:  
CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 22 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
  
66$:  
CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERRCR 4 ;*****TEST 22 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
```

2484	006724			675:		
2485	006724	005067	171754	18:	CLR	TEMP
2486	006730	016704	171712		MOV	SRCAD,R4
2487	006734	016705	171712		MOV	DSTAD,R5
2488	006740	121415		28:	CMPB	(R4),(R5)
2489	006742	001004			BNE	3\$
2490	006744	005267	171734		INC	TEMP
2491	006750	122425			CMPB	(R4)+,(R5)+
2492	006752	000772			BR	2\$
2493	006754			33:		
2494	006754	020401			CMF	R4,R1
2495	006756	001401			BEG	4\$
2496	006760	104005			ERROR	5
2497						
2498						
2499						
2500	006762	020503		43:	CMF	R5,R3
2501	006764	001401			BEG	5\$
2502	006766	104006			ERROR	6
2503						
2504						
2505						
2506	006770	016704	171650	55:	MOV	SRCLN,R4
2507	006774	166704	171704		SUB	TEMP,R4
2508	007000	020400			CMP	R4,R0
2509	007002	001401			BEG	6\$
2510	007004	104007			ERROR	7
2511						
2512						
2513						
2514	007006	016704	171636	63:	MOV	DSTLN,R4
2515	007012	166704	171666		SUB	TEMP,R4
2516	007016	020402			CMP	R4,R2
2517	007020	001401			BEG	7\$
2518	007022	104010			ERROR	10
2519						
2520						
2521	007024	000240		73:	NCP	

```

:POINT R4 TO SOURCE1 ADDRESS
:POINT R5 TO SOURCE2 ADDRESS
:COMPARE SOURCES
:BR IF NOT EQUAL

:POINT STRINGS TO NEXT BYTES

:CHECK ADDRESS & LENGTH DESCRIPTORS
:CHECK R1, SOURCE1 ADDRESS
:BR IF OK
*****TEST 22 - ERROR 5*****
:SOURCE1 ADDRESS ERROR
:R4 CONTAINS THE EXPECTED ADDRESS POINTER
:R1 CONTAINS THE BAD VALUE
:CHECK R3, SOURCE2 ADDRESS
:BR IF OK
*****TEST 22 - ERROR 6*****
:SOURCE2 ADDRESS ERROR
:R5 CONTAINS THE EXPECTED ADDRESS POINTER
:R3 CONTAINS THE BAD VALUE
:CALCULATE LENGTH OF SRC1 REMAINDER

:CHECK R0, SOURCE1 LENGTH
:BR IF OK
*****TEST 22 - ERROR 7*****
:SOURCE1 LENGTH ERROR
:R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR
:R0 CONTAINS THE BAD VALUE
:CALCULATE LENGTH OF SRC2 REMAINDER

:CHECK R2, SOURCE2 LENGTH
:BR IF OK
*****TEST 22 - ERROR 10*****
:SOURCE2 LENGTH ERROR
:R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR
:R2 CONTAINS THE BAD VALUE
    
```

G05

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078

007026 000004
007030 004567 006156
007034 000007
007036 016434
007040 000007
007042 016434
007044 000377
007046 004567 006234
007052 000204
007054 004767 006066
007060 000277
007062 000244

007064 076044

007066 106767 171600
007072 042767 177400 171572
007100 023767 006674 171564
007106 001401
007110 104001

007112
007112 023706 000676
007116 001403
007120 010637 000700
007124 104002

007126
007126 105700
007130 001401
007132 104003

007134 105702
007136 001401
007140 104004

007142 016700 171500
007146 066700 171472
007152 020001
007154 001401
007156 104005

```
*****  
*TEST 23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS  
*****  
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS THE  
*INDICATION THAT SOURCE1 STRING EQUALS SOURCE2 STRING:  
*R0 & R2 EQUAL ZERO, R1 & R3 EQUAL THE ADDRESS+1 OF THE  
*LSB OF THE SOURCE1 & SOURCE2 STRINGS RESPECTIVELY,  
*AND ALL CONDITION CODES CLEAR, EXCEPT Z=1  
*****  
*S*23: SCOPE  
JSR RS,PREP ;SET JP INSTRUCTION ARGUMENTS  
7 ;SOURCE1 LENGTH  
BUFI ;SOURCE1 ADDRESS  
7 ;SOURCE2 LENGTH  
BUFI ;SOURCE2 ADDRESS  
277 ;FILL CHARACTER  
JSR RS,XPSW ;STORE EXPECTED PSW VALUE  
204  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET ALL CONDITION CODES, EXCEPT  
CLR ;CLEAR "Z"  
 ;EXECUTE "CMPC" INSTRUCTION  
  
 ;CHECK RESULTS  
MFPS ;STORE THE PSW  
CCODES ;CLEAR ALL UNUSED BITS  
BIC #177400,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
CMP #EXPPSW,CCODES ;BR, IF EQUAL  
BEQ 64$  
ERROR ;*****TEST 23 - ERROR 1*****  
 ;PSW ERROR  
 ;EXPECTED PSW IS STORED AT "EXPPSW"  
 ;ACTUAL PSW IS STORED AT "CCODES"  
  
64$:  
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 23 - ERROR 2*****  
 ;STACK POINTER NOT RESTORED BY INSTRUCTION  
 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
 ;ERRONEOUS VALUE IS AT "BADR6"  
  
65$:  
TSTB R0 ;CHECK R0  
BEQ 1$ ;BR, IF ZERO  
ERROR 3 ;*****TEST 23 - ERROR 3*****  
 ;R0 SHOULD BE ZERO  
  
1$:  
TSTB R2 ;CHECK R2  
BEQ 2$ ;BR, IF ZERO  
ERROR 4 ;*****TEST 23 - ERROR 4*****  
 ;R2 SHOULD BE ZERO  
  
2$:  
MOV SRCAD,R0 ;CALCULATE ADDRESS+1 OF LSB  
ADD SRCLN,R0  
CMP RC,R1  
BEQ 3$  
ERROR 5 ;*****TEST 23 - ERROR 5*****
```

H05

DATA.P11 MACY11 2710005 22-DEC-76 10:49 PAGE 59
22-DEC-76 10:49 723 TEST "CMP" FOR S1L=S2L. SOURCE1=SOURCE2 STRINGS

EEG 0000

007160	026703		38:	CMP	R3 R3	:R1 SHOULD EQUAL (SRCAD)+:SRCLN.
007162	026704			BFG	408	:CHECK R3
007164	104006			ERROR	6	:BR. IF EQUAL TO ADDRESS+1 OF SB
						:*****TEST 23 - ERROR 6*****
007166			408:			:R3 SHOULD EQUAL (DSTAD)+:DSTLN.
007168	026704	171462		CMP	FILL R4	:CHECK R4 UNCHANGED
007172	026705			BFG	568	:BR IF OK
007174	104007			ERROR	7	:*****TEST 23 - ERROR 7*****
						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
007176	026705	171464	668:	CMP	TABLE R5	:CHECK R5 UNCHANGED
007202	026706			BFG	578	:BR IF OK
007204	104008			ERROR	8	:*****TEST 23 - ERROR 8*****
						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
007206			678:			

EST "CMPR" WITH SOURCE2=0 & SOURCE1 CONTAINS ALL FILLS

SEG 0002

007344
007346
007352
007356
007362
007364
007366

007370
007372
007374
007376

007400
007404
007406

007410

016702
016704
016706
016708
016710
016712
016714

026704
026706
026708
026710

026706
026708
026710

104006

104007

104010
104012

104010

REC-76

124

25: MOV R0
26: MOV R1
27: MOV R2
28: MOV R3
29: MOV R4
30: MOV R5
31: MOV R6
32: MOV R7
33: MOV R8
34: MOV R9
35: MOV R10
36: MOV R11
37: MOV R12
38: MOV R13
39: MOV R14
40: MOV R15
41: MOV R16
42: MOV R17
43: MOV R18
44: MOV R19
45: MOV R20
46: MOV R21
47: MOV R22
48: MOV R23
49: MOV R24
50: MOV R25
51: MOV R26
52: MOV R27
53: MOV R28
54: MOV R29
55: MOV R30
56: MOV R31
57: MOV R32
58: MOV R33
59: MOV R34
60: MOV R35
61: MOV R36
62: MOV R37
63: MOV R38
64: MOV R39
65: MOV R40
66: MOV R41
67: MOV R42
68: MOV R43
69: MOV R44
70: MOV R45
71: MOV R46
72: MOV R47
73: MOV R48
74: MOV R49
75: MOV R50
76: MOV R51
77: MOV R52
78: MOV R53
79: MOV R54
80: MOV R55
81: MOV R56
82: MOV R57
83: MOV R58
84: MOV R59
85: MOV R60
86: MOV R61
87: MOV R62
88: MOV R63
89: MOV R64
90: MOV R65
91: MOV R66
92: MOV R67
93: MOV R68
94: MOV R69
95: MOV R70
96: MOV R71
97: MOV R72
98: MOV R73
99: MOV R74
100: MOV R75
101: MOV R76
102: MOV R77
103: MOV R78
104: MOV R79
105: MOV R80
106: MOV R81
107: MOV R82
108: MOV R83
109: MOV R84
110: MOV R85
111: MOV R86
112: MOV R87
113: MOV R88
114: MOV R89
115: MOV R90
116: MOV R91
117: MOV R92
118: MOV R93
119: MOV R94
120: MOV R95
121: MOV R96
122: MOV R97
123: MOV R98
124: MOV R99
125: MOV R100

: SOURCE1 ADDRESS ERROR
: R0 CONTAINS THE EXPECTED ADDRESS POINTER
: R1 CONTAINS THE BAD VALUE
: CHECK R2=0
: BR IF OK
: *****TEST 24 - ERROR 5*****
: SOURCE2 LENGTH ERROR
: R2 SHOULD BE ZERO

: CALCULATE ADDRESS OF LSB+1

: CHECK R3, SOURCE2 ADDRESS
: BR IF OK
: *****TEST 24 - ERROR 6*****
: SOURCE2 ADDRESS ERROR
: R2 CONTAINS THE EXPECTED ADDRESS POINTER
: R3 CONTAINS THE BAD VALUE

: CHECK R4 UNCHANGED
: BR IF OK
: *****TEST 24 - ERROR 7*****
: R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
: CHECK R5 UNCHANGED
: BR IF OK
: *****TEST 24 - ERROR 10*****
: R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

K05

2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734

007410 000004
007412 004567 005574
007416 000004
007420 017034
007422 000000
007424 177777
007426 000377
007430 012700 017034
007434 012720 177777
007440 005020
007442 004567 005640
007446 000201
007450 004767 005472
007454 000277
007456 000250
007460 076044
007462 106767 171204
007466 042767 177400 171176
007474 023767 000674 171170
007502 001401
007504 104001
007506
007506 023706 000676
007512 001403
007514 010637 000700
007520 104002
007522
007522 022700 000002
007526 001401
007530 104003
007532 022701 017036
007536 001401
007540 104004
007542 005702
007544 001401

```
*****
*TEST 25 TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO
*****
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS R1&R0
*EQUAL ADDRESS & LENGTH OF UNEQUAL SOURCE1 STRING,
*R2 EQUALS SOURCE2 ADDRESS, R2 EQUALS ZERO, AND ALL
*CONDITION CODES CLEAR., EXCEPT N=1
*****
S*25: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGJMENTS
4 ;SOURCE1 LENGTH
BUF2 ;SOURCE1 ADDRESS
0 ;SOURCE2 LENGTH
NXM ;SOURCE2 ADDRESS
377 ;FILL CHARACTER
MOV #BUF2,R0 ;GENERATE SOURCE1 STRING STARTING
MOV #177777,(R0)+ ;WITH TWO FILL BYTES
CLR (R0)+ ;THEN TWO "ZERO" BYTES
JSR R5,XPSW ;STORE EXPECTED PSW VALUE
WORD 201
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET ALL CONDITION CODES, EXCEPT
CLN ; CLEAR N=1
CMPC ;EXECUTE "CMPC" INSTRUCTION

;CHECK RESULTS
MFP5 C00DES ;STORE THE PSW
BIC #177400,C00DES ;CLEAR ALL UNUSED BITS
CMP @#EXPPSW,C00DES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 25 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "C00DES"

64$:
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65$ ;BR IF OK
MOV SP,@#BADR6 ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 25 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
;ERRONEOUS VALUE IS AT "BADR6"

65$:
CMP #2,R0 ;CHECK R0=2
BEQ 1$
ERROR 3 ;*****TEST 25 - ERROR 3*****
;SRC1 LENGTH ERROR
;CHECK R1=BUF2+2

1$:
CMP #BUF2+2,R1
BEQ 2$
ERROR 4 ;*****TEST 25 - ERROR 4*****
;SRC1 ADDRESS ERROR
;CHECK R2=0

2$:
TST R2
BEQ 3$
```

L05

MAIN. MACY11 271006) 22-DEC-76 10:49 PAGE 63

CVKRA.F11 22-DEC-76 10:47 725

TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO

SEG 0024

2735	007546	104005			ERROR	5		:*****TEST 25 - ERROR 5*****
2736								:SRC2 LENGTH ERROR
2737	007550	026703	171075	3\$:	CMP	DSTAD,R3		:CHECK R3=SRC2 ADDRESS
2738	007554	001401			BEG	4\$		
2739	007556	104006			ERROR	6		:*****TEST 25 - ERROR 6*****
2740								:SRC2 ADDRESS ERROR
2741	007560			4\$:				
2742	007560	026704	171070		CMP	FILL,R4		:CHECK R4 UNCHANGED
2743	007564	001401			BEG	66\$:BR IF OK
2744	007566	104007			ERROR	7		:*****TEST 25 - ERROR 7*****
2745								:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2746	007570	026705	171062	66\$:	CMP	TABLE,R5		:CHECK R5 UNCHANGED
2747	007574	001401			BEG	67\$:BR IF OK
2748	007576	104010			ERROR	10		:*****TEST 25 - ERROR 10*****
2749								:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2750	007600			67\$:				

M05

2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806

007600 000004
007602 004567 005404
007606 000007
007610 017034
007612 000003
007614 015434
007616 000377
007620 004567 005442
007624 000007
007626 017034
007630 105037 017036
007634 004567 005446
007640 000211
007642 004767 005300
007646 000251
007650 000266
007652 076044
007654 106767 171012
007660 042767 177400 171004
007666 023767 000674 170776
007674 001401
007676 104001
007700
007700 023706 000676
007704 001403
007706 010637 000700
007712 104002
007714
007714 026704 170734
007720 001401
007722 104003
007724 026705 170726
007730 001401
007732 104004

```
*****  
*TEST 26 TEST "CMPC" FOR SOURCE1 LESS THAN SOURCE2  
*****  
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS  
*R0&R2 EQUAL THE REMAINING LENGTH OF THE UNEQUAL BYTES  
*OF THE SOURCE1&SOURCE2 STRINGS, R1&R3 EQUAL THE ADDRESS  
*OF THE FIRST UNEQUAL BYTES IN THE SOURCE1 AND SOURCE2  
*STRINGS, AND CONDITION CODE C,N=1 AND V,Z=0  
*****  
†ST26: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
7 ;SOURCE1 LENGTH  
BUF2 ;SOURCE1 ADDRESS  
3 ;SOURCE2 LENGTH  
BUF1 ;SOURCE2 ADDRESS  
377 ;FILL CHARACTER  
JSR R5,GENSRC ;GENERATE SOURCE1 STRING  
.WORD 7 ;STRING LENGTH  
.WORD BUF2 ;STRING ADDRESS  
CLRB @#BUF2+2 ;PLACE A NON-COMPARE CHARACTER IN SOURCE1 STRING  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 211  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLC!C,N ;CLEAR CONDITION CODES C&N  
+SEZ!SEV ;SET CONDITION CODES Z&V  
;EXECUTE "CMPC" INSTRUCTION  
CMPC ;CHECK RESULT  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 26 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$:  
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
EPROR 2 ;*****TEST 26 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
65$:  
CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 26 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
66$:  
CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 26 - ERROR 4*****
```

N05

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 65
 CVKATA.P11 22-DEC-76 10:47 T26

TEST "CMPC" FOR SOURCE1 LESS THAN SOURCE2

SEQ 0066

```

2807                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2808 007734                               67$:
2809 007734                               1$:
2810 007734 005067 170744                 CLR     TEMP
2811 007740 016704 170702                 MOV     SRCAD,R4           ;POINT R4 TO SOURCE1 ADDRESS
2812 007744 016705 170702                 MOV     DSTAD,R5           ;POINT R5 TO SOURCE2 ADDRESS
2813 007750 121415                         2$:  CMPB   (R4),(R5)         ;COMPARE SOURCES
2814 007752 001004                         BNE     3$                 ;BR, IF NOT EQUAL
2815 007754 005267 170724                 INC     TEMP
2816 007760 122425                         CMPB   (R4)+,(R5)+         ;POINT STRINGS TO NEXT BYTES
2817 007762 000772                         BR      2$
2818 007764                               3$:
2819 007764 020401                         CMP     R4,R1              ;CHECK R1 (SOURCE1 STRING)
2820 007766 001401                         BEQ     4$
2821 007770 104005                         ERROR   5                   ;*****TEST 26 - ERROR 5*****
2822                                     ;SRC1 ADDRESS ERROR
2823 007772 020503                         4$:  CMP     R5,R3              ;CHECK R3 (SOURCE2 STRING)
2824 007774 001401                         BEQ     5$                 ;BR IF EQUAL
2825 007776 104006                         ERROR   6                   ;*****TEST 26 - ERROR 6*****
2826                                     ;SRC2 ADDRESS ERROR
2827 010000 016704 170640                         5$:  MOV     SRCLN,R4           ;CALCULATE LENGTH OF SRC1 REMAINDER
2828 010004 166704 170674                         SUB     TEMP,R4
2829 010010 020400                         CMP     R4,R0              ;CHECK R0 (SRC1 LENGTH)
2830 010012 001401                         BEQ     6$                 ;BR, IF EQUAL
2831 010014 104007                         ERROR   7                   ;*****TEST 26 - ERROR 7*****
2832                                     ;SRC1 STRING LENGTH ERROR
2833 010016 016704 170626                         6$:  MOV     DSTLN,R4           ;CALCULATE LENGTH OF SRC2 REMAINDER
2834 010022 166704 170656                         SUB     TEMP,R4
2835 010026 020402                         CMP     R4,R2              ;CHECK R2 (SRC2 LENGTH)
2836 010030 001401                         BEQ     40$                ;BR, IF EQUAL
2837 010032 104010                         ERROR   10                  ;*****TEST 26 - ERROR 10*****
2838                                     ;SRC2 STRING LENGTH ERROR
2839 010034                               40$:
  
```

```

2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850 010034 000004
2851 010036 004567 005150
2852 010042 000010
2853 010044 016434
2854 010046 000012
2855 010050 017034
2856 010052 000377
2857 010054 004767 005160
2858 010060 004567 005202
2859 010064 000010
2860 010066 017034
2861 010070 012721 177777
2862 010074 004567 005206
2863 010100 000204
2864 010102 004767 005040
2865 010106 000277
2866 010110 000244
2867
2868 010112 076044
2869
2870 010114 106767 170552
2871 010120 042767 177400 170544
2872 010126 023767 000674 170536
2873 010134 001401
2874 010136 104001
2875
2876
2877
2878 010140
2879 010140 023706 000676
2880 010144 001403
2881 010146 010637 000700
2882 010152 104002
2883
2884
2885
2886 010154
2887 010154 005700
2888 010156 001401
2889 010160 104003
2890
2891 010162 016700 170460
2892 010166 066700 170452
2893 010172 020001
2894 010174 001401
2895 010176 104004

```

```

*****
*TEST 27 TEST "CMPC" WITH SOURCE1 LENGTH .GT. SOURCE2 LENGTH & SOURCE1=SOURCE2
*****
*THIS TEST VERIFIES THAT "CMPC" INDICATES EQUAL STRINGS WHEN THE
*STRING LENGTHS ARE NOT EQUAL BUT THE EXCESS OF THE LONGER
*STRING CONTAINS ALL FILLS.
*****
TST27: SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        10          ;SOURCE1 LENGTH
        BUF1        ;SOURCE1 ADDRESS
        12          ;SOURCE2 LENGTH
        BUF2        ;SOURCE2 ADDRESS
        377        ;FILL CHARACTER
        JSR      PC,CLDST        ;CLEAR SOURCE2 AREA
        JSR      R5,GENSRC       ;GENERATE SOURCE2 STRING
        .WORD    10             ;MAKE FIRST 10 BYTES IDENTICAL
        .WORD    BUF2          ;TO SOURCE1 STRING AND THE
        MOV      #177777,(R1)+   ;LAST TWO BYTES EQUAL TO FILL CHARACTER
        JSR      R5,XPSW        ;STORE EXPECTED PSW VALUE
        .WORD    204
        JSR      PC,GENR        ;SET UP GENERAL REGISTERS
        SCC                      ;SET ALL CONDITION CODES, EXCEPT
        CLZ                      ;CLEAR Z
        CMPC                     ;EXECUTE "CMPC" INSTRUCTION
        MFPS      CCODES        ;STORE THE PSW
        BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$           ;BR, IF EQUAL
        ERROR    1             ;*****TEST 27 - ERROR 1*****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT "EXPPSW"
        ;ACTUAL PSW IS STORED AT "CCODES"
64$:
        CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
        BEQ      65$           ;BR IF OK
        MOV      SP,@#BADR6     ;STORE BAD SP VALUE
        ERROR    2             ;*****TEST 27 - ERROR 2*****
        ;STACK POINTER NOT RESTORED BY INSTRUCTION
        ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
        ;ERRONEOUS VALUE IS AT "BADR6"
65$:
        TST      R0              ;CHECK R0=0
        BEQ      1$
        ERROR    3
        ;*****TEST 27 - ERROR 3*****
        ;SOURCE1 LENGTH ERROR
        ;CALCULATE ADDRESS OF LSB+1
1$:
        MOV      SRCAD,R0
        ADD      SRCLN,R0
        CMP      R0,R1
        BEQ      2$
        ERROR    4
        ;*****TEST 27 - ERROR 4*****

```

```

2896                                     ;SOURCE1 ADDRESS ERROR
2897 010200 005702                       ;CHECK R2=0
      2$:  TST  R2
2898 010202 001401                       3$:
2899 010204 104005                       ;*****TEST 27 - ERROR 5*****
      3$:  BEQ  5
2900                                     ;SOURCE2 LENGTH ERROR
2901 010206 016702 170440                 ;CALCULATE ADDRESS OF LSB+1
2902 010212 066702 170432                 MOV  DSTAD,R2
2903 010216 020203                       ADD  DSTLN,R2
2904 010220 001401                       CMP  R2,R3
2905 010222 104006                       ;CHECK R3,SOURCE2 ADDRESS
      4$:  BEQ  4$
2906                                     ;*****TEST 27 - ERROR 6*****
2907 010224                               ;SOURCE2 ADDRESS ERROR
2908 010224 026704 170424                 ;CHECK R4 UNCHANGED
2909 010230 001401                       ;BR IF OK
2910 010232 104007                       ;*****TEST 27 - ERROR 7*****
2911                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2912 010234 025705 170416                 66$:  CMP  TABLE,R5
2913 010240 001401                       ;CHECK R5 UNCHANGED
2914 010242 104010                       ;BR IF OK
2915                                     ;*****TEST 27 - ERROR 10*****
2916 010244                               ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
    
```

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050

010244 000004
010246 004567 004740
010252 016434
010254 000010
010256 017034
010260 000010
010262 000377
010264 012737 076045 010312
010272 013767 000010 170406
010300 012737 010322 000010
010306 004767 004634

010312 076045

010314 016700 177772
010320 104001

010322 012626
010324 005267 177762
010330 022767 076050 177754
010336 001363
010340 016737 170342 000010

```
*****  
*TEST 30 TEST THAT BAD "SEARCH" OPCODES TRAP  
*****  
*THIS TEST VERIFIES THAT OPCODES 076045-->076047 TRAP TO  
*LOCATION 10.  
*****  
*ST30: SCOPE  
JSR RS,PREP ;SET UP INSTRUCTION ARGUMENTS  
BUF1 ;SOURCE LENGTH  
10 ;SOURCE ADDRESS  
BUF2 ;DESTINATION LENGTH  
10 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
MOV #076045,0#BD30 ;STORE THE FIRST BAD MOVE OPCODE  
MOV 0#10,TEMP1 ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR  
MOV #T30CONT,0#10 ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST  
REF30: JSR PC,GENR ;SET UP GENERAL REGISTERS  
  
BD30: .WORD 076045 ;EXECUTE BAD SEARCH INSTRUCTION  
  
MOV BD30,R0 ;STORE BAD OPCODE THAT DID NOT TRAP  
ERROR 1 ;*****TEST 30 - ERROR 1*****  
;BAD SEARCH OPCODE DID NOT TRAP  
;R0 CONTAINS THE BAD OPCODE  
  
T30CONT:MOV (SP)+,(SP)+ ;RESTORE THE STACK POINTER AFTER THE TRAP  
INC R030 ;INCREMENT INSTRUCTION OPCODE  
CMP #076050,R030 ;FINISHED WITH BAD SEARCH OPCODES?  
BNE REF30 ;BR IF NOT  
MOV TEMP1,0#10 ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
```

2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006

010346 000004
010350 105777 170164
010354 100537
010356 026767 170166 170274
010364 001007
010366 032767 000001 170212
010374 001403
010376 005767 170172
010402 001124
010404
010404 004567 004602
010410 000400
010412 016434
010414 000400
010416 016434
010420 000377
010422 012767 010510 170240
010430 012777 015356 170226
010436 005077 170224
010442 004767 004664
010446 013777 000554 170206
010454 004567 004626
010460 000004
010462 106427 000000
010466 052777 000100 170164
010474 004767 004446
010500 010637 000676
010504 000277
010506 000244
010510 076044
010512 106767 170154
010516 032777 000100 170134
010524 001363
010526 042767 177400 170136
010534 023767 000674 170130
010542 001401
010544 104001

*TEST 31 TEST INTERRUPTABILITY OF "CMPC" INSTRUCTION

*THIS TEST INTERRUPTS THE EXECUTION OF THE "CMPC"
*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
*INTERRUPT, VERIFIES THE RESULT. THE PROPER RESULT
*IS FOR SOURCE1=SOURCE2.

TST31: SCOPE
TSTB @SWR ;TEST BIT 7 OF SWR
BMI TST32 ;SKIP TO NEXT TEST IF SET
CMP \$TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T31CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BITC, \$ENV ;IF YES, CHECK IF ON APT
BEQ T31CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
TST \$PASS ;CHECK IF ON FIRST PASS
BNE TST32 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST
T31CONT:
JSR RS, PREP ;SET UP INSTRUCTION ARGUMENTS
400 ;SOURCE1 LENGTH
BUF1 ;SOURCE1 ADDRESS
400 ;SOURCE2 LENGTH
BUF1 ;SOURCE2 ADDRESS
377 ;FILL CHARACTER
MOV @CMC, PC ;STORE PC OF TEST INSTRUCTION
MOV @INTA, @TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC, TDONE
MOV @NULL, @TBUF ;SEND CARRIAGE RETURN
JSR RS, XPSW ;STORE EXPECTED PSW VALUE
.WORD 04
MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
BIS #100, @TCSR ;ENABLE TTY INTERRUPTS
REPCMC: JSR PC, GENR ;SET UP GENERAL REGISTERS
MOV SP, @SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
SCC ;SET ALL CONDITION CODES, EXCEPT
CLZ ; CLEAR Z=1
CMC: CMPC ;EXECUTE "CMPC" INSTRUCTION
MFPS CCODES ;CHECK RESULTS
BIT #100, @TCSR ;STORE THE PSW
BNE REPCMC ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
BIC #177400, CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
CMP @EXPPSW, CCODES ;CLEAR ALL UNUSED BITS
BEQ 64\$;CHECK PSW AGAINST EXPECTED VALUE
ERROR 1 ;BR, IF EQUAL
*****TEST 31 - ERROR 1*****
PSW ERROR
EXPECTED PSW IS STORED AT "EXPPSW"
ACTUAL PSW IS STORED AT "CCODES"

6-3:

TEST INTERRUPTABILITY OF "CMPC" INSTRUCTION

3007	010546	023706	000676		CMP	2#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
3008	010552	001403			BEG	65\$:BR IF OK
3009	010554	010637	000700		MOV	SP,2#BADR6	:STORE BAD SP VALUE
3010	010560	104002			ERROR	2	:*****TEST 31 - ERROR 2*****
3011							:STACK POINTER NOT RESTORED BY INSTRUCTION
3012							:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3013							:ERRONEOUS VALUE IS AT "BADR6"
3014	010562			65\$:			
3015	010562	105700			TSTB	R0	:CHECK R0
3016	010564	001401			BEG	1\$:BR, IF ZERO
3017	010566	104003			ERROR	3	:*****TEST 31 - ERROR 3*****
3018							:R0 SHOULD BE ZERO
3019	010570	105702		1\$:	TSTB	R2	:CHECK R2
3020	010572	001401			BEG	2\$:BR, IF ZERO
3021	010574	104004			ERROR	4	:*****TEST 31 - ERROR 4*****
3022							:R2 SHOULD BE ZERO
3023	010576	016700	170044	2\$:	MOV	SRCAD,R0	:CALCULATE ADDRESS+1 OF LSB
3024	010602	066700	170036		ADD	SRCLN,R0	
3025	010606	020001			CMP	R0,R1	:CHECK R1
3026	010610	001401			BEG	3\$:BR, IF EQUAL TO ADDRESS+1 OF LSB
3027	010612	104005			ERROR	5	:*****TEST 31 - ERROR 5*****
3028							:R1 SHOULD EQUAL (SRCAD)+(SRCLN)
3029	010614	020003		3\$:	CMP	R0,R3	:CHECK R3
3030	010616	001401			BEG	40\$:BR, IF EQUAL TO ADDRESS+1 OF LSB
3031	010620	104006			ERROR	6	:*****TEST 31 - ERROR 6*****
3032							:R3 SHOULD EQUAL (DSTAD)+(DSTLN)
3033	010622			40\$:			
3034	010622	026704	170026		CMP	FILL,R4	:CHECK R4 UNCHANGED
3035	010626	001401			BEG	66\$:BR IF OK
3036	010630	104007			ERROR	7	:*****TEST 31 - ERROR 7*****
3037							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3038	010632	026705	170020	66\$:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
3039	010636	001401			BEG	67\$:BR IF OK
3040	010640	104010			ERROR	10	:*****TEST 31 - ERROR 10*****
3041							:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3042	010642			67\$:			
3043	010642	106427	000200		MTPS	#200	:RESTORE PSW TO PRIORITY 7
3044	010646	016777	170014 170010		MOV	*PSW,STVECT	:RESTORE TRAP CATCHER

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100

```

*****
*TEST 32 TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
*ADDRESS OF THE FIRST OCCURRENCE OF THE "CHARACTER"
*IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH
*OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR.
*****
+ST32: SCOPE
JSR RS,PREP ;SET UP INSTRUCTION ARGUMENTS
10 ;SOURCE LENGTH
BUF: ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
4 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
JSR RS,XPSW ;STORE EXPECTED PSW VALUE
WORD EQU ;
JSR PC,GENR ;SET UP GENERAL REGISTERS
SOC ;SET ALL CONDITION CODES
;EXECUTE "LOCC"
010654 000004
010656 004567 004330
010662 000010
010664 016434
010666 177777
010670 177777
010672 000004
010674 004567 004406
010700 000200
010702 004757 004240
010706 000277
010710 076040
010712 106767 167754
010716 042767 177400 167746
010724 023767 000674 167746
010732 001401
010734 104001
010736 000676 643:
010742 001403
010744 010637 000700
010750 104002
010752 000676 653:
010756 001401
010760 174003
010762 026735 167670 663:
010766 001401
010770 104004
010772 016703 167650 673:
010776 016702 167640
010782 161367 167640 683:
010786 001403

```

```

;CHECK RESULTS
;STORE THE PSW
;CLEAR ALL UNUSED BITS
;CHECK PSW AGAINST EXPECTED VALUE
;BR, IF EQUAL
*****TEST 32 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "XPPSW"
;ACTUAL PSW IS STORED AT "CCODES"
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
;STORE BAD SP VALUE
*****TEST 32 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
;ERRONEOUS VALUE IS AT "BADR6"
;CHECK R4 UNCHANGED
;BR IF OK
*****TEST 32 - ERROR 3*****
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
;CHECK R5 UNCHANGED
;BR IF OK
*****TEST 32 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
;LOCATE "CHARACTER"

```


011032 000004
011034 004567
011040 100010
011042 016434
011044 177777
011046 177777
011050 000004
011052 004567
011056 000210
011060 004767
011064 000277
011066 076040
011070 106767
011074 042767
011102 023767
011110 001401
011112 104001
011114 022706
011120 001403
011122 010637
011126 104002
011130 026704
011134 001401
011136 104003
011140 026705
011144 001401
011146 104004
011150 016703
011154 016704
011158 016705
011162 016706

```
*****  
:TEST 33 TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE. BL LGT. 100000  
*****  
:PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE  
:ADDRESS OF THE FIRST OCCURRENCE OF THE "CHARACTER"  
:IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH  
:OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR  
:EXCEPT N=1.  
*****  
:*****
```

```
5733: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
100010 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
4 ;SEARCH (MASK FOR SCANC/SPANC) CHARACTER  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
210  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SOC ;SET ALL CONDITION CODES  
;EXECUTE "LOCC"  
LOCC  
;CHECK RESULTS  
MFPS COODES ;STORE THE PSW  
BIC #177400,COODES ;CLEAR ALL UNUSED BITS  
CMP #EXPPSW,COODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 643 ;BR, IF EQUAL  
ERROR 1  
;*****TEST 33 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "COODES"  
643: CMP #SAVRE,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 558 ;BR IF OK  
MOV SP,#BADRE ;STORE BAD SP VALUE  
ERROR 2  
;*****TEST 33 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVRE"  
;ERRONEOUS VALUE IS AT "BADRE"  
658: CMP R4,R4 ;CHECK R4 UNCHANGED  
BEQ 668 ;BR IF OK  
ERROR 3  
;*****TEST 33 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
668: CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 678 ;BR IF OK  
ERROR 4  
;*****TEST 33 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"  
678: MOV SACAD,R3 ;LOCATE "CHARACTER"  
MOV SACLN,R2  
688: CMPB #R3,FILL  
BEQ 698
```

J06

22-DEC-76 10:49 PAGE 74
22-DEC-76 10:47 733

TEST "L000" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE. SL .ST. 100000 SEQ 0079

011166 105723
011170 005302
011172 001372
011174 020200
011176 001401
011200 104005
011202 020301
011204 001401
011206 104005
011210

593:

703:

713:

TSTB (R3)+
DEC R2
BNE 58\$
CMP R2,R0
BEG 70\$
ERROR 5
CMP R3,R1
BEG 71\$
ERROR 6

:POINT TO NEXT BYTE
:DECREMENT BYTE COUNT
:BR, IF NOT FINISHED
:CHECK R0=REMAINING SOURCE LENGTH
:*****TEST 33 - ERROR 5*****
:LOCATE LENGTH ERROR
:CHECK R1=ADDRESS OF CHARACTER
:*****TEST 33 - ERROR 5*****
:LOCATE ADDRESS ERROR

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044

```

:*****
:*TEST 34      TEST "LOCC" WITH SEARCH CHARACTER NOT IN SOURCE
:*****
:*PROPER TERMINATION OF THIS TEST IS R0 EQUALS ZERO, R1
:*EQUALS ADDRESS OF LSB OF SOURCE PLUS ONE, AND ALL
:*CONDITION CODES CLEAR EXCEPT Z=1.
:*****
↑ST34: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGJMENTS
      BJJ      377          ;SOURCE LENGTH
      BJJ      000000       ;SOURCE ADDRESS
      NXM      177777      ;STORE NON-ZERO VALUES TO TEST R2 & R3
      NXM      177777      ; UNAFFECTED BY DIS STRING INSTRUCTIONS
      C        000000       ;SEARCH IMASK FOR SCANC SPANC CHARACTER
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD   204
      JSR      PC,GENR     ;SET UP GENERAL REGISTERS
      CLZ          ;CLEAR CONDITION CODE Z
      +SEV!SEN!SEZ       ;SET ALL OTHER CONDITION CODES
                          ;EXECUTE "LOCC"
      LOCC
                          ;CHECK RESULTS
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      2*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$         ;BR, IF EQUAL
      ERROR   1           ;*****TEST 34 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "EXPPSW"
                          ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      2*SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$         ;BR IF OK
      MOV      SP,2*BADR6   ;STORE BAD SP VALUE
      ERROR   2           ;*****TEST 34 - ERROR 2*****
                          ;STACK POINTER NOT RESTORED BY INSTALCTION
                          ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                          ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      CMP      FILL,R4     ;CHECK R4 UNCHANGED
      BEQ      66$         ;BR IF OK
      ERROR   3           ;*****TEST 34 - ERROR 3*****
                          ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      66$:
      CMP      TABLE,R5   ;CHECK R5 UNCHANGED
      BEQ      67$         ;BR IF OK
      ERROR   4           ;*****TEST 34 - ERROR 4*****
                          ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV      SRCAD,R3     ;LOCATE "CHARACTER"
      MOV      SRCLN,R2
      68$:
      CMPB    (R3),FILL
      BEQ      69$

```

```

011210 000004
011212 004567 003774
011216 000377
011220 016434
011222 177777
011224 177777
011226 000000
011230 004567 004052
011234 000204
011236 004767 003704
011242 000244
011244 000276
011246 076040
011250 106767 167416
011254 042767 177400 167410
011252 023767 000674 167402
011270 001401
011272 104001
011274
011274 023706 000676 64$:
011300 001403
011302 010637 000700
011306 104002
011310
011310 026704 167340
011314 001401
011316 104003
011320 026705 167332
011324 001401
011326 104004
011330
011330 016703 167312
011334 016702 167304
011340 121067 167310
011344 001403

```

000037	011346	105723
000038	011350	005302
000039	011352	001372
000040	011354	020200
000041	011356	001401
000042	011360	104005
000043		
000044	011362	020301
000045	011364	001401
000046	011366	104006
000047		
000048	011370	

695:

705:

715:

```

TSTB (R3)+
DEC R2
BNE 685
CMP R2,R0
BEG 705
ERROR 5

CMP R3,R1
BEG 715
ERROR 6

```

```

:POINT TO NEXT BYTE
:DECREMENT BYTE COUNT
:BR, IF NOT FINISHED
:CHECK R0=REMAINING SOURCE LENGTH

:*****TEST 34 - ERROR 5*****
:LOCATE LENGTH ERROR
:CHECK R1=ADDRESS OF CHARACTER

:*****TEST 34 - ERROR 6*****
:LOCATE ADDRESS ERROR

```

3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299

011370 000004
011372 004567 003614
011376 000000
011400 177777
011402 177777
011404 177777
011406 000001
011410 004567 003672
011414 000204
011416 004767 003524
011422 000244
011424 000273
011426 076040
011430 106767 167236
011434 042767 177400 167230
011442 023767 000674 167222
011450 001401
011452 104001
011454
011454 023706 000676
011460 001403
011462 010637 000700
011466 104002
011470
011470 105700
011472 001401
011474 104003
011476 026701 167144
011502 001401
011504 104004
011506

```
*****  
*TEST 35 TEST "LOCC" WITH ZERO SOURCE LENGTH  
*****  
*PROPER TERMINATION OF TEST IS NO CHANGE IN CONTENTS OF  
*R0&R1, AND ALL CONDITION CODES CLEAR EXCEPT Z=1.  
*****  
†ST35: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
0 ;SOURCE LENGTH  
NXM ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
1 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 204  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
CLZ ;CLEAR CONDITION CODE Z  
+SEV!SEN!SEC ;SET ALL OTHER CONDITION CODES  
;EXECUTE "LOCC"  
LOCC  
;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 35 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
54$:  
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 35 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
65$:  
TSTB R0 ;CHECK R0  
BEQ 1$  
ERROR 3 ;*****TEST 35 - ERROR 3*****  
;R0 CHANGED  
1$:  
CMP SRCAD,R1 ;CHECK R1  
BEQ 40$  
ERROR 4 ;*****TEST 35 - ERROR 4*****  
;R1 CHANGED  
40$:
```

3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355

011506 000004
011510 105777 167024
011514 100536
011516 026767 167026 167134
011524 001007
011526 032767 000001 167052
011534 001403
011536 005767 167032
011542 001123
011544
011544 004567 003442
011550 000377
011552 016434
011554 177777
011556 177777
011560 000200
011562 012767 011646 167100
011570 012777 015356 167066
011576 005077 167064
011602 004767 003524
011606 013777 000554 167046
011614 004567 003466
011620 000000
011622 106427 000000
011626 052777 000100 167024
011634 004767 003306
011640 010637 000676
011644 000277
011646 076040
011650 106767 167016
011654 032777 000100 166776
011662 001364
011664 042767 177400 167000
011672 023767 000674 166772
011700 001401
011702 104001
011704
011704 023706 000676
011710 001403
011712 010637 000700

```
*****  
;TEST 36 TEST INTERRUPTABILITY OF "LOCC" INSTRUCTION  
*****  
;THIS TEST INTERRUPTS THE EXECUTION OF "LOCC" INSTRUCTION,  
;RESUMES EXECUTION AFTER THE INTERRUPT, AND CHECKS THE  
;RESULTS. RESULTS INDICATE THE SEARCH CHARACTER FOUND.  
*****  
;*****  
;*****  
TST36: SCOPE  
TSTB @SWR ;TEST BIT 7 OF SWR  
BMI TST37 ;SKIP TO NEXT TEST IF SET  
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?  
BNE T36CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST  
BIT @BIT0, $ENV ;IF YES, CHECK IF ON APT  
BEQ T36CONT ;BT IF NOT UNDER APT, AND DO THIS TEST  
TST $PASS ;CHECK IF ON FIRST PASS  
BNE TST37 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST  
  
T36CONT: JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS  
377 ;SOURCE LENGTH  
EJF1 ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
200 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER  
MOV @LOC, PCI ;STORE PC OF TEST INSTRUCTION  
MOV @INTR, @TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE  
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT  
JSR PC, TDCNE  
MOV @NULL, @TBUF ;SEND CARRIAGE RETURN  
JSP R5, XPSW ;STORE EXPECTED PSW VALUE  
WORD 00  
MTPS #0 ;SET PSW TO ALLOW INTERRUPTS  
BIS #100, @TCSR ;ENABLE TTY INTERRUPTS  
RELOC: JSR PC, GENR ;SET UP GENERAL REGISTERS  
MOV SP, @SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION  
SCC ;SET ALL CONDITION CODES  
  
LOC: LOCC ;EXECUTE "LOCC"  
  
MFFS CCODES ;STORE THE PSW  
BIT #100, @TCSR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE  
BNE RELOC ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED  
BIC #177400, CCODES ;CLEAR ALL UNUSED BITS  
CMP @EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 36 - ERROR 1*****  
PSW ERROR  
EXPECTED PSW IS STORED AT "EXPPSW"  
ACTUAL PSW IS STORED AT "CCODES"  
  
64$: CMP @SAVR6, SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP, @BADR6 ;STORE BAD SP VALUE
```

```

3356 011716 104002          ERROR 2          ;*****TEST 36 - ERROR 2*****
3357                               ;STACK POINTER NOT RESTORED BY INSTRUCTION
3358                               ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3359                               ;ERRONEOUS VALUE IS AT "BADR6"
3360 011720                55$:
3361 011720 026704 166730    CMP      FILL,R4          ;CHECK R4 UNCHANGED
3362 011724 001401          BEQ      66$            ;BR IF OK
3363 011726 104003          ERROR 3          ;*****TEST 36 - ERROR 3*****
3364                               ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3365 011730 026705 166722    66$:    CMP      TABLE,R5       ;CHECK R5 UNCHANGED
3366 011734 001401          BEQ      67$            ;BR IF OK
3367 011736 104004          ERROR 4          ;*****TEST 36 - ERROR 4*****
3368                               ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3369 011740                67$:
3370 011740 016703 166702    MOV      SRCAD,R3        ;LOCATE "CHARACTER"
3371 011744 016702 166674    MOV      SRCLN,R2
3372 011750 121367 166700    68$:    CMPB     (R3),FILL
3373 011754 001403          BEQ      69$
3374 011756 105723          TSTB    (R3)+           ;POINT TO NEXT BYTE
3375 011760 005302          DEC     R2              ;DECREMENT BYTE COUNT
3376 011762 001372          BNE     68$            ;BR IF NOT FINISHED
3377 011764 020200          69$:    CMP     R2,R0           ;CHECK R0=REMAINING SOURCE LENGTH
3378 011766 001401          BEQ     70$
3379 011770 104005          ERROR 5          ;*****TEST 36 - ERROR 5*****
3380                               ;LOCATE LENGTH ERROR
3381 011772 020301          70$:    CMP     R3,R1           ;CHECK R1=ADDRESS OF CHARACTER
3382 011774 001401          BEQ     71$
3383 011776 104006          ERROR 6          ;*****TEST 36 - ERROR 6*****
3384                               ;LOCATE ADDRESS ERROR
3385 012000                71$:
3386
3387 012000 106427 000200    40$:    MTPS   #2CO           ;RESTORE PSW TO PRIORITY 7
3388 012004 016777 166656 166652    MOV     TPSW,STVECT    ;RESTORE TRAP CATCHER
    
```



```

3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402 012012 000004
3403 012014 004567 003172
3404 012020 000007
3405 012022 017034
3406 012024 177777
3407 012026 177777
3408 012030 000001
3409 012032 004567 003250
3410 012036 000200
3411 012040 004767 003102
3412 012044 000277
3413
3414 012046 076041
3415
3416 012050 106767 166616
3417 012054 042767 177400 166610
3418 012062 023767 000674 166602
3419 012070 001401
3420 012072 104001
3421
3422
3423
3424 012074
3425 012074 023706 000676
3426 012100 001403
3427 012102 010637 000700
3428 012106 104002
3429
3430
3431
3432 012110
3433 012110 026704 166540
3434 012114 001401
3435 012116 104003
3436
3437 012120 026705 166532
3438 012124 001401
3439 012126 104004
3440
3441 012130
3442 012130 016703 166512
3443 012134 016702 166504
3444 012140 121367 166510

```

```

*****
*TEST 37 TEST "SKPC" INSTRUCTION WITH NON-SKIP CHARACTERS IN SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
*ADDRESS OF THE FIRST OCCURRENCE OF A CHARACTER
*OTHER THAN THE SEARCH CHARACTER, R0 EQUALS THE
*REMAINING LENGTH OF THE SOURCE STRING, AND ALL
*CONDITION CODES CLEAR
*****
†ST37: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      7              ;SOURCE LENGTH
      BUF2         ;SOURCE ADDRESS
      NXM          ;STORE NON-ZERO VALUES TO TEST R2 & R3
      NXM          ; UNAFFECTED BY DIS STRING INSTRUCTIONS
      1            ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      JSR      R5,XPSW     ;STORE EXPECTED PSW VALUE
      .WORD     200
      JSR      PC,GENR     ;SET UP GENERAL REGISTERS
      SCC              ;SET ALL CONDITION CODES
                      ;EXECUTE "SKPC"
      SKPC
                      ;CHECK RESULTS
                      ;STORE THE PSW
      MFPS      CCODES    ;CLEAR ALL UNUSED BITS
      BIC      #177400,CCODES
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$       ;BR, IF EQUAL
                      ;*****TEST 37 - ERROR 1*****
      ERROR    1        ;PSW ERROR
                      ;EXPECTED PSW IS STORED AT "EXPPSW"
                      ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$       ;BR IF OK
      MOV      SP,@#BADR6 ;STORE BAD SP VALUE
      ERROR    2        ;*****TEST 37 - ERROR 2*****
                      ;STACK POINTER NOT RESTORED BY INSTRUCTION
                      ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                      ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      CMP      FILL,R4    ;CHECK R4 UNCHANGED
      BEQ      66$       ;BR IF JK
      ERROR    3        ;*****TEST 37 - ERROR 3*****
                      ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      66$:
      CMP      TABLE,R5 ;CHECK R5 UNCHANGED
      BEQ      67$       ;BR IF OK
      ERROR    4        ;*****TEST 37 - ERROR 4*****
                      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV      SRCAD,R3   ;SKIP "CHARACTER"
      MOV      SRCLN,R2
      68$:
      CMPB     (R3),FILL

```

3445	012144	001003	BNE	69\$	
3446	012146	105723	TSTB	(R3)+	:POINT TO NEXT BYTE
3447	012150	005302	DEC	R2	:DECREMENT BYTE COUNT
3448	012152	001372	BNE	68\$:BR, IF NOT FINISHED
3449	012154	020200	69\$: CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3450	012156	001401	BEQ	70\$	
3451	012160	104005	ERROR	5	:*****TEST 37 - ERROR 5*****
3452					:SKIP LENGTH ERROR
3453	012162	020301	70\$: CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3454	012164	001401	BEQ	71\$	
3455	012166	104006	ERROR	6	:*****TEST 37 - ERROR 6*****
3456					:SKIP ADDRESS ERROR
3457	012170		71\$:		

3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513

012170 000004
012172 004567 003014
012176 000010
012200 017034
012202 177777
012204 177777
012206 000377
012210 016700 166432
012214 016701 166424
012220 116720 166430
012224 005301
012226 001374
012230 004567 003052
012234 000204
012236 004767 002704
012242 000244
012244 000273
012246 076041
012250 106767 166416
012254 042767 177400 166410
012252 023767 000674 166402
012270 001401
012272 104001
012274
012274 023706 000676
012300 001403
012302 010637 000700
012306 104002
012310
012310 026704 166340
012314 001401
012316 104003
012320 026705 166332
012324 001401
012326 104004

```
*****  
*TEST 40 TEST "SKPC" WITH ALL SKIP CHARACTERS IN THE SOURCE  
*****  
*PROPER TERMINATION OF THIS TEST IS RO EQUALS ZERO  
*R1 EQUALS THE ADDRESS+1 OF THE LSB IN THE SOURCE STRING,  
*AND ALL CONDITION CODES ARE CLEAR EXCEPT Z=1.  
*****  
TEST40: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
10 ;SOURCE LENGTH  
BUF2 ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
377 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER  
MOV SRCAD,R0 ;GENERATE SOURCE STRING  
MOV SRCLN,R1 ;OF ALL SKIP CHARACTERS  
1$: MOVB FILL,(R0)+  
DEC R1  
BNE 1$  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 204  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
CLZ ;CLEAR CONDITION CODE "Z"  
+SEV!SEN!SEC ;SET ALL OTHER CONDITION CODES  
;EXECUTE "SKPC"  
SKPC  
;CHECK RESULTS  
MFPS ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEG 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 40 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
64$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 40 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
55$: CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 40 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 40 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
```

3514	012330		67\$:			
3515	012330	016703		MOV	SRCAD,R3	:SKIP "CHARACTER"
3516	012334	016702		MOV	SRCLN,R2	
3517	012340	121367	68\$:	CMPB	(R3),FILL	
3518	012344	001003		BNE	69\$	
3519	012346	105723		TSTB	(R3)+	:POINT TO NEXT BYTE
3520	012350	005302		DEC	R2	:DECREMENT BYTE COUNT
3521	012352	001372		BNE	68\$:BR IF NOT FINISHED
3522	012354	020200	69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3523	012356	001401		BEQ	70\$	
3524	012360	104005		ERROR	5	:*****TEST 40 - ERROR 5*****
3525						:SKIP LENGTH ERROR
3526	012362	020301	70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3527	012364	001401		BEQ	71\$	
3528	012366	104005		ERROR	6	:*****TEST 40 - ERROR 5*****
3529						:SKIP ADDRESS ERROR
3530	012370		71\$:			

35330
35331
35332
35333
35334
35335
35336
35337
35338
35339
35340
35341
35342
35343
35344
35345
35346
35347
35348
35349
35350
35351
35352
35353
35354
35355
35356
35357
35358
35359
35360
35361
35362
35363
35364
35365
35366
35367
35368
35369
35370
35371
35372
35373
35374
35375
35376
35377
35378
35379
35380

012370 000004
012372 004567 002614
012376 000000
012400 177777
012402 177777
012404 177777
012406 000000
012410 004567 002672
012414 000204
012416 004767 003324
012422 000244
012424 000267

012426 076041

012430 106767 166236
012434 042767 177400 166230
012442 023767 000674 166222
012450 001401
012452 104001

012454
012454 023706 000676
012460 001403
012462 010537 000700
012466 104002

012470
012470 105700
012472 001401
012474 104003

012476 026701 166144
012502 001401
012504 104004

012506

```
*****  
*TEST 41 TEST "SKPC" WITH ZERO SOURCE LENGTH  
*****  
*PROPER TERMINATION OF TEST IS NO CHANGE IN CONTENTS OF  
*R0&R1, AND ALL CONDITION CODES CLEAR EXCEPT Z=1.  
*****  
*ST41: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
0 ;SOURCE LENGTH  
NXM ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
0 ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
JSR R5,XPSW ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER  
.WORD 204 ;STORE EXPECTED PSW VALUE  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
CLZ ;CLEAR "Z" CONDITION CODE  
+SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES  
;EXECUTE "SKPC"  
  
SKPC ;CHECK RESULTS  
MFP5 CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSED BITS  
CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 41 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
  
64$:  
CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 41 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
  
65$:  
TSTB R0 ;CHECK R0  
BEQ 1$  
ERROR 3 ;*****TEST 41 - ERROR 3*****  
;R0 CHANGED  
  
1$:  
CMP SRCAD,R1 ;CHECK R1  
BEQ 40$  
ERROR 4 ;*****TEST 41 - ERROR 4*****  
;R1 CHANGED  
  
40$:
```


000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200

```

*****
*TEST 43 TEST "SCANC" WITH A MEMBER CHARACTER IN SOURCE & SOURCE LENGTH AT CHARACTER
*****
*PROPER TERMINATION OF TEST IS R1 EQUAL ADDRESS OF SOURCE
*BYTE WHICH WHEN ANDED WITH MASK=1 R0 EQUALS REMAINDER
*OF SOURCE LENGTH, AND ALL CONDITION CODES CLEAR.
*****
*543: SCOPE
JSR RS,PREP ;SET UP INSTRUCTION ARGUMENTS
3 ;SOURCE LENGTH
BLF1 ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
2 ;SEARCH IMASK FOR SCANC/SPANC CHARACTER
MOV #BUF1, TABLE
JSR RS, XPSW ;STORE EXPECTED PSW VALUE
WORD R00
JSR PC, GENR ;SET UP GENERAL REGISTERS
MOV TABLE, R5
SCC ;SET ALL CONDITION CODES
;EXECUTE "SCANC"
C13104 C76042 SCANC ;CHECK RESULTS
C13106 106767 165560 MFPS C00DES ;STORE THE PSW
C13112 042767 177400 165552 BIC #177400, C00DES ;CLEAR ALL UNUSED BITS
C13120 023767 000674 165544 CMP #EXPPSW, C00DES ;CHECK PSW AGAINST EXPECTED VALUE
C13126 001401 BEQ 64$ ;BR, IF EQUAL
C13130 104001 ERROR 1 ;*****TEST 43 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "C00DES"
C13132 C76042 64$: CMP #SAVRE, SP ;VERIFY STACK POINTER IS RESTORED
C13136 001401 BEQ 65$ ;BR IF OK
C13140 010637 000700 MOV SP, #BADR6 ;STORE BAD SP VALUE
C13144 104002 ERROR 2 ;*****TEST 43 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVRE"
;ERRONEOUS VALUE IS AT "BADR6"
C13146 C76042 65$: CMP FILL, R4 ;CHECK R4 UNCHANGED
C13152 001401 BEQ 66$ ;BR IF OK
C13154 104003 ERROR 3 ;*****TEST 43 - ERROR 3*****
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
C13156 C76042 66$: CMP TABLE, R5 ;CHECK R5 UNCHANGED
C13162 001401 BEQ 67$ ;BR IF OK
C13164 104004 ERROR 4 ;*****TEST 43 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
C13166 C76042 67$: MOV SRCAD, R3 ;POINT R3 TO SOURCE STRING
C13172 016702 165446 MOV SRCLEN, R2 ;STORE SOURCE LENGTH IN R2
C13176 111204 68$: MOVSB (R3), R4 ;USE SOURCE BYTE AS INDEX

```


3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806

013236 000004
013240 004567 001746
013244 000200
013246 016434
013250 177777
013252 177777
013254 000000
013256 012767 016434 165372
013264 004567 002016
013270 000204
013272 004767 001650
013276 016705 165354
013302 000277
013304 000244
013306 076042
013310 106767 165356
013314 042767 177400 165350
013322 023767 000674 165342
013330 001401
013332 104001
013334
013334 023706 000676
013340 001403
013342 010637 000700
013346 104002
013350
013350 026704 165300
013354 001401
013356 104003
013360 026705 165272
013364 001401
013366 104004
013370
013370 016703 165252
013374 016702 165244

```
*****  
*TEST 44 TEST "SCANC" FAILS, ALL NON-MEMBER CHARACTERS IN STRING  
*****  
*THIS TEST VERIFIES THAT "SCANC" INDICATES NO MEMBERS FOUND  
*RESULT IS R0=0, R1 EQUALS THE ADDRESS+1 OF LSB IN SOURCE,  
*AND ALL CONDITION CODES CLEAR EXCEPT Z=1.  
*****  
†ST44: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
200 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
0 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER  
MOV #BUF1,TABLE  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 204  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
MOV TABLE,R5  
SCC ;SET ALL CONDITION CODES, EXCEPT  
CLZ ;CLEAR "Z" CONDITION CODE  
SCANC ;EXECUTE "SCANC"  
  
;CHECK RESULTS  
MFPS CCODES ;STORE THE PSW  
BIC #177400,CCODES ;CLEAR ALL UNUSEC BITS  
CMP J#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 44 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "EXPPSW"  
;ACTUAL PSW IS STORED AT "CCODES"  
  
64$:  
CMP J#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,J#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 44 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"  
;ERRONEOUS VALUE IS AT "BADR6"  
  
65$:  
CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 44 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"  
;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 44 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"  
  
67$:  
MOV SRCAD,R3 ;POINT R3 TO SOURCE STRING  
MOV SRCLN,R2 ;STORE SOURCE LENGTH IN R2
```

3807	013400	111304		69\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
3808	013402	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
3809	013406	136467	016434 165240		BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
3810	013414	001003			BNE	69\$	
3811	013416	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
3812	013420	005302			DEC	R2	:DECREMENT BYTE COUNT
3813	013422	001366			BNE	68\$:BR. IF NOT FINISHED
3814	013424	020200		69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3815	013426	001401			BEQ	70\$	
3816	013430	104005			ERROR	5	:*****TEST 44 - ERROR 5*****
3817							:SCAN LENGTH ERROR
3818	013432	020301		70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3819	013434	001401			BEQ	71\$	
3820	013436	104006			ERROR	6	:*****TEST 44 - ERROR 6*****
3821							:SCAN ADDRESS ERROR
3822	013440			71\$:			

3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878

013440 000004
013442 105777 165072
013446 100550
013450 026767 165074 165202
013456 001007
013460 032767 000001 165120
013466 001403
013470 005767 165100
013474 001135
013476
013476 004567 001510
013502 000400
013504 016434
013506 177777
013510 177777
013512 000000
013514 012767 016434 165134
013522 012767 013614 165140
013530 012777 015356 165126
013536 005077 165124
013542 004767 001564
013546 013777 000554 165106
013554 004567 001526
013560 000004
013562 106427 000000
013566 052777 000100 165064
013574 004767 001346
013600 010637 000676
013604 016705 165046
013610 000244
013612 000267
013614 076042
013616 106767 165050
013622 032777 000100 165030
013630 001361
013632 042767 177400 165032
013640 023767 000674 165024
013646 001401
013650 104001

```
*****  
;TEST 45 TEST INTERRUPTABILITY OF "SCANC"  
*****  
;THIS TEST INTERRUPTS THE "SCANC" INSTRUCTION, RESUMES  
;EXECUTION AFTER THE INTERRUPT, AND CHECKS RESULTS.  
;RESULTS ARE R0=0, R1 EQUALS ADDRESS+1 OF LSB IN SOURCE,  
;AND ALL CONDITION CODES CLEAR EXCEPT Z=1.  
*****  
;*****  
;*****  
TST45: SCOPE  
TSTB @SWR ;TEST BIT 7 OF SWR  
BMI TST46 ;SKIP TO NEXT TEST IF SET  
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?  
BNE T45CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST  
BIT @BIT0, $ENV ;IF YES, CHECK IF ON APT  
BEQ T45CONT ;BR IF NOT UNDER APT, AND DO THIS TEST  
TST $PASS ;CHECK IF ON FIRST PASS  
BNE TST46 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST  
  
T45CONT: JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS  
400 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3  
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS  
0 ;SEARCH (MASK FOR SCANC/SPANC) CHARACTER  
MOV @BUF1, TABLE  
MOV @SCN, PCI ;STORE PC OF TEST INSTRUCTION  
MOV @INTR, @TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE  
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT  
JSR PC, TDONE  
MOV @NULL, @TBUF ;SEND CARRIAGE RETURN  
JSR R5, XPSW ;STORE EXPECTED PSW VALUE  
WORD 04  
MTPS #0 ;SET PSW TO ALLOW INTERRUPTS  
BIS @100, @TCSR ;ENABLE TTY INTERRUPTS  
REPSCN: JSR PC, GENR ;SET UP GENERAL REGISTERS  
MOV SP, @SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION  
MOV TABLE, R5  
CLZ ;CLEAR "Z" CONDITION CODE  
+SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES  
  
SCN: SCANC ;EXECUTE "SCANC"  
  
MFPS CCODES ;CHECK RESULTS  
BIT @100, @TCSR ;STORE THE PSW  
BNE REPSCN ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE  
BIC @177400, CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED  
CMP @EXPPSW, CCODES ;CLEAR ALL UNUSED BITS  
BEQ 645 ;CHECK PSW AGAINST EXPECTED VALUE  
ERROR 1 ;BR, IF EQUAL  
*****TEST 45 - ERROR 1*****  
PSW ERROR  
EXPECTED PSW IS STORED AT "EXPPSW"
```

```

3879                                     ;ACTUAL PSW IS STORED AT "CCODES"
3880 013652                               64$:
3881 013652 023706 000676                CMP    2#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
3882 013656 001403                        BEQ    65$            ;BR IF OK
3883 013660 010637 000700                MOV    SP,2#BADR6    ;STORE BAD SP VALUE
3884 013664 104002                        ERROR  2             ;*****TEST 45 - ERROR 2*****
3885                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
3886                                     ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3887                                     ;ERRONEOUS VALUE IS AT "BADR6"
3888 013666                               65$:
3889 013666 026704 164762                CMP    FILL,R4       ;CHECK R4 UNCHANGED
3890 013672 001401                        BEQ    66$            ;BR IF OK
3891 013674 104003                        ERROR  3             ;*****TEST 45 - ERROR 3*****
3892                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3893 013676 026705 164754                66$:  CMP    TABLE,R5    ;CHECK R5 UNCHANGED
3894 013702 001401                        BEQ    67$            ;BR IF OK
3895 013704 104004                        ERROR  4             ;*****TEST 45 - ERROR 4*****
3896                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3897 013706                               67$:
3898 013706 016703 164734                MOV    SRCAD,R3      ;POINT R3 TO SOURCE STRING
3899 013712 016702 164726                MOV    SRCLN,R2     ;STORE SOURCE LENGTH IN R2
3900 013716 111304                               68$:  MOVSB  (R3),R4       ;USE SOURCE BYTE AS INDEX
3901 013720 042704 177400                BIC    #177400,R4   ;CLEAR ANY BYTE SIGN EXTENSION
3902 013724 136467 164722                BITB  BUF1(R4),FILL ;"AND" TABLE ENTRY WITH MASK
3903 013732 001003                        BNE    69$
3904 013734 105723                        TSTB  (R3)+         ;POINT TO NEXT SOURCE BYTE
3905 013736 005302                        DEC    R2           ;DECREMENT BYTE COUNT
3906 013740 001366                        BNE    68$          ;BR IF NOT FINISHED
3907 013742 020200                               69$:  CMP    R2,R0        ;CHECK R0=REMAINING SOURCE LENGTH
3908 013744 001401                        BEQ    70$
3909 013746 104005                        ERROR  5             ;*****TEST 45 - ERROR 5*****
3910                                     ;SCAN LENGTH ERROR
3911 013750 020301                               70$:  CMP    R3,R1        ;CHECK R1=ADDRESS OF CHARACTER
3912 013752 001401                        BEQ    71$
3913 013754 104006                        ERROR  6             ;*****TEST 45 - ERROR 6*****
3914                                     ;SCAN ADDRESS ERROR
3915 013756                               71$:
3916
3917 013756 106427 000200                               70$:  MTPS  #200         ;RESTORE PSW TO PRIORITY 7
3918 013762 016777 164700 164674        MOV    TPSW,3TVECT  ;RESTORE TRAP CATCHER
    
```

```

3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931 013770 000004
3932 013772 004567 001214
3933 013776 000020
3934 014000 016434
3935 014002 177777
3936 014004 177777
3937 014006 000007
3938 014010 012767 016434 164640
3939 014016 004567 001264
3940 014022 000200
3941 014024 004767 001116
3942 014030 016705 164622
3943 014034 000244
3944 014036 000267
3945
3946 014040 076043
3947
3948 014042 106767 164624
3949 014046 042767 177400 164616
3950 014054 023767 000674 164610
3951 014062 001401
3952 014064 104001
3953
3954
3955
3956 014066
3957 014066 023706 000676
3958 014072 001403
3959 014074 010637 000700
3960 014100 104002
3961
3962
3963
3964 014102
3965 014102 026704 164846
3966 014106 001401
3967 014110 104003
3968
3969 014112 026705 164540
3970 014116 001401
3971 014120 104004
3972
3973 014122
3974 014122 016703 164520

```

```

*****
*TEST 46 TEST "SCANC" WITH NON-MEMBER CHARACTER IN SOURCE, SL .GT. 0 AT CHARACTER
*****
*THIS TEST VERIFIES THAT "SCANC" INDICATES A NON-MEMBER
*FOUND.
*THE RESULT IS R1 EQUALS ADDRESS OF NON-MEMBER CHARACTER, R0
*EQUALS THE REMAINING SOURCE LENGTH, AND ALL
*CONDITION CODES CLEAR.
*****
*ST46: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
20 ;SOURCE LENGTH
BUF1 ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
7 ;SEARCH [MASK FOR SCANC/SCANC] CHARACTER
MOV #BUF1, TABLE
JSR R5, XPSW ;STORE EXPECTED PSW VALUE
WORD 200
JSR PC, GENR ;SET UP GENERAL REGISTERS
MOV TABLE, R5
CLZ ;CLEAR "Z" CONDITION CODE
+SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES
;EXECUTE "SCANC"
SCANC
;CHECK RESULTS
MFPS CCODES ;STORE THE PSW
BIC #177400, CCODES ;CLEAR ALL UNUSED BITS
CMP @#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 46 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "CCODES"
64$:
CMP @#SAVR6, SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65$ ;BR IF OK
MOV SP, @#BADR6 ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 46 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
;ERRONEOUS VALUE IS AT "BADR6"
65$:
CMP FILL, R4 ;CHECK R4 UNCHANGED
BEQ 66$ ;BR IF OK
ERROR 3 ;*****TEST 46 - ERROR 3*****
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
66$:
CMP TABLE, R5 ;CHECK R5 UNCHANGED
BEQ 67$ ;BR IF OK
ERROR 4 ;*****TEST 46 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:
MOV SRCAD, R3 ;PCINT R3 TO SOURCE STRING

```

3975	014126	016702	164512		MOV	SACLN,R2	:STORE SOURCE LENGTH IN R2
3976	014132	111304		69\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
3977	014134	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
3978	014140	136467	216434	164506	BITB	BJF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
3979	014146	001403			BEQ	69\$	
3980	014150	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
3981	014152	005302			DEC	R2	:DECREMENT BYTE COUNT
3982	014154	001366			BNE	68\$:BR, IF NOT FINISHED
3983	014156	020200		69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3984	014160	001401			BEQ	70\$	
3985	014162	104005			ERROR	5	:*****TEST 46 - ERROR 5*****
3986							:SPAN LENGTH ERROR
3987	014164	020301		70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3988	014166	001401			BEQ	71\$	
3989	014170	104006			ERROR	6	:*****TEST 46 - ERROR 6*****
3990							:SPAN ADDRESS ERROR
3991	014172			71\$:			

```

3999
3999
3999
3999
3999
3999
3999
3999
3999
3999
4000
4001
4002
4003 014172 000004
4004 014174 004567 001012
4005 014200 000100
4006 014202 016434
4007 014204 177777
4008 014206 177777
4009 014210 000377
4010 014212 012767 016434 164436
4011 014220 004567 001062
4012 014224 000204
4013 014226 004767 000714
4014 014232 016705 164420
4015 014236 000244
4016 014240 000267
4017
4018 014242 076043
4019
4020 014244 106767 164422
4021 014250 042767 177400 164414
4022 014256 023767 000674 164406
4023 014254 001401
4024 01426E 104001
4025
4026
4027
4028 014270
4029 014270 023706 000676
4030 014274 001403
4031 014276 010637 000700
4032 014302 104002
4033
4034
4035
4036 014304
4037 014304 026704 164344
4038 014310 001401
4039 014312 104003
4040
4041 014314 026705 164336
4042 014320 001401
4043 014322 104004
4044
4045 014324
4046 014324 016703 164316
4047 014330 016702 164310

```

```

*****
*TEST 47 TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS R0=0, R1 EQUALS
*ADDRESS+1 OF LSB OF SOURCE, AND ALL CONDITION CODES
*CLEAR EXCEPT Z=1
*****
+ST47: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
;SOURCE LENGTH
;SOURCE ADDRESS
;STORE NON-ZERO VALUES TO TEST R2 & R3
; UNAFFECTED BY DIS STRING INSTRUCTIONS
;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
MOV #BUF1,TABLE
JSR R5,XPSW ;STORE EXPECTED PSW VALUE
;WORD 204
JSR PC,GENR ;SET UP GENERAL REGISTERS
MOV TABLE,R5
C-LZ ;CLEAR "Z" CONDITION CODE
+SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES
;EXECUTE "SPANC"
SPANC
;CHECK RESULTS
;STORE THE PSW
;CLEAR ALL UNUSED BITS
;CHECK PSW AGAINST EXPECTED VALUE
;BR, IF EQUAL
*****TEST 47 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "CCODES"
64$:
CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65$ ;BR IF OK
MOV SP,@BADR6 ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 47 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
;ERRONEOUS VALUE IS AT "BADR6"
65$:
CMP FILL,R4 ;CHECK R4 UNCHANGED
BEQ 66$ ;BR IF OK
ERROR 3 ;*****TEST 47 - ERROR 3*****
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
;CHECK R5 UNCHANGED
;BR IF OK
66$:
CMP TABLE,R5 ;CHECK R5 UNCHANGED
BEQ 67$ ;BR IF OK
ERROR 4 ;*****TEST 47 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:
MOV SRCAD,R3 ;POINT R3 TO SOURCE STRING
MOV SRCLN,R2 ;STORE SOURCE LENGTH IN R2

```


F08

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 96
 SYKRA.P11 22-DEC-76 10:47 T47

TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE

SEC 0097

4048	014334	111304	69\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
4049	014336	042704	177400	BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
4050	014342	136467	016434 164304	BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
4051	014350	001403		SEQ	69\$	
4052	014352	105723		TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
4053	014354	005302		DEC	R2	:DECREMENT BYTE COUNT
4054	014356	001366		BNE	68\$:BR IF NOT FINISHED
4055	014360	020200	69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
4056	014362	001401		BEG	70\$	
4057	014364	104005		ERROR	5	:*****TEST 47 - ERROR 5*****
4058						:SPAN LENGTH ERROR
4059	014366	020301	70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
4060	014370	001401		BEG	71\$	
4061	014372	104006		ERROR	6	:*****TEST 47 - ERROR 6*****
4062						:SPAN ADDRESS ERROR
4063	014374		71\$:			

4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

:*****
:TEST 50      TEST INTERRUPTABILITY OF "SPANC"
:*****
:THIS TEST INTERRUPTS THE "SPANC" INSTRUCTION, RESUMES
:EXECUTION AFTER THE INTERRUPT, AND CHECKS RESULTS.
:RESULTS ARE R1 EQUALS ADDRESS OF FIRST BYTE WHICH
:WHEN ANDED WITH MASK=0, R0 EQUALS REMAINING SOURCE
:LENGTH, AND ALL CONDITION CODES CLEAR
:*****
:*****
TST50:  SCOPE
        STB      $SWR      ;TEST BIT 7 OF SWR
        BMT      $EOP      ;SKIP TO NEXT TEST IF SET
        CMP      $TPS,$CSR  ;IS SLL USED FOR INTERRUPTS THE CONSOLE?
        BNE      $PCCONT    ;BR. IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT      $BITC,$ENV ;IF YES, CHECK IF ON APT
        BEQ      $PCCONT    ;BR IF NOT UNDER APT, AND DO THIS TEST
        TST      $PASS     ;CHECK IF ON FIRST PASS
        BNE      $EOP      ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

SPCCONT: JSR      RS,PREP   ;SET UP INSTRUCTION ARGUMENTS
        LCC      ;SOURCE LENGTH
        BUF1    ;SOURCE ADDRESS
        NXM     ;DESTINATION LENGTH
        NXM     ;DESTINATION ADDRESS
        L77    ;FILL CHARACTER

        MOV      $BUF1, TABLE
        MOV      $SPN, PCI   ;STORE PC OF TEST INSTRUCTION
        MOV      $INTR, $TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
        CLR      $TPSW      ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
        JSR      PC,TDONE
        MOV      $R0,$NULL, $TBUF ;SEND CARRIAGE RETURN
        JSR      RS,$XPSW    ;STORE EXPECTED PSW VALUE
        .WORD   00
        MTPS    $0          ;SET PSW TO ALLOW INTERRUPTS
        BIS      $100, $TCSR ;ENABLE TTY INTERRUPTS
REPSPN: JSR      PC,GENR    ;SET UP GENERAL REGISTERS
        MOV      SP, $SAVR6  ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
        MOV      TABLE, R5
        SCC      ;SET ALL CONDITION CODES

SPN:    SPANC              ;EXECUTE "SPANC"

        MFPS    CCODES     ;CHECK RESULTS
        BIT      $100, $TCSR ;STORE THE PSW
        BNE      REPSPN    ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BIC      $177400, CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
        CMP      $EXPPSW, CCODES ;CLEAR ALL UNUSED BITS
        BEQ      $4$       ;CHECK PSW AGAINST EXPECTED VALUE
        ERROR   !          ;BR, IF EQUAL
:*****TEST 50 - ERROR *****
:PSW ERROR
:EXPECTED PSW IS STORED AT "EXPPSW"

```

```

4: 014604          :ACTUAL PSW IS STORED AT "CODES"
4: 014604 023706 000676 64$: CMP 2#SAVR6, R4 :VERIFY STACK POINTER IS RESTORED
4: 014610 001403 :BR IF OK
4: 014612 010637 000700 :STORE BAD SP VALUE
4: 014616 104002 :*****TEST 50 - ERROR 2*****
4: :STACK POINTER NOT RESTORED BY INSTRUCTION
4: :EXPECTED VALUE OF SP IS STORED AT "SAVR6"
4: :ERRONEOUS VALUE IS AT "BADR6"
4: 014620          65$: CMP FILL, R4 :CHECK R4 UNCHANGED
4: 014620 026704 164030 :BR IF OK
4: 014624 001401 :*****TEST 50 - ERROR 3 *****
4: 014626 104003 :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
4: :CHECK R5 UNCHANGED
4: 014630 026705 164022 66$: CMP TABLE, R5 :BR IF OK
4: 014634 001401 :*****TEST 50 - ERROR 4*****
4: 014636 104004 :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
4: 014640          67$: MOV SRCAD, R3 :POINT R3 TO SOURCE STRING
4: 014644 016703 164002 :MOV SRCLEN, R2 :STORE SOURCE LENGTH IN R2
4: 014650 111304 68$: MOVB (R3), R4 :USE SOURCE BYTE AS INDEX
4: 014652 042704 177400 :BZC #177400, R4 :CLEAR ANY BYTE SIGN EXTENSION
4: 014656 136467 016434 163770 :BITB BUF1(R4), FILL :AND TABLE ENTRY WITH MASK
4: 014664 001403 :MOV R2, R3 :POINT TO NEXT SOURCE BYTE
4: 014666 105723 :DEC R2 :DECREMENT BYTE COUNT
4: 014670 005302 :BNE 68$ :BR IF NOT FINISHED
4: 014672 001366 :CMP R2, R0 :CHECK R0=REMAINING SOURCE LENGTH
4: 014674 020200 69$: BEQ 70$ :*****TEST 50 - ERROR 5*****
4: 014676 001401 :SPAN LENGTH ERROR
4: 014700 104005 :CHECK R1=ADDRESS OF CHARACTER
4: :*****TEST 50 - ERROR 6*****
4: :SPAN ADDRESS ERROR
4: 014702 020301 70$: CMP R2, R1
4: 014704 001401 :*****TEST 50 - ERROR 6*****
4: 014706 104006 :SPAN ADDRESS ERROR
4: 014710          71$:
4: 014712          40$: MTPS #200 :RESTORE PSW TO PRIORITY 7
4: 014714 016777 163746 163742 :MOV *PSW, 2*VECT :RESTORE TRAP CATCHER

```


J08

11:49 AM 11/10/76 10:49 PAGE 100
END OF PASS ROUTINE

380 0:00

11:49 AM 11/10/76

10:49 PAGE 100

.EVEN

1

```

4218
4219
4220
4221 015106 011637 000702 TZERO: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 0
4222 015112 104200 ;MOV (SP),3#OLDPC ;GET PC+2 WHERE UNEXPECTED TRAP OCCURRED
4223 ;ERROR 200 ;*****ERROR 200*****
4224 ;UNEXPECTED TRAP TO LOCATION 0
4225 015114 000000 ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4226 ;PROGRAM MUST BE RESTARTED AT THIS POINT
4227
4228 015116 011637 000702 TIMTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 4
4229 015122 104204 ;MOV (SP),3#OLDPC ;GET PC+2 WHERE UNEXPECTED TIMEOUT TRAP OCCURRED
4230 ;ERROR 204 ;*****ERROR 204*****
4231 ;UNEXPECTED TRAP TO LOCATION 4
4232 015124 000000 ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4233 ;PROGRAM MUST BE RESTARTED AT THIS POINT
4234
4235 015126 011637 000702 ILLTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 10
4236 015132 104210 ;MOV (SP),3#OLDPC ;GET PC+2 WHERE UNEXPECTED ILLEGAL INSTRUCTION TRAP OCCUR
4237 ;ERROR 210 ;*****ERROR 210*****
4238 ;UNEXPECTED TRAP TO LOCATION 10
4239 015134 000000 ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4240 ;PROGRAM MUST BE RESTARTED AT THIS POINT
4241
4242 015136 011637 000702 IOTTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 20
4243 015142 104220 ;MOV (SP),3#OLDPC ;GET PC+2 WHERE UNEXPECTED IOT INSTRUCTION TRAP OCCURRED
4244 ;ERROR 220 ;*****ERROR 220*****
4245 ;UNEXPECTED TRAP TO LOCATION 20
4246 015144 000000 ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4247 ;PROGRAM MUST BE RESTARTED AT THIS POINT
4248
4249
4250 015146 016700 163472 GENR: ;SUBROUTINE TO SETUP GENERAL REGISTERS FOR INSTRUCTION EXECUTION
4251 015152 016701 163470 ;MOV SRCLN,R0 ;STORE SOURCE LENGTH
4252 015156 016702 163468 ;MOV SRCAD,R1 ;STORE SOURCE ADDRESS
4253 015162 016703 163466 ;MOV DSTLN,R2 ;STORE DESTINATION LENGTH
4254 015166 016704 163464 ;MOV DSTAD,R3 ;STORE DESTINATION ADDRESS
4255 015172 016705 163462 ;MOV FILL,R4 ;STORE FILL CHARACTER
4256 015176 010637 000676 ;MOV TABLE,R5 ;
4257 015202 062737 000002 000676 ;MOV SP,3#SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4258 015210 000207 ;ADD #2,3#SAVR5 ;ADJUST SAVED SP BECAUSE OF JSR TO THIS ROUTINE
4259 ;RTS PC
4260
4261 015212 012567 163426 PREP: ;SUBROUTINE TO PREPARE INSTRUCTION PARAMETERS
4262 015216 012567 163424 ;MOV (R5)+,SRCLN ;SET SOURCE LENGTH
4263 015222 012567 163422 ;MOV (R5)+,SRCAD ;SET SOURCE ADDRESS
4264 015226 012567 163420 ;MOV (R5)+,DSTLN ;SET DESTINATION LENGTH
4265 015232 012567 163418 ;MOV (R5)+,DSTAD ;SET DESTINATION ADDRESS
4266 015236 000203 163416 ;MOV (R5)+,FILL ;SET FILL CHARACTER
4267 ;RTS R5 ;RETURN

```

```

4270 :SUBROUTINE TO CLEAR DESTINATION & BOUNDARY BYTES
4271 CLOST: MOV DSTAD,RO :POINT RO TO DESTINATION AREA
4272 015240 016700 163406 DEC RO :CLEAR LOWER BOUNDARY BYTE OF DEST
4273 015244 005300 CLAB (RO)+
4274 015246 105320 MOV DSTLN,R1 :STORE DEST. BYTE COUNT IN R1
4275 015250 016701 163374 CONTROL: CLAB (RO)+ :CLEAR DESTINATION
4276 015254 105020 DEC R1 :DECREMENT DEST. BYTE COUNT
4277 015256 005301 BNE CONTROL :BR IF NOT FINISHED
4278 015260 001375 CLAB (RO)+ :CLEAR UPPER BYTE BOUNDARY OF DEST
4279 015262 105020 RTS PC :RETURN
4280 015264 000207
4281 :SUBROUTINE TO GENERATE A SOURCE STRING
4282 GENSRC: MOV #BUF1,RO :POINT RO TO THE PATTERN STORED
4283 015272 012502 MOV (R5)+,R2 :STORE SOURCE BYTE COUNT IN R2
4284 015274 012501 MOV (R5)+,R1 :POINT R1 TO THE SOURCE ADDRESS
4285 015276 112021 CON: MOVB (RO)+,(R1)+ :TRANSFER PATTERN TO SOURCE
4286 015300 005302 DEC R2 :DECREMENT BYTE COUNT
4287 015302 001375 BNE GENCON :BR IF NOT FINISHED
4288 015304 000205 RTS R5 :RETURN
4289
4290 :SUBROUTINE TO RECORD EXPECTED PSW
4291 XPSW: MOV (R5)+,Q#EXPPSW :STORE EXPECTED PSW VALUE
4292 015312 106700 MFPS RO
4293 015314 032700 000020 BIT #TBIT,RO
4294 015320 001403 BEQ IS
4295 015322 052737 000020 000674 BIS #TBIT,Q#EXPPSW :OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4296 015330 000205 IS: RTS R5
4297
4298
    
```

```

4299
4300
4301
4302 015332 005077 163346 TDONE: CLR @TEMP ;SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4303 015336 105777 163316 IS: TSTB @TCSR ;CLEAR A TIMER
4304 015342 100404 BMI RETN ;IS SLU READY?
4305 015344 005277 163334 INC @TEMP ;BR IF READY
4306 015350 001372 BNE IS ;OTHERWISE INCREMENT TIMER
4307 015352 104300 ERROR 300 ;BR IF NOT TIMED OUT
4308 ;*****ERROR 300*****
4309 015354 000207 RETN: RTS PC ;NEVER GOT TRANSMIT DONE FLAG
4310 ;RETURN
4311 ;SUBROUTINE TO HANDLE TTY INTERRUPTS IN INSTRUCTION
4312 ;INTERRUPTABILITY TESTS
4313
4314 015356 INTR:
4315 015356 021667 163306 CMP (SP),PCI ;WAS PC AT INSTRUCTION UNDER TEST?
4316 015362 001003 BNE SEND ;BR, IF NO
4317 015364 032704 177400 CKR4: BIT #177400,R4 ;IF YES, CHECK UPPER BYTE OF R4
4318 015370 001004 BNE CLRINT ;IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4319 015372 013777 000554 163262 SEND: MOV @#NULL,@TBUF ;SEND ANOTHER CHARACTER
4320 015400 000002 RTI ;RETURN
4321 015402 042777 000100 163250 CLRINT: BIC #100,@TCSR ;IF NON-ZERO, CLEAR INTERRUPT ENABLE
4322 015410 000002 RTI ;CONTINUE INSTRUCTION
    
```


4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374

.SBTTL SCOPE HANDLER ROUTINE

```
*****  
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT  
*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)  
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>  
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
*SW14=1 LOOP ON TEST  
*SW09=1 LOOP ON ERROR  
*SW08=1 LOOP ON TEST IN SWR<5:0>  
*CALL  
* SCOPE ;;SCOPE=IOT
```

```
$SCOPE:  
1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?  
BNE $OVER ;;YES IF SW14=1  
*****START OF CODE FOR THE XOR TESTER*****  
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE  
;;THIS INSTRUCTION TO A "NOP" (NOP=240)  
MOV @#ERRVEC, -(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR  
MOV #55, @#ERRVEC ;;SET FOR TIMEOUT  
TST @#177060 ;;TIME OUT ON XOR?  
MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR  
BR $SVLAD ;;GO TO THE NEXT TEST  
5$: CMP (SP)+, (SP)+ ;;CLEAR THE STACK AFTER A TIME OUT  
MOV (SP)+, @#ERRVEC ;;RESTORE THE ERROR VECTOR  
BR 7$ ;;LOOP ON THE PRESENT TEST  
6$: *****END OF CODE FOR THE XOR TESTER*****  
BIT #BIT08, @SWR ;;LOOP ON SPEC. TEST?  
BEQ 2$ ;;BR IF NO  
MOV @SWR, -(SP) ;;SET DESIRED TEST NUM. FROM SWR  
BIC #55, @SWR (SP) ;;STRIP AWAY UNDESIED BITS  
CMPB (SP)+, $TSTNM ;;ON THE RIGHT TEST?  
BEQ $OVER ;;BR IF YES  
2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?  
BEQ $SVLAD ;;BR IF NO  
BIT #BIT09, @SWR ;;LOOP ON ERROR?  
BEQ 4$ ;;BR IF NO  
7$: MOV $LPERR, $LPADR ;;SET LOOP ADDRESS TO LAST SCOPE  
BR $OVER  
4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG  
$SVLAD: INCB $TSTNM ;;COUNT TEST NUMBERS  
MOVB $TSTNM, $'ESTN ;;SET TEST NUMBER IN APT MAILBOX  
MOV (SP), $LPADR ;;SAVE SCOPE LOOP ADDRESS  
MOV (SP), $LPERR ;;SAVE ERROR LOOP ADDRESS  
CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS  
MOVB #1, $ERMAX ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST  
$OVER: MOV $TSTNM, @DISPLAY ;;DISPLAY TEST NUMBER  
MOV $LPADR, (SP) ;;FUDGE RETURN ADDRESS  
RTI ;;FIXES PS
```

4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424

.SBTTL POWER DOWN AND UP ROUTINES

::*****

POWER DOWN ROUTINE

```

$PWRDN: MOV    $SILLUP, @#PWRVEC    ;; SET FOR FAST UP
        MOV    #340, @#PWRVEC+2    ;; PRIO:7
        MOV    RO, -(SP)            ;; PUSH RO ON STACK
        MOV    R1, -(SP)            ;; PUSH R1 ON STACK
        MOV    R2, -(SP)            ;; PUSH R2 ON STACK
        MOV    R3, -(SP)            ;; PUSH R3 ON STACK
        MOV    R4, -(SP)            ;; PUSH R4 ON STACK
        MOV    R5, -(SP)            ;; PUSH R5 ON STACK
        MOV    @SWR, -(SP)          ;; PUSH @SWR ON STACK
        MOV    SP, $SAVR6          ;; SAVE SP
        MOV    $PWRUP, @#PWRVEC    ;; SET UP VECTOR
        HALT
        BR     .-2                  ;; HANG UP

```

::*****

POWER UP ROUTINE

```

$PWRUP: MOV    $SILLUP, @#PWRVEC    ;; SET FOR FAST DOWN
        MOV    $SAVR6, SP          ;; GET SP
        CLR    $SAVR6              ;; WAIT LOOP FOR THE TTY
1$:      INC    $SAVR6              ;; WAIT FOR THE INC
        BNE    1$                  ;; OF WORD
        CLR    $TSTNM
        MOV    (SP)+, @SWR         ;; POP STACK INTO @SWR
        MOV    (SP)+, R5           ;; POP STACK INTO R5
        MOV    (SP)+, R4           ;; POP STACK INTO R4
        MOV    (SP)+, R3           ;; POP STACK INTO R3
        MOV    (SP)+, R2           ;; POP STACK INTO R2
        MOV    (SP)+, R1           ;; POP STACK INTO R1
        MOV    (SP)+, R0           ;; POP STACK INTO R0
        MOV    $PWRDN, @#PWRVEC    ;; SET UP THE POWER DOWN VECTOR
        MOV    #340, @#PWRVEC+2    ;; PRIO:7
        TYPE   $POWER              ;; REPORT THE POWER FAILURE
        SPWRMG: .WORD $POWER        ;; POWER FAIL MESSAGE POINTER
        MOV    (PC)+, (SP)         ;; RESTART AT $LOOP
        SPWRAD: .WORD $LOOP         ;; RESTART ADDRESS
        BIC    #20, 2(SP)          ;; CLEAR "T" BIT
        CLR    $TBIT              ;; CLEAR THE "T" BIT FLAG
        RTI
        HALT
        BR     .-2                  ;; THE POWER UP SEQUENCE WAS STARTED
        ;; BEFORE THE POWER DOWN WAS COMPLETE
        $SAVR6: 0                  ;; PUT THE SP HERE
        $POWER: .ASCIZ <15><12>"POWER"
        .EVEN
        .EVEN

```

```

4425
4426          .SBTTL  ERROR HANDLER ROUTINE
4427
4428          ;*****
4429          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
4430          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4431          ;*AND TYPE OUT THE PC OF THE ERROR INSTRUCTION
4432          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4433          ;*SW15=1      HALT ON ERROR
4434          ;*SW13=1      INHIBIT ERROR TYPEOUTS
4435          ;*SW09=1      LOOP ON ERROR
4436          ;*CALL
4437          ;*      ERROR  N      ;ERROR=EMT AND N=ERROR ITEM NUMBER
4438
4439          ;:*****
4440
4441          $ERROR:
4442          7$:      INCB      $ERFLG      ;SET THE ERROR FLAG
4443                   BEQ      7$          ;DON'T LET THE FLAG GO TO ZERO
4444                   MOV      $TSTNM, $DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
4445                   INC      $ERTTL      ;INC THE ERROR COUNT
4446                   MOV      (SP), $ERRPC ;GET ADDRESS OF ERROR INSTRUCTION
4447                   SUB      #2, $ERRPC
4448                   MOVB    $ERRPC, $ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
4449                   BIT      #BIT13, $SWR  ;SKIP TYPEOUT IF SET
4450                   BNE     20$          ;SKIP TYPEOUTS
4451                   TYPE    $CRLF
4452                   MOV      $ERRPC, -(SP) ;SAVE $ERRPC FOR TYPEOUT
4453                   ;ERROR ADDRESS
4454                   JSR     PC, TYOCT    ;GO TYPE--OCTAL ASCII(ALL DIGITS,
4455                   TYPE    , $CRLF
4456
4457          20$:     CMPB     #APTENV, $ENV  ;RUNNING IN APT MODE
4458                   BNE     2$          ;NO, SKIP APT ERROR REPORT
4459                   TST     $MSGTYPE     ;FINISHED LAST MESSAGE?
4460                   BNE     21$          ;IF NOT, WAIT
4461                   MOVB   $ITEMB, $FATAL ;REPORT ERROR NUMBER TO APT
4462                   INC     $MSGTYPE     ;TELL APT TO TAKE ERROR
4463                   BR     22$          ;APT ERROR LOOP
4464                   2$:     TST     $SWR  ;HALT ON ERROR
4465                   BPL     3$          ;SKIP IF CONTINUE
4466                   HALT    ;HALT ON ERROR!
4467                   3$:     BIT      #BIT09, $SWR ;LOOP ON ERROR SWITCH SET
4468                   BEQ     4$          ;BR IF NO
4469                   MOV     $LPERR, (SP) ;FUDGE RETURN FOR LOOPING
4470                   TST     $ESCAPE     ;CHECK FOR AN ESCAPE ADDRESS
4471                   BEQ     5$          ;BR IF NONE
4472                   MOV     $ESCAPE, (SP) ;FUDGE RETURN ADDRESS FOR ESCAPE
4473
4474          5$:     CMP      #SENAD, #42   ;ACT-11 AUTO-ACCEPT?
4475                   BNE     6$          ;BRANCH IF NC
4476                   HALT    ;YES
4477
4478          6$:     RTI

```

4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508

016204 010046
016206 017600 000002
016212 132767 000040 162367
016220 001003
016222 112046
016224 001005
016226 005726
016230 012600
016232 062716 000002
016236 000002
016240 105777 162304
016244 100375
016246 112677 162300
016252 000763

```
.SBTTL TYPE ROUTINE
:*****
:ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
:*
:*CALL:
:*1) USING A TRAP INSTRUCTION
:*   TYPE      .MESADR      ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:*JR
:*   TYPE
:*   MESADR
:*
:;*****
$TYPE:  MOV      RD, -(SP)      ;SAVE RD
        MOV      @2(SP), RD    ;GET ADDRESS OF ASCIZ STRING
        BITB     #APCSUP, $ENVM ;IS CONSOLE SUPPRESSED?
        BNE     60$           ;YES, SKIP TYPE OUT
2$:     MOVB     (RD)+, -(SP)   ;PUSH CHARACTER TO BE TYPED ONTO STACK
        BNE     4$           ;BR IF IT ISN'T THE TERMINATOR
        TST     (SP)+         ;IF TERMINATOR POP IT OFF THE STACK
60$:    MOV      (SP)+, RD     ;RESTORE RD
3$:     ADD      #2, (SP)      ;ADJUST RETURN PC
        RTI                    ;RETURN
4$:     TSTB     @$TPS        ;WAIT UNTIL PRINTER IS READY
        BP      4$
        MOVB    (SP)+, @3$TPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
        BR      2$
```

4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546

SBTTL BINARY TO OCTAL (ASCII) AND TYPE

*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*CALL:

* MOV NUM, -(SP) ;NUMBER TO BE TYPED
* JSR PC, TYPOCT ;CALL FOR TYPEOUT

016254	112767	000005	000070	TYPOCT:	MOVB	#5, \$OCNT	;SET ITERATION COUNT
016262	010446				MOV	R4, -(SP)	;SAVE R4
016264	010546				MOV	R5, -(SP)	;SAVE R5
016266	016605	000006			MOV	6(SP), R5	;PICKUP THE INPUT NUMBER
016272	005004				CLR	R4	;CLEAR THE OUTPUT WORD
016274	006105			1\$:	ROL	R5	;ROTATE MSB INTO "C"
016276	000404				BR	3\$;GO DO MSB
016300	006105			2\$:	ROL	R5	;FORM THIS DIGIT
016302	006105				ROL	R5	
016304	006105				ROL	R5	
016306	010504				MOV	R5, R4	
016310	006104			3\$:	ROL	R4	;GET LSB OF THIS DIGIT
016312	042704	177770			BIC	#177770, R4	;GET RID OF JUNK
016316	052704	000060			BIS	#'0, R4	;MAKE THIS DIGIT ASCII
016322	110467	000022			MOVB	R4, 8\$;SAVE FOR TYPING
016326	104401	016350			TYPE	8\$;GO TYPE THIS DIGIT
016332	105367	000014		7\$:	DECB	\$OCNT	;COUNT BY 1
016336	002360				BGE	2\$;BR IF MORE TO DO
016340	012605			6\$:	MOV	(SP)+, R5	;RESTORE R5
016342	012604				MOV	(SP)+, R4	;RESTORE R4
016344	012616				MOV	(SP)+, (SP)	;SET THE STACK FOR RETURNING
016346	000207				RTS	PC	;RETURN
016350	000			8\$:	.BYTE	00	;STORAGE FOR ASCII DIGIT
016351	000				.BYTE	00	;TERMINATOR FOR TYPE ROUTINE
016352	000000			\$OCNT:	.WORD	0	;OCTAL DIGIT COUNTER

4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590

.SBTTL TRAP DECODER

::*****
:*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
:*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
:*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
:*GO TO THAT ROUTINE.

016354 010046
016356 016600 000002
016362 005740
016364 111000
016366 006300
016370 016000 016410
016374 000200

\$TRAP: MOV R0, -(SP) ::SAVE R0
MOV 2(SP), R0 ::GET TRAP ADDRESS
TST -(R0) ::BACKUP BY 2
MOVB (R0), R0 ::GET RIGHT BYTE OF TRAP
ASL R0 ::POSITION FOR INDEXING
MOV \$TRPAD(R0), R0 ::INDEX TO TABLE
RTS R0 ::GO TO ROUTINE

::THIS IS USE TO HANDLE THE "GETPRI" MACRO

016376 011646
016400 016666 000004 000002
016406 000002

\$TRAP2: MOV (SP), -(SP) ::MOVE THE PC DOWN
MOV 4(SP), 2(SP) ::MOVE THE PSW DOWN
RTI ::RESTORE THE PSW

.SBTTL TRAP TABLE

::THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
:*BY THE "TRAP" INSTRUCTION.

016410 016376
016412 016204

: ROUTINE
:-----
\$TRPAD: .WORD \$TRAP2
\$.TYPE ::CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE

016414 005015 042115 030455
016422 026461 053104 040513
016430 026511 000101

NAME: .ASCIZ <15><12>/MD-11-DVKAI-A/

016434 000200
017034 000200
000001

BUF1: .BLKW 200
BUF2: .BLKW 200
.END

CROSS REFERENCE TABLE -- USER SYMBOLS

UCYFC	000702	664#	4221*	4228*	4235*	4242*											
UCYFEBYT	000714	670#															
UCYFEBUS	000712	669#	851														
UCYFEBGEN	001300	722#	735#														
UCYFEB	000670	659#	1632*	2185*	2979*	3327*	3616*	3852*	4094*	4315							
UCYFEB =	177772	372#															
UCYFEB =	000240	466#															
UCYFEB =	015212	756#	852	948	1046	1155	1265	1371	1479	1577	1625	1746	1847	1954			
		2057	2178	2297	2369	2441	2535	2607	2691	2755	2851	2927	2973	3030			
		3126	3193	3260	3321	3403	3470	3541	3603	3691	3762	3845	3923	4004			
		4087	4261#														
PRC	= 000000	389#															
PR1	= 000040	390#															
PR2	= 000100	391#															
PR3	= 000140	392#															
PR4	= 000200	393#															
PR5	= 000240	394#															
PR6	= 000300	395#															
PR7	= 000340	396#															
PS	= 177776	369#	370														
PSW	= 177776	370#															
PWRVEC	= 000024	461#	590*	691*	4380*	4391*	4390*	4396*	4409*	4410*							
REPCMC	010474	2988#	2999														
REPLC	011634	3336#	3344														
REPMC	004150	1641#	1649														
REPMRC	005760	2194#	2203														
REPSCN	013574	3861#	3872														
REPSKP	012660	3625#	3634														
REPSPN	014530	4103#	4113														
REP11	003756	1586#	1598														
REP30	010306	2936#	2948														
RESVEC	= 000010	456#	700	701*	708*												
REYN	015354	4304	4309#														
SAVR6	000676	662#	780	878	971	1070	1182	1290	1398	1505	1659	1769	1871	1980			
		2084	2195*	2213	2320	2394	2468	2558	2633	2717	2792	2879	2989*	3007			
		3080	3148	3216	3283	3337*	3353	3425	3498	3564	3626*	3643	3715	3788			
		3862*	3881	3957	4029	4104*	4122	4256*	4257*								
		479#	3704	3776	3867												
SCANC	= 076042	479#	3704	3776	3867												
SCN	013614	3852	3867#														
SENO	015372*	4316	4319#														
SKP	012672*	3616	3629#														
SKPC	= 076041	479#	3414	3487	3553	3629											
SPANC	= 076042	479#	3946	4018	4108												
SPCCON	014432	4081	4083	4096#													
SPN	014546	4094	4108#														
SRCAC	000645	650#	851*	1016	1113	1347	1550	1702	1816	1924	2027	2137	2266	2332			
		2406	2486	2574	2646	2811	2891	3023	3097	3165	3233	3295	3370	3442			
		3476	3515	3576	3609	3660	373	3805	3898	3974	4046	4139	4251	4262*			
SRCLN	000644	649#	886	988	1114	1124	122	1235	1307	1348	1442	1452	1522	1703			
		1713	1786	1814	1914	1925	1997	2025	2127	2138	2256	2367	2328	2506			
		2575	2647	2827	2892	3024	3098	3166	3234	3371	3443	3477	3516	3610			
		3661	3733	3806	3899	3975	4047	4140	4250	4261*							
STARCK	= 001100	360#															
STAR*	000716	507	673#														
UCY_MT	= 177774	371#															
WAR	000540	571#	680	712*	719*	1616	2169	2964	3312	3594	3836	4078	4199	4339			

\$MACR3	000630	624#																		
\$MADR4	000634	627#																		
\$MAIL	000566	537	541	590#	4367															
\$MAMS1	000616	611#																		
\$MAMS2	000622	619#																		
\$MAMS3	000626	622#																		
\$MAMS4	000632	625#																		
\$MBADP	000402	537#																		
\$MSGAD	000602	597#																		
\$MSGLG	000604	598#																		
\$MSGTY	000566	591#	674*	4459	4462*															
\$MTYP1	000617	612#																		
\$MTYP2	000623	620#																		
\$MTYP3	000627	623#																		
\$MT =4	000633	626#																		
\$NULL	000554	577#	1636	2189	2983	3331	3620	3856	4099	4319										
\$NWTST=	000301	748#	750	842#	844	936#	938	1035#	1037	1141#	1143	1253#	1255	1358#						
		1360	1468#	1470	1569#	1571	1604#	1606	1733#	1735	1835#	1837	1942#	1944						
		2046#	2048	2157#	2159	2287#	2289	2357#	2359	2430#	2432	2524#	2526	2597#						
		2599	2691#	2683	2754#	2756	2842#	2844	2919#	2921	2954#	2956	3048#	3050						
		3115#	3117	3184#	3186	3252#	3254	3303#	3305	3392#	3394	3461#	3463	3533#						
		3535	3584#	3596	3692#	3684	3753#	3755	3826#	3828	3921#	3923	3995#	3997						
		4067#	4069																	
\$OCNT	016352	4521#	4537*	4545#																
\$OVER	015574	4340	4358	4364	4372#															
\$PASS	000574	594#	716*	1622	2175	2970	3318	3600	3842	4084	4173*	4174*	4211							
\$PASTM	000406	539#																		
\$POWER	016000	4412	4421#																	
\$PWRAD	015754	4414#																		
\$PWRDN	015610	690	4360#	4409																
\$PWARMG	015750	4412#																		
\$PWRUP	015662	4090	4396#																	
\$QUES	000562	582#																		
\$ROCHR=	*****	4582																		
\$RDDEC=	*****	4582																		
\$RDLIN=	*****	4582																		
\$RDOCT=	*****	4582																		
\$RTNAD	015060	4210#																		
\$RTRN	015054	697	699*	705*	4186	4205#														
\$R2A =	*****	4582																		
\$SAVRE=	*****	4582																		
\$SAVR6	015776	4389#	4397	4398*	4399*	4420#														
\$SCOPE	015412	684	4338#																	
\$SETUP=	000037	500#	4172	4339	4416															
\$STUP =	177777	500#																		
\$SVLAD	015540	4348	4360	4366#																
\$SVPC =	003400	514#	519																	
\$SWR =	171400	471#	581	582	756	85	948	1046	1155	1265	1371	1479	1577	1616						
		1745	1847	1954	2057	2169	2297	2368	2441	2535	2606	2691	2765	2851						
		2927	2964	3058	3126	3193	3260	3312	3403	3470	3541	3594	3691	3762						
		3836	3932	4004	4078	4166	4173	4184	4197	4211	4331	4332	4333	4334						
		4339	4351	4353	4354	4359	4360	4361	4366	4369	4372	4375	4415							
\$SWREG	000610	602#	719																	
\$SWRMK=	000300	472#	4334	4335	4355	4356														
\$TBIT	015062	709#	4201*	4211#	4416*															
\$TEBTH	000572	593#	675*	4367*																

ORIGIN	330*	4221	4228	4235	4242															
ORIGIN	467*																			
ORIGIN	332*	2297	2369	2441	2535	2607	2691	2765	2851	2973										
ORIGIN	339*	775	783	790	796	800	804	809	813	820	827	835	873	881	888					
ORIGIN	893*	897	901	906	910	917	923	928	966	974	981	985	992	997	1001					
ORIGIN	1005*	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128					
ORIGIN	1136*	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293					
ORIGIN	1300*	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421					
ORIGIN	1426*	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546					
ORIGIN	1554*	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722					
ORIGIN	1764*	1772	1779	1783	1790	1795	1799	1803	1810	1820	1828	1866	1874	1881	1886					
ORIGIN	1890*	1894	1899	1903	1909	1918	1928	1936	1975	1983	1990	1994	2001	2006	2010					
ORIGIN	2014*	2021	2031	2039	2079	2087	2094	2099	2103	2107	2112	2116	2122	2131	2141					
ORIGIN	2149*	2207	2216	2223	2228	2232	2236	2241	2245	2251	2260	2270	2278	2315	2323					
ORIGIN	2330*	2334	2338	2342	2347	2351	2389	2397	2404	2408	2412	2416	2421	2425	2463					
ORIGIN	2471*	2478	2482	2496	2502	2510	2518	2553	2561	2568	2572	2578	2582	2587	2591					
ORIGIN	2628*	2636	2643	2650	2656	2664	2671	2675	2712	2720	2727	2731	2735	2739	2744					
ORIGIN	2748*	2787	2795	2802	2806	2821	2825	2831	2837	2874	2882	2889	2895	2899	2905					
ORIGIN	2910*	2914	2941	3002	3010	3017	3021	3027	3031	3036	3040	3075	3083	3090	3094					
ORIGIN	3106*	3110	3143	3151	3158	3162	3174	3178	3211	3219	3226	3230	3242	3246	3278					
ORIGIN	3286*	3293	3297	3348	3356	3363	3367	3379	3383	3420	3428	3435	3439	3451	3455					
ORIGIN	3493*	3501	3508	3512	3524	3528	3559	3567	3574	3578	3638	3646	3653	3657	3669					
ORIGIN	3673*	3710	3718	3725	3729	3743	3747	3783	3791	3798	3802	3816	3820	3876	3884					
ORIGIN	3891*	3895	3909	3913	3952	3960	3967	3971	3985	3989	4024	4032	4039	4043	4057					
ORIGIN	4061*	4117	4125	4132	4136	4150	4154													
ORIGIN	467*	4181																		
ORIGIN	336*	775	783	790	796	800	804	809	813	820	827	835	873	881	888					
ORIGIN	339*	897	901	906	910	917	923	928	966	974	981	985	992	997	1001					
ORIGIN	893*	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128					
ORIGIN	1005*	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293					
ORIGIN	1136*	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421					
ORIGIN	1300*	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546					
ORIGIN	1426*	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722					
ORIGIN	1554*																			
ORIGIN	1764*																			
ORIGIN	1890*																			
ORIGIN	2014*																			
ORIGIN	2149*																			
ORIGIN	2330*																			
ORIGIN	2471*																			
ORIGIN	2628*																			
ORIGIN	2748*																			
ORIGIN	2910*																			
ORIGIN	3106*																			
ORIGIN	3286*																			
ORIGIN	3493*																			
ORIGIN	3673*																			
ORIGIN	3891*																			
ORIGIN	4061*																			
ORIGIN	361*	775	783	790	796	800	804	809	813	820	827	835	873	881	888					
ORIGIN	893*	897	901	906	910	917	923	928	966	974	981	985	992	997	1001					
ORIGIN	1005*	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128					
ORIGIN	1136*	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293					
ORIGIN	1300*	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421					
ORIGIN	1426*	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546					
ORIGIN	1554*	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722					

H10

.MAIN. MACY11 27:1005) 22-DEC-76 10:49 PAGE 126
 DVKAI.A.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- MACRO NAMES

SEC 0:25

TYPBIN	4670															
TYPDEC	4670															
TYPNAM	4670															
TYPNUM	4670															
TYPJCS	4670															
TYPJCT	4670															
TYPJXT	4670															
JBYTE	3470	833	925	1026	1134	1245	1459	1560	1720	1826	1934	2037	2147	2276		
JDEST	3360	824	917	1014	1125	1225	1236	1442	1551	1703	1817	1915	1925	2028	2128	
	2138	2257	2267													
SSCHRA	5440															
SSCHTH	5440															
SSSCHD	4670															
SSSCHY	4670	748	842	926	1035	1141	1253	1359	1469	1559	1604	1732	1825	1942	2046	
	2157	2287	2357	2430	2524	2597	2681	2754	2842	2919	2954	3049	3115	3194	3252	
	3303	3392	3461	3532	3584	3682	3753	3826	3921	3995	4067					
SSS	4570															
SSS1	4570															
SSS2	4570															
SSS3	4570															
SSS4	4570															
SSS5	4570															
SSS6	4570															
SSS7	4570															
SSS8	4570															
SSS9	4570															
SSS10	4570															
SSS11	4570															
SSS12	4570															
SSS13	4570															
SSS14	4570															
SSS15	4570															
SSS16	4570															
SSS17	4570															
SSS18	4570															
SSS19	4570															
SSS20	4570															
SSS21	4570															
SSS22	4570															
SSS23	4570															
SSS24	4570															
SSS25	4570															
SSS26	4570															
SSS27	4570															
SSS28	4570															
SSS29	4570															
SSS30	4570															
SSS31	4570															
SSS32	4570															
SSS33	4570															
SSS34	4570															
SSS35	4570															
SSS36	4570															
SSS37	4570															
SSS38	4570															
SSS39	4570															
SSS40	4570															
SSS41	4570															
SSS42	4570															
SSS43	4570															
SSS44	4570															
SSS45	4570															
SSS46	4570															
SSS47	4570															
SSS48	4570															
SSS49	4570															
SSS50	4570															

. ABS. 017434 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DVKAI.A DVKAI.A/SCL/CRF=DVKAI.A.P11
 RUN-TIME: 62 52 5 SECONDS
 RUN-TIME RATE: 745/121=6.1
 CORE USED: 11K (41 PAGES)