

FP11

DIAGNOSTIC NO 3
MD-11-DFFPC-A

EP-DFFPC-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 2

DEC 1976
digital
MADE IN USA

This image displays a grid of 100 small diagnostic charts or data tables, arranged in 10 rows and 10 columns. Each cell contains a small-scale version of the diagnostic information shown in the header. The charts appear to be diagnostic data for a specific system, likely related to the MD-11 aircraft mentioned in the header. The data is organized into columns and rows, with some cells containing text and others containing graphical representations or numerical data. The overall layout is a dense grid of diagnostic information.

FP11

DIAGNOSTIC NO. 3
MD-11-DFFPC-A

EP-DFFPC-A-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

DEC 1976
digital
MADE IN USA

.REM 8

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DFFPC-A-D
 PRODUCT NAME: PDF-11/34 FPP DIAGNOSTIC PART 3
 DATE: DECEMBER 1976
 AUTHOR: ANTHONY VEZZA

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

Vertical text on the left margin, possibly a page number or document identifier, appearing as a series of characters and symbols.

113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168

THE THREE PROGRAMS:

DFFPA DFFPB DFFPC

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/34 FPII-A FLOATING POINT PROCESSOR. THE DESIGN IS AN ATTEMPT TO REACH ALL ROM STATES, TAKE ALL BRANCH MICRO TESTS (BUT'S) AND VERIFY ALL THE LOGIC. THEY CONSIST OF 155 (OCT) INDIVIDUAL TESTS SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY FAULTS WITH A MINIMUM HARDWARE OR SOFTWARE LEVEL. THE TESTS ARE PARTIONED INTO THREE STAND-ALONE PROGRAMS DESCRIBED BELOW.

NOTE THAT ERROR REPORTS IN THESE PROGRAMS ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS HAVE BEEN RUN AND IN MOST CASE THAT THERE IS ONLY A SINGLE POINT FAULT IN THE FPII-A. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. DFFPA

DFFPA TESTS:

LDFPS
STFPS
CFCC
SETF, SETD, SETI AND SETL
STST
LDF AND LDD (ALL SOURCE MODES)
STD (MODE 0 AND 1)
ADDF, ADDD AND SUBD (MOST CONDITIONS)

B. DFFPB

DFFPB TESTS:

ADDF, ADDD AND SJBD (ALL CONDITIONS NOT TESTED IN DFFPA)
CMPD AND CMPF
DIVD AND DIVF
MULD AND MULF
MODD AND MODF

C. DFFPC

DFFPC TESTS:

STF AND STD (ALL MODES)
STCFD AND STCDF
CLRD AND CLRF
NEGF AND NEG0

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224

ABSF AND ABSD
TSTF AND TSTD
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
LDFPS (ALL SOURCE MODES)
LDCIF AND LDCLF
LDCID AND LDCLD
LDEXP
STEPS (ALL DESTINATION MODES)
STCFL AND STCFI
STCDL AND STCDI
STEXP
STST

2. REQUIREMENTS

2.1 EQUIPMENT

A PDP 11/34 (WITH OR WITHOUT CONSOLE), LA30 (OR EQUIVALENT) AND AN FP11-A FLOATING POINT PROCESSOR. NOTE THAT A SPECIAL INTERRUPTS TEST MODULE IS BEING DESIGNED FOR USE IN THE MANUFACTURING ENVIRONMENT. WHEN THIS DEVICE IS PRESENT THE PROGRAM DFFPB WILL MAKE USE OF IT TO TEST THE FPP INTERRUPT ON BUS REQUEST FUNCTIONS.

2.2 STORAGE

ALL THREE PROGRAMS REQUIRE A MEMORY SYSTEM OF AT LEAST 16K TO LOAD AND RUN.

2.3 PRELIMINARY PROGRAMS

THESE THREE DIAGNOSTICS WILL ASSUME THAT THE PDP 11/34 CENTRAL PROCESSOR IS FAULTLESS. THEREFORE WHEN IN DOUBT RUN THE PDP 11/34 PROCESSOR DIAGNOSTICS BEFORE THESE FP11-A DIAGNOSTICS.

3. LOADING PROCEDURE

THE PROGRAMS WILL BE SUPPLIED ON THE 11/34 DIAGNOSTIC MEDIA. REFER TO THE XXDP OPERATING MANUAL FOR FURTHER INFORMATION.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 PROGRAM AND OPERATOR ACTION

225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280

1. LOAD PROGRAM INTO MEMORY
2. LOAD ADDRESS 200
3. SET CONSOLE SWITCHES (IF CONSOLE IS PRESENT)
4. PRESS START
ON FIRST PASS THE PROGRAM WILL IDENTIFY ITSELF. NOTE THAT IF THERE IS NO PHYSICAL CONSOLE THE PROGRAM WILL REQUEST THE OPERATOR FOR INITIAL VALUE FOR THE SOFTWARE SWITCH REGISTER (SEE SECTION 8.5). IF RUNNING UNDER ACT, APT OR CHAIN THIS DOES NOT APPLY.
5. THE PROGRAM WILL LOOP AND AN END OF PASS AND ERROR SUMMARY WILL BE TYPED AT THE END OF EVERY PASS.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

THE SWITCH SETTING ARE:

	OCTAL	
SW<15>=1...	10000	HALT ON ERROR
SW<14>=1...	4000	LOOP ON CURRENT TEST
SW<13>=1...	2000	INHIBIT ERROR TYPE OL'S
SW<12>=1...	1000	INHIBIT T-BIT TRAPPING
SW<11>=1...	400	INHIBIT ITERATIONS
SW<10>=1...	200	RING TTY BELL ON ERROR
SW<9>=1....	100	LOOP ON ERROR
SW<8>=1....	40	LOOP ON TEST SPECIFIED IN SW<6> THROUGH SW<0>
SW<7>=1....	20	PRINT ERROR SUMMARY EVEN IF SW<13>=1 THIS APPLIES ONLY TO PROGRAM DFFPA.
SW<7>=1....	20	DESELECT CORRECT INTERRUPT TEST IN PROGRAM DFFPB. NOTE THAT THIS TEST WILL AUTOMATICALLY BE DESELECTED BY THE ABSENCE OF THE SPECIAL TEST EQUIPMENT DESIGNED TO CONDUCT THIS TEST. IF THIS EQUIPMENT IS NOT INSTALLED THERE IS NO NEED TO DESELECT THIS TEST. THIS APPLIES ONLY TO PROGRAM DFFPB!

6. ERRORS

6.1 SUMMARIES

IN PROGRAM DFFPA TESTS 1 AND 11 HAVE A SPECIAL ERROR SUMMARY FEATURE. THESE TWO TEST RUN MANY TEST PATTERNS THROUGH THE LOGIC. AFTER AN ERROR IS ENCOUNTERED, ONLY THE FIRST FIVE ERRORS ARE REPORTED

281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336

(TYPED ON THE TTY). EVERY ERROR THOUGH IS LOGGED AND AN ERROR SUMMARY IS PRINTED WHEN THE TEST IS COMPLETE. NOTE THAT IS SW<13>=1 THIS SUMMARY WILL NOT BE TYPED UNLESS SW<7>=1. IN OTHER WORDS TO GET JUST AN ERPOR SUMMARY FROM EITHER OF THESE TWO TESTS 1 AND 11 IN PROGRAM OFFPA BOTH SWITCHES 13 AND 7 MUST = 1.

6.2 ERROR RECOVERY

SW<15:9>=0... MOST ERRORS WILL CAUSE EXECUTION TO GO TO THE START OF THE NEXT TEST AFTER THE MESSAGE IS TYPED. A FEW TESTS ARE IN SECTIONS. IN THESE TESTS AN ERROR WILL CAUSE EXECUTION TO GO TO THE NEXT SECTION AFTER THE MESSAGE IS TYPED.

SW<15>=1... THE PROGRAM WILL HALT AFTER TYPING THE ERROR MESSAGE. PRESSING THE CONSOLE CONTINUE WILL CAUSE THE PROGRAM TO CONTINUE AS IF SW<15>=0.

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIMES

LESS THAN 10 SECONDS FOR EACH PROGRAM ON ANY PASS.

8.2 STACK POINTER

THE STACK POINTER IS INITIALIZED TO 1100 IN EACH OF THE THREE PROGRAMS.

8.3 PASS COUNT

THE PROGRAM MAKES ONE PASS FOR EACH END OF PASS MESSAGE TYPED. THE END OF PASS MESSAGE DESCRIBES THE TOTAL NUMBER OF PASSES COMPLETED AND THE TOTAL NUMBER OF ERRORS SINCE THE LAST END OF PASS MESSAGE.

8.4 T-BIT TRAPPING

IF SW<12>=0 EACH PROGRAM WILL RUN WITH TRACE TRAPS ON EVERY OTHER PASS. FIRST PASS WILL NOT ENABLE TRACE TRAPS. NOTE SW<12>=1 DISABLES T-BIT TRAPS.

8.5 SOFTWARE SWITCH REGISTER

337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392

EACH OF THE THREE PROGRAMS WILL RUN WITH OR WITHOUT A CONSOLE SWITCH REGISTER. IF A PHYSICAL CONSOLE SWITCH REGISTER IS PRESENT ON THE SYSTEM, THEN THESE PROGRAMS WILL GO AHEAD AND USE IT FOR THE SWITCH FUNCTIONS DESCRIBED IN 5.1 ABOVE. IF HOWEVER THERE IS NO CONSOLE SWITCH REGISTER ON THE SYSTEM A SOFTWARE SWITCH REGISTER WILL BE USED. THIS SOFTWARE SWITCH REGISTER CAN BE EXAMINED OR MODIFIED AT ANY TIME BY THE USER IF HE TYPES CONTROL G WHILE THE PROGRAM IS RUNNING. THIS CONTROL G WILL CAUSE THE CONTENTS OF THE SOFTWARE SWITCH REGISTER TO BE TYPED ON THE TTY AND ASK THE USER FOR A NEW VALUE. WHEN THE USER TYPES A VALUE AND CARRIAGE RETURN THEN THE PROGRAM WILL RESUME TESTING AT THE SAME POINT AT WHICH IT LEFT OFF WHEN THE USER TYPED CONTROL G. NOTE THAT WHEN NOT RUNNING UNDER ACT, APT OR CHAIN THE USER WILL BE ASKED FOR A SOFTWARE SWITCH REGISTER VALUE AFTER LOADING ADDRESS 200 AND STARTING THE PROGRAM THE FIRST TIME THE PROGRAM IS RUN AFTER LOADING (ONLY IF NO CONSOLE SWITCH REGISTER IS ON THE SYSTEM).

8.6 INTERRUPTS TEST

IN PROGRAM DFFPB THERE IS A SPECIAL TEST FOR CHECKING THE CORRECT FLOWS OF THE FPP. THIS TEST CAN BE RUN ONLY IF A SPECIAL TEST MODULE IS IN THE SYSTEM. THIS MODULE WILL PROBABLY ONLY BE USED IN MANUFACTURING. IF THIS MODULE IS NOT IN THE SYSTEM THIS TEST WILL AUTOMATICALLY BE DESELECTED. IF THIS TEST MODULE IS ON THE SYSTEM AND SW(7)=0 THIS TEST WILL BE RUN. IF SW(7)=1 THIS TEST WILL BE DESELECTED.

8.7 ACT, APT AND XXDP COMPATIBILITY

THESE PROGRAMS ARE FULLY COMPATIBLE WITH:
APT
ACT
XXDP MONITOR AND CHAIN PROGRAMS.

9. PROGRAM DESCRIPTION

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

TEST 1 STF WITH ILLEGAL ACCUMULATOR TEST

THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.

TEST 2 FDST MODE 1, FLOATING MODE, TEST

THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.

TEST 3 FDST MODE 2 TEST

THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.

TEST 4 FDST MODE 2, WITH GR7, TEST

THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.

TEST 5 FDST MODE 4 TEST

THIS IS A TEST OF STD WITH FDST MODE 4.

TEST 6 FDST MODE 3 TEST

THIS IS A TEST OF FDST MODE 3 USING STD.

TEST 7 FDST MODE 5 TEST

THIS IS A TEST OF FDST MODE 5 USING STD.

TEST 10 FDST MODE 6, INDEX MODE, TEST

THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.

TEST 11 FDST MODE 7, INDEX DEFERRED MODE, TEST

THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD.

TEST 12 STCFD TEST

MO1

617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
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
670
671
672

THIS IS A TEST OF SOURCE MODE 4 USING THE LDFPS INSTR

TEST 50 SOURCE MODES, MODE 3 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 3 USING THE LDFPS INSTR

TEST 51 SOURCE MODES, MODE 5 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 5 USING THE LDFPS INSTR

TEST 52 SOURCE MODES, MODE 6 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 6 USING THE LDFPS INSTR

TEST 53 SOURCE MODES, MODE 7 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 7 USING THE LDFPS INSTR

TEST 54 SOURCE MODES, MODE 2 GR7 (FL=1), TEST

THIS IS A TEST OF THE LDCLD WITH IMMEDIATE ADDRESSING MODE

TEST 55 SOURCE MODES, MODE 2 (FL=1), TEST

THIS IS A TEST OF THE LDCLD INSTR WITH MODE 2.

TEST 56 LDCIF AND LDCLF TEST

THIS IS A TEST OF THE LDCIF AND THE LDCLF INSTRUCTIONS.

TEST 57 LDCID AND LDCLD TEST

THIS IS A TEST OF LDCID AND LDCLD

TEST 60 LDEXP TEST

THIS IS A TEST OF THE LDEXP INST A SUBROUTINE IS USED TO SET UP OPERANDS, EXECUTE THE LDEXP INST AND CHECK THE RESULTS.

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

- TEST 61 DESTINATION MODES, MODE 1 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 1 USING THE STFPS
 INSTRUCTION
- TEST 62 DESTINATION MODES, MODE 2 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 2 USING THE STFPS
 INSTRUCTION
- TEST 63 DESTINATION MODES, MODE 4 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 4 USING THE STFPS
 INSTRUCTION
- TEST 64 DESTINATION MODES, MODE 3 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 3 USING THE STFPS
 INSTRUCTION
- TEST 65 DESTINATION MODES, MODE 5 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 5 USING THE STFPS
 INSTRUCTION
- TEST 66 DESTINATION MODES, MODE 6 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 6 USING THE STFPS
 INSTRUCTION
- TEST 67 DESTINATION MODES, MODE 7 (FL=0), TEST

 THIS IS A TEST OF DESTINATION MODE 7 USING THE STFPS
 INSTRUCTION
- TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST

 THIS IS A TEST OF DESTINATION MODE 2 USING STCOL
 WITH REGISTER 0
- TEST 71 DESTINATION MODES, MODE 4 (FL=1), TEST

 THIS IS A TEST OF DESTINATION MODE 4 USING STCOL
 WITH REGISTER 0
- TEST 72 STCDI AND STCDL TEST

729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
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

THIS IS A TEST OF THE STCDI AND STCDL INSTRUCTIONS.
NOTE THAT A SUBROUTINE, STCSJB, IS USED TO SET UP
THE OPERANDS, EXECUTE THE STC INSTRUCTION AND CHECK
THE RESULT.

TEST 73 STCFL AND STCFI TEST

THIS IS A TEST OF STCFL AND STCFI. IT MAKES USE OF
THE SAME SUBROUTINE, STCSUB, WHICH WAS USED TO TEST
STCDL AND STCDI.

TEST 74 STEXP TEST

THIS IS A TEST OF THE STEXP INSTRUCTION

TEST 75 STST TEST

THIS IS A TEST OF THE STST INSTRUCTION. FIRST AN
ILLEGAL FPS OP CODE (INSTRUCTION) IS USED TO ENTER
AN ERROR CONDITION IN THE FEC AND FEA. THE STST IS
EXECUTED AND THE FEC AND FEA ARE CHECKED

10. _LISTING

8
MNUMBER=443
PROGNUM=3

000443
000003

.LIST ME
.NLIST MD,MC,CND

800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840

.ENABL ABS

.TITLE MAINDEC-11-FPP34-A PDP 11/34 FPP DIAGNOSTIC
.*COPYRIGHT (C) AUG 1976
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY ANTHONY S. VEZZA
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
.*

000001
160000

\$TN=1
\$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

000244
177400
000200
000011
000015

FPVECT=244
\$SWR=177400
\$SWRMSK=200
TAB=11
CRLF=15

.SBTTL BASIC DEFINITIONS

001100

.*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

000011

.*MISCELLANEOUS DEFINITIONS
HT= 11 ;;CODE FOR HORIZONTAL TAB

841	000012	LF=	12	:: CODE FOR LINE FEED
842	000015	CR=	15	:: CODE FOR CARRIAGE RETURN
843	000200	CRLF=	200	:: CODE FOR CARRIAGE RETURN-LINE FEED
844	177776	PS=	177776	:: PROCESSOR STATUS WORD
845		.EQUIV	PS, PSW	
846	177774	STKLMT=	177774	:: STACK LIMIT REGISTER
847	177772	PIRQ=	177772	:: PROGRAM INTERRUPT REQUEST REGISTER
848	177570	OSWR=	177570	:: HARDWARE SWITCH REGISTER
849	177570	DDISP=	177570	:: HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS

850		RC=	%0	:: GENERAL REGISTER
851	000000	R1=	%1	:: GENERAL REGISTER
852	000001	R2=	%2	:: GENERAL REGISTER
853	000002	R3=	%3	:: GENERAL REGISTER
854	000003	R4=	%4	:: GENERAL REGISTER
855	000004	R5=	%5	:: GENERAL REGISTER
856	000005	R6=	%6	:: GENERAL REGISTER
857	000006	R7=	%7	:: GENERAL REGISTER
858	000007	SP=	%6	:: STACK POINTER
859	000006	PC=	%7	:: PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS

860	000000	PR0=	0	:: PRIORITY LEVEL 0
861	000040	PR1=	40	:: PRIORITY LEVEL 1
862	000100	PR2=	100	:: PRIORITY LEVEL 2
863	000140	PR3=	140	:: PRIORITY LEVEL 3
864	000200	PR4=	200	:: PRIORITY LEVEL 4
865	000240	PR5=	240	:: PRIORITY LEVEL 5
866	000300	PR6=	300	:: PRIORITY LEVEL 6
867	000340	PR7=	340	:: PRIORITY LEVEL 7

.*"SWITCH REGISTER" SWITCH DEFINITIONS

872	100000	SW15=	100000	
873	040000	SW14=	40000	
874	020000	SW13=	20000	
875	010000	SW12=	10000	
876	004000	SW11=	4000	
877	002000	SW10=	2000	
878	001000	SW09=	1000	
879	000400	SW08=	400	
880	000200	SW07=	200	
881	000100	SW06=	100	
882	000040	SW05=	40	
883	000020	SW04=	20	
884	000010	SW03=	10	
885	000004	SW02=	4	
886	000002	SW01=	2	
887	000001	SW00=	1	
888		.EQUIV	SW09, SW9	
889		.EQUIV	SW08, SW8	
890		.EQUIV	SW07, SW7	
891		.EQUIV	SW06, SW6	
892		.EQUIV	SW05, SW5	
893		.EQUIV	SW04, SW4	
894		.EQUIV	SW03, SW3	

BASIC DEFINITIONS

.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
SPTVEC= 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

.SBTTL FPP REGISTER DEFINITIONS

AC0 =%0
AC1 =%1
AC2 =%2
AC3 =%3
AC4 =%4
AC5 =%5
AC6 =%6
AC7 =%7

.SBTTL 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
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


```

953
954      000000
955
956
957
958      000174
959 000174 000000
960 000176 000000
961
962 000200 000137 006106

```

```

      =0
:*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
      =174
DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
.SBTTL  STARTING ADDRESS(ES)
      JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

```

COMMON TAGS

.SBTTL COMMON TAGS

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

963
964
965
966
967
968
969 001100
970 00110C
971 001100 000000
972 001102 000
973 001103 000
974 001104 000000
975 001106 000000
976 001110 000000
977 001112 000000
978 001114 000
979 001115 001
980 001116 000000
981 001120 000000
982 001122 000000
983 001124 000000
984 001126 000000
985 001130 000000
986 001132 000000
987 001134 000
988 001135 000
989 001136 000000
990 001140 177570
991 001142 177570
992 001144 177560
993 001146 177562
994 001150 177564
995 001152 177566
996 001154 000
997 001155 002
998 001156 012
999 001157 000
1000 001160 000000
1001
1002 001162 000000
1003 001164 000000
1004 001166 000000
1005 001170 000000
1006 001172 000000
1007 001174 000000
1008 001176 000000
1009 001200 000000
1010 001202 000000
1011 001204 000000
1012 001206 000000
1013 001210 000000
1014 001212 000000
1015 001214 000000
1016 001216 000000
1017 001220 000000
1018 001222 000000

SCMTAG: =1100

.WORD 0
STSTNM: .BYTE 00
SERFLG: .BYTE 00
SICNT: .WORD 00
SLPADR: .WORD 00
SLPERR: .WORD 00
SERTTL: .WORD 00
SITEMB: .BYTE 00
SERMAX: .BYTE 1
SERRPC: .WORD 00
SGDADR: .WORD 00
SBDADR: .WORD 00
SGDDAT: .WORD 00
SBDDAT: .WORD 00
SAUTOB: .BYTE 00
SINTAG: .BYTE 0
SWR: .WORD DSWR
DISPLAY: .WORD DDISP
STKS: 177560
STKB: 177562
STPS: 177564
STPB: 177566
SNULL: .BYTE 0
SFILLS: .BYTE 2
SFILLC: .BYTE 12
STPFLG: .BYTE 0
SREGAD: .WORD 0
SREG0: .WORD 0
SREG1: .WORD 00
SREG2: .WORD 00
SREG3: .WORD 00
SREG4: .WORD 00
SREG5: .WORD 00
SREG6: .WORD 00
SREG7: .WORD 00
SREG10: .WORD 00
SREG11: .WORD 00
SREG12: .WORD 00
SREG13: .WORD 00
SREG14: .WORD 00
SREG15: .WORD 00
SREG16: .WORD 00
SREG17: .WORD 00
SREG20: .WORD 0

;; START OF COMMON TAGS
;; CONTAINS THE TEST NUMBER
;; CONTAINS ERROR FLAG
;; CONTAINS SUBTEST ITERATION COUNT
;; CONTAINS SCOPE LOOP ADDRESS
;; CONTAINS SCOPE RETURN FOR ERRORS
;; CONTAINS TOTAL ERRORS DETECTED
;; CONTAINS ITEM CONTROL BYTE
;; CONTAINS MAX. ERRORS PER TEST
;; CONTAINS PC OF LAST ERROR INSTRUCTION
;; CONTAINS ADDRESS OF 'GOOD' DATA
;; CONTAINS ADDRESS OF 'BAD' DATA
;; CONTAINS 'GOOD' DATA
;; CONTAINS 'BAD' DATA
;; RESERVED--NOT TO BE USED
;; AUTOMATIC MODE INDICATOR
;; INTERRUPT MODE INDICATOR
;; ADDRESS OF SWITCH REGISTER
;; ADDRESS OF DISPLAY REGISTER
;; TTY KBD STATUS
;; TTY KBD BUFFER
;; TTY PRINTER STATUS REG. ADDRESS
;; TTY PRINTER BUFFER REG. ADDRESS
;; CONTAINS NULL CHARACTER FOR FILLS
;; CONTAINS # OF FILLER CHARACTERS REQUIRED
;; INSERT FILL CHARS. AFTER A "LINE FEED"
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
;; CONTAINS THE ADDRESS FROM
;; WHICH (SREG0) WAS OBTAINED
;; CONTAINS ((SREGAD)+0)
;; CONTAINS ((SREGAD)+2)
;; CONTAINS ((SREGAD)+4)
;; CONTAINS ((SREGAD)+6)
;; CONTAINS ((SREGAD)+10)
;; CONTAINS ((SREGAD)+12)
;; CONTAINS ((SREGAD)+14)
;; CONTAINS ((SREGAD)+16)
;; CONTAINS ((SREGAD)+20)
;; CONTAINS ((SREGAD)+22)
;; CONTAINS ((SREGAD)+24)
;; CONTAINS ((SREGAD)+26)
;; CONTAINS ((SREGAD)+30)
;; CONTAINS ((SREGAD)+32)
;; CONTAINS ((SREGAD)+34)
;; CONTAINS ((SREGAD)+36)
;; CONTAINS ((SREGAD)+40

1019	001224	000000	\$REG21:	.WORD	0	::	CONTAINS ((SREGAD)+42)	
1020	001226	000000	\$REG22:	.WORD	0	::	CONTAINS ((SREGAD)+44)	
1021	001230	000000	\$REG23:	.WORD	0	::	CONTAINS ((SREGAD)+46)	
1022	001232	000000	STMP0:	.WORD	0	::	USER DEFINED	
1023	001234	000000	STMP1:	.WORD	0	::	USER DEFINED	
1024	001236	000000	STMP2:	.WORD	0	::	USER DEFINED	
1025	001240	000000	STMP3:	.WORD	0	::	USER DEFINED	
1026	001242	000000	STMP4:	.WORD	0	::	USER DEFINED	
1027	001244	000000	STMP5:	.WORD	0	::	USER DEFINED	
1028	001246	000000	STMP6:	.WORD	0	::	USER DEFINED	
1029	001250	000000	STMP7:	.WORD	0	::	USER DEFINED	
1030	001252	000000	STMP10:	.WORD	0	::	USER DEFINED	
1031	001254	000000	STMP11:	.WORD	0	::	USER DEFINED	
1032	001256	000000	STMP12:	.WORD	0	::	USER DEFINED	
1033	001260	000000	STMP13:	.WORD	0	::	USER DEFINED	
1034	001262	000000	STMP14:	.WORD	0	::	USER DEFINED	
1035	001264	000000	STMP15:	.WORD	0	::	USER DEFINED	
1036	001266	000000	STMP16:	.WORD	0	::	USER DEFINED	
1037	001270	000000	STMP17:	.WORD	0	::	USER DEFINED	
1038	001272	000000	STMP20:	.WORD	0	::	USER DEFINED	
1039	001274	000000	STMP21:	.WORD	0	::	USER DEFINED	
1040	001276	000000	STMP22:	.WORD	0	::	USER DEFINED	
1041	001300	000000	STMP23:	.WORD	0	::	USER DEFINED	
1042	001302	000000	STIMES:	0		::	MAX. NUMBER OF ITERATIONS	
1043	001304	000000	SESCAPE:	0		::	ESCAPE ON ERROR ADDRESS	
1044	001306	177607	\$BELL:	.ASCIZ	<207><377><377>	::	CODE FOR BELL	
1045	001312	077	\$QUES:	.ASCII	/?/	::	QUESTION MARK	
1046	001313	015	\$CRLF:	.ASCII	<15>	::	CARRIAGE RETURN	
1047	001314	000012	\$LF:	.ASCIZ	<12>	::	LINE FEED	
1048			;*****					
1049			.SBTTL					APT MAILBOX-ETABLE
1050			;*****					
1051			.EVEN					
1052			\$MAIL:			::	APT MAILBOX	
1053	001316		\$MSGTY:	.WORD	AMSGTY	::	MESSAGE TYPE CODE	
1054	001316	000000	\$FATAL:	.WORD	AFATAL	::	FATAL ERROR NUMBER	
1055	001320	000000	\$TESTN:	.WORD	ATESTN	::	TEST NUMBER	
1056	001322	000000	\$PASS:	.WORD	APASS	::	PASS COUNT	
1057	001324	000000	\$DEVCT:	.WORD	ADEVCT	::	DEVICE COUNT	
1058	001326	000000	\$UNIT:	.WORD	AUNIT	::	I/O UNIT NUMBER	
1059	001330	000000	\$MSGAD:	.WORD	AMSGAD	::	MESSAGE ADDRESS	
1060	001332	000000	\$MSGLG:	.WORD	AMSGLG	::	MESSAGE LENGTH	
1061	001334	000000	\$ETABLE:			::	APT ENVIRONMENT TABLE	
1062	001336		\$ENV:	.BYTE	AENV	::	ENVIRONMENT BYTE	
1063	001336	000	\$ENVM:	.BYTE	AENVM	::	ENVIRONMENT MODE BITS	
1064	001337	000	\$SWREG:	.WORD	ASWREG	::	APT SWITCH REGISTER	
1065	001340	000000	\$USWR:	.WORD	AUSWR	::	USER SWITCHES	
1066	001342	000000	\$CPUOP:	.WORD	ACPUOP	::	CPU TYPE, OPTIONS	
1067	001344	000000	;*					BITS 15-11=CPU TYPE
1068			;*					11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1069			;*					11/70=06, PDQ=07, Q=10
1070			;*					BIT 10=REAL TIME CLOCK
1071			;*					BIT 9=FLOATING POINT PROCESSOR
1072			;*					BIT 8=MEMORY MANAGEMENT
1073			;*					::
1074	001346	000	\$MAMS1:	.BYTE	AMAMS1	::	HIGH ADDRESS, M.S. BYTE	

1075	001347	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE BLK#1
1076			.*		MEM. TYPE BYTE -- (HIGH BYTE)
1077			.*		900 NSEC CORE=001
1078			.*		300 NSEC BIPOLAR=002
1079			.*		500 NSEC MOS=003
1080	001350	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS, BLK#1
1081			.*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1082	001352	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS, M.S. BYTE
1083	001353	000	\$MTYP2: .BYTE	AMTYP2	::MEM. TYPE, BLK#2
1084	001354	000000	\$MADR2: .WORD	AMADR2	::MEM. LAST ADDRESS, BLK#2
1085	001356	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS, M.S. BYTE
1086	001357	000	\$MTYP3: .BYTE	AMTYP3	::MEM. TYPE, BLK#3
1087	001360	000000	\$MADR3: .WORD	AMADR3	::MEM. LAST ADDRESS, BLK#3
1088	001362	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS, M.S. BYTE
1089	001363	000	\$MTYP4: .BYTE	AMTYP4	::MEM. TYPE, BLK#4
1090	001364	000000	\$MADR4: .WORD	AMADR4	::MEM. LAST ADDRESS, BLK#4
1091	001366	000000	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
1092	001370	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
1093	001372	000000	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
1094	001374	000000	\$DEV1: .WORD	ADEV1	:: DEVICE MAP
1095	001376	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
1096	001400	000000	\$CDW2: .WORD	ACDW2	:: CONTROLLER DESCRIPTION WORD#2
1097	001402	000000	\$DDW0: .WORD	ADDW0	:: DEVICE DESCRIPTOR WORD#0
1098	001404	000000	\$DDW1: .WORD	ADDW1	:: DEVICE DESCRIPTOR WORD#1
1099	001406	000000	\$DDW2: .WORD	ADDW2	:: DEVICE DESCRIPTOR WORD#2
1100	001410	000000	\$DDW3: .WORD	ADDW3	:: DEVICE DESCRIPTOR WORD#3
1101	001412	000000	\$DDW4: .WORD	ADDW4	:: DEVICE DESCRIPTOR WORD#4
1102	001414	000000	\$DDW5: .WORD	ADDW5	:: DEVICE DESCRIPTOR WORD#5
1103	001416	000000	\$DDW6: .WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
1104	001420	000000	\$DDW7: .WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
1105	001422	000000	\$DDW8: .WORD	ADDW8	:: DEVICE DESCRIPTOR WORD#8
1106	001424	000000	\$DDW9: .WORD	ADDW9	:: DEVICE DESCRIPTOR WORD#9
1107	001426	000000	\$DDW10: .WORD	ADDW10	:: DEVICE DESCRIPTOR WORD#10
1108	001430	000000	\$DDW11: .WORD	ADDW11	:: DEVICE DESCRIPTOR WORD#11
1109	001432	000000	\$DDW12: .WORD	ADDW12	:: DEVICE DESCRIPTOR WORD#12
1110	001434	000000	\$DDW13: .WORD	ADDW13	:: DEVICE DESCRIPTOR WORD#13
1111	001436	000000	\$DDW14: .WORD	ADDW14	:: DEVICE DESCRIPTOR WORD#14
1112	001440	000000	\$DDW15: .WORD	ADDW15	:: DEVICE DESCRIPTOR WORD#15
1113					
1114					
1115	001442		\$ETEND:		
1116					

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF SITEMB 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

1117									
1118									
1119									
1120									
1121									
1122									
1123									
1124									
1125									
1126									
1127									
1128									
1129									
1130									
1131	001442				\$ERRTB:				
1132					;ITEM 1				
1133	001442	043132	067322	071026	.WORD	EM1,DH1,DT1,DF1			
1134	001450	070460							
1135					;ITEM 2				
1136	001452	043171	067375	071046	.WORD	EM2,DH2,DT2,DF2			
1137	001460	070467							
1138					;ITEM 3				
1139	001462	043224	067465	071070	.WORD	EM3,DH3,DT3,DF3			
1140	001470	070467							
1141					;ITEM 4				
1142	001472	043257	067555	071112	.WORD	EM4,DH4,DT4,DF4			
1143	001500	070467							
1144					;ITEM 5				
1145	001502	043317	067644	071134	.WORD	EM5,DH5,DT5,DF5			
1146	001510	070477							
1147					;ITEM 6				
1148	001512	043341	067644	071162	.WORD	EM6,DH6,DT6,DF6			
1149	001520	070511							
1150					;ITEM 7				
1151	001522	043445	067555	071112	.WORD	EM7,DH7,DT7,DF7			
1152	001530	070467							
1153					;ITEM 10				
1154	001532	043506	067644	071134	.WORD	EM10,DH10,DT10,DF10			
1155	001540	070477							
1156					;ITEM 11				
1157	001542	043531	067555	071112	.WORD	EM11,DH11,DT11,DF11			
1158	001550	070467							
1159					;ITEM 12				
1160	001552	043572	067644	071134	.WORD	EM12,DH12,DT12,DF12			
1161	001560	070515							
1162					;ITEM 13				
1163	001562	043615	067705	071162	.WORD	EM13,DH13,DT13,DF13			
1164	001570	070511							
1165					;ITEM 14				
1166	001572	043615	067705	071162	.WORD	EM14,DH14,DT14,DF14			
1167	001600	070511							
1168					;ITEM 15				
1169	001602	043651	067644	071134	.WORD	EM15,DH15,DT15,DF15			
1170	001610	070515							
1171					;ITEM 16				
1172	001612	043672	067745	071174	.WORD	EM16,DH16,DT16,DF16			

K02

MAINDEC-11-FPP34-A PDP 11-34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 23
 DFFPCA.P11 31-OCT-76 17:16 ERROR POINTER TABLE

1173	001620	070467				
1174					; ITEM 17	
1175	001622	043721	067705	071162	.WORD	E 17, DH17, DT17, DF17
1176	001630	070511				
1177					; ITEM 20	
1178	001632	043757	067555	071174	.WORD	EM20, DH20, DT20, DF20
1179	001640	070467				
1180					; ITEM 21	
1181	001642	044020	067644	071134	.WORD	EM21, DH21, DT21, DF21
1182	001650	070515				
1183					; ITEM 22	
1184	001652	044020	067644	071134	.WORD	EM22, DH22, DT22, DF22
1185	001660	070515				
1186					; ITEM 23	
1187	001662	044043	067705	071162	.WORD	EM23, DH23, DT23, DF23
1188	001670	070511				
1189					; ITEM 24	
1190	001672	044102	067555	071174	.WORD	EM24, DH24, DT24, DF24
1191	001700	070467				
1192					; ITEM 25	
1193	001702	044144	067644	071134	.WORD	EM25, DH25, DT25, DF25
1194	001710	070515				
1195					; ITEM 26	
1196	001712	044170	067705	071162	.WORD	EM26, DH26, DT26, DF26
1197	001720	070511				
1198					; ITEM 27	
1199	001722	044227	067555	071174	.WORD	EM27, DH27, DT27, DF27
1200	001730	070467				
1201					; ITEM 30	
1202	001732	044271	067644	071134	.WORD	EM30, DH30, DT30, DF30
1203	001740	070515				
1204					; ITEM 31	
1205	001742	044315	067705	071162	.WORD	EM31, DH31, DT31, DF31
1206	001750	070511				
1207					; ITEM 32	
1208	001752	044353	067555	071174	.WORD	EM32, DH32, DT32, DF32
1209	001760	070467				
1210					; ITEM 33	
1211	001762	044414	067644	071134	.WORD	EM33, DH33, DT33, DF33
1212	001770	070515				
1213					; ITEM 34	
1214	001772	044437	067705	071162	.WORD	EM34, DH34, DT34, DF34
1215	002000	070511				
1216					; ITEM 35	
1217	002002	044476	067555	071174	.WORD	EM35, DH35, DT35, DF35
1218	002010	070467				
1219					; ITEM 36	
1220	002012	044540	067644	071134	.WORD	EM36, DH36, DT36, DF36
1221	002020	070515				
1222					; ITEM 37	
1223	002022	044564	070034	071216	.WORD	EM37, DH37, DT37, DF37
1224	002030	070527				
1225					; ITEM 40	
1226	002032	044610	070034	071216	.WORD	EM40, DH40, DT40, DF40
1227	002040	070527				
1228					; ITEM 41	

1229	002042	044636	070124	071262	.WORD	EM41,DH41,DT41,DF41
1230	002050	070550				
1231					; ITEM 42	
1232	002052	044664	070034	071216	.WORD	EM42,DH42,DT42,DF42
1233	002060	070527				
1234					; ITEM 43	
1235	002062	044743	070034	071216	.WORD	EM43,DH43,DT43,DF43
1236	002070	070527				
1237					; ITEM 44	
1238	002072	045047	070034	071216	.WORD	EM44,DH44,DT44,DF44
1239	002100	070527				
1240					; ITEM 45	
1241	002102	045147	070034	071216	.WORD	EM45,DH45,DT45,DF45
1242	002110	070527				
1243					; ITEM 46	
1244	002112	045225	070034	071216	.WORD	EM46,DH46,DT46,DF46
1245	002120	070527				
1246					; ITEM 47	
1247	002122	045331	070034	071216	.WORD	EM47,DH47,DT47,DF47
1248	002130	070527				
1249					; ITEM 50	
1250	002132	045431	070034	071216	.WORD	EM50,DH50,DT50,DF50
1251	002140	070527				
1252					; ITEM 51	
1253	002142	045545	070034	071216	.WORD	EM51,DH51,DT51,DF51
1254	002150	070527				
1255					; ITEM 52	
1256	002152	045571	070034	071216	.WORD	EM52,DH52,DT52,DF52
1257	002160	070527				
1258					; ITEM 53	
1259	002162	045615	070124	071262	.WORD	EM53,DH53,DT53,DF53
1260	002170	070527				
1261					; ITEM 54	
1262	002172	045641	070034	071216	.WORD	EM54,DH54,DT54,DF54
1263	002200	070527				
1264					; ITEM 55	
1265	002202	045720	070034	071216	.WORD	EM55,DH55,DT55,DF55
1266	002210	070527				
1267					; ITEM 56	
1268	002212	046046	070034	071216	.WORD	EM56,DH56,DT56,DF56
1269	002220	070527				
1270					; ITEM 57	
1271	002222	046150	070034	071216	.WORD	EM57,DH57,DT57,DF57
1272	002230	070527				
1273					; ITEM 60	
1274	002232	046260	070034	071216	.WORD	EM60,DH60,DT60,DF60
1275	002240	070527				
1276					; ITEM 61	
1277	002242	046370	070034	071216	.WORD	EM61,DH61,DT61,DF61
1278	002250	070527				
1279					; ITEM 62	
1280	002252	046472	067375	071174	.WORD	EM62,DH62,DT62,DF62
1281	002260	070467				
1282					; ITEM 63	
1283	002262	046576	067465	071174	.WORD	EM63,DH63,DT63,DF63
1284	002270	070467				

1285					; ITEM 64	
1286	002272	046621	067644	071134	.WORD	EM64, DH64, DT64, DF64
1287	002300	070477				
1288					; ITEM 65	
1289	002302	046700	067375	071174	.WORD	EM65, DH65, DT65, DF65
1290	002310	070467				
1291					; ITEM 66	
1292	002312	046723	067555	071112	.WORD	EM66, DH66, DT66, DF66
1293	002320	070467				
1294					; ITEM 67	
1295	002322	046762	067375	071112	.WORD	EM67, DH67, DT67, DF67
1296	002330	070467				
1297					; ITEM 70	
1298	002332	047063	067465	071112	.WORD	EM70, DH70, DT70, DF70
1299	002340	070467				
1300					; ITEM 71	
1301	002342	047154	067644	071326	.WORD	EM71, DH71, DT71, DF71
1302	002350	070571				
1303					; ITEM 72	
1304	002352	047173	067375	071112	.WORD	EM72, DH72, DT72, DF72
1305	002360	070467				
1306					; ITEM 73	
1307	002362	047254	067644	071362	.WORD	EM73, DH73, DT73, DF73
1308	002370	070571				
1309					; ITEM 74	
1310	002372	047275	067555	071112	.WORD	EM74, DH74, DT74, DF74
1311	002400	070467				
1312					; ITEM 75	
1313	002402	047317	067375	071046	.WORD	EM75, DH75, DT75, DF75
1314	002410	070467				
1315					; ITEM 76	
1316	002412	047342	067705	071162	.WORD	EM76, DH76, DT76, DF76
1317	002420	070511				
1318					; ITEM 77	
1319	002422	047403	067644	071362	.WORD	EM77, DH77, DT77, DF77
1320	002430	070571				
1321					; ITEM 100	
1322	002432	047425	067555	071112	.WORD	EM100, DH100, DT100, DF100
1323	002440	070467				
1324					; ITEM 101	
1325	002442	047450	067375	071046	.WORD	EM101, DH101, DT101, DF101
1326	002450	070467				
1327					; ITEM 102	
1328	002452	047474	067705	071162	.WORD	EM102, DH102, DT102, DF102
1329	002460	070511				
1330					; ITEM 103	
1331	002462	047535	067644	071362	.WORD	EM103, DH103, DT103, DF103
1332	002470	070571				
1333					; ITEM 104	
1334	002472	047557	067555	071112	.WORD	EM104, DH104, DT104, DF104
1335	002500	070467				
1336					; ITEM 105	
1337	002502	047602	067375	071046	.WORD	EM105, DH105, DT105, DF105
1338	002510	070467				
1339					; ITEM 106	
1340	002512	047626	067705	071162	.WORD	EM106, DH106, DT106, DF106

1341	002520	070511				
1342					: ITEM 107	
1343	002522	047214	067705	071162	.WORD	EM107, DH107, DT107, DF107
1344	002530	070511				
1345					: ITEM 110	
1346	002532	047670	067644	071362	.WORD	EM110, DH110, DT110, DF110
1347	002540	070571				
1348					: ITEM 111	
1349	002542	047713	067555	071112	.WORD	EM111, DH111, DT111, DF111
1350	002550	070467				
1351					: ITEM 112	
1352	002552	047737	067375	071046	.WORD	EM112, DH112, DT112, DF112
1353	002560	070467				
1354					: ITEM 113	
1355	002562	047764	067705	071162	.WORD	EM113, DH113, DT113, DF113
1356	002570	070511				
1357					: ITEM 114	
1358	002572	050026	067644	071362	.WORD	EM114, DH114, DT114, DF114
1359	002600	070571				
1360					: ITEM 115	
1361	002602	050051	067555	071112	.WORD	EM115, DH115, DT115, DF115
1362	002610	070467				
1363					: ITEM 116	
1364	002612	050075	067375	071046	.WORD	EM116, DH116, DT116, DF116
1365	002620	070467				
1366					: ITEM 117	
1367	002622	050122	067705	071162	.WORD	EM117, DH117, DT117, DF117
1368	002630	070511				
1369					: ITEM 120	
1370	002632	050163	067644	071362	.WORD	EM120, DH120, DT120, DF120
1371	002640	070571				
1372					: ITEM 121	
1373	002642	050205	067555	071112	.WORD	EM121, DH121, DT121, DF121
1374	002650	070467				
1375					: ITEM 122	
1376	002652	050230	067375	071046	.WORD	EM122, DH122, DT122, DF122
1377	002660	070467				
1378					: ITEM 123	
1379	002662	050254	067705	071162	.WORD	EM123, DH123, DT123, DF123
1380	002670	070511				
1381					: ITEM 124	
1382	002672	050316	067644	071362	.WORD	EM124, DH124, DT124, DF124
1383	002700	070571				
1384					: ITEM 125	
1385	002702	050341	067555	071112	.WORD	EM125, DH125, DT125, DF125
1386	002710	070467				
1387					: ITEM 126	
1388	002712	050365	067375	071046	.WORD	EM126, DH126, DT126, DF126
1389	002720	070467				
1390					: ITEM 127	
1391	002722	050412	067705	071162	.WORD	EM127, DH127, DT127, DF127
1392	002730	070511				
1393					: ITEM 130	
1394	002732	050454	067644	071362	.WORD	EM130, DH130, DT130, DF130
1395	002740	070571				
1396					: ITEM 131	

1397	002742	050477	067375	071046	.WORD	EM131,DH131,DT131,DF131
1398	002750	070467				
1400	002752	050524	067705	071162	.WORD	EM132,DH132,DT132,DF132
1401	002760	070511				
1402	002762	050567	067644	071362	.WORD	EM133,DH133,DT133,DF133
1403	002770	070571				
1404	002772	050613	067375	071046	.WORD	EM134,DH134,DT134,DF134
1405	003000	070467				
1406	003002	050641	067644	071134	.WORD	EM135,DH135,DT135,DF135
1407	003010	070515				
1408	003012	050714	067644	071134	.WORD	EM136,DH136,DT136,DF136
1409	003020	070515				
1410	003022	050733	067375	071174	.WORD	EM137,DH137,DT137,DF137
1411	003030	070467				
1412	003032	050754	067644	071134	.WORD	EM140,DH140,DT140,DF140
1413	003040	070515				
1414	003042	050775	067555	071112	.WORD	EM141,DH141,DT141,DF141
1415	003050	070467				
1416	003052	051044	067375	071112	.WORD	EM142,DH142,DT142,DF142
1417	003060	070467				
1418	003062	051067	067644	071134	.WORD	EM143,DH143,DT143,DF143
1419	003070	070515				
1420	003072	051111	067555	071112	.WORD	EM144,DH144,DT144,DF144
1421	003100	070467				
1422	003102	051161	067375	071112	.WORD	EM145,DH145,DT145,DF145
1423	003110	070467				
1424	003112	051205	067644	071134	.WORD	EM146,DH146,DT146,DF146
1425	003120	070515				
1426	003122	051227	067555	071112	.WORD	EM147,DH147,DT147,DF147
1427	003130	070467				
1428	003132	051277	067375	071112	.WORD	EM150,DH150,DT150,DF150
1429	003140	070467				
1430	003142	051323	067644	071134	.WORD	EM151,DH151,DT151,DF151
1431	003150	070515				
1432	003152	051346	067555	071112	.WORD	EM152,DH152,DT152,DF152
1433	003160	070467				
1434	003162	051417	067375	071112	.WORD	EM153,DH153,DT153,DF153
1435	003170	070467				

1453					: ITEM 154	
1454	003172	051444	067644	071134	.WORD	EM154,DM154,DT154,DF154
1455	003200	070515				
1456					: ITEM 155	
1457	003202	051467	067555	071112	.WORD	EM155,DM155,DT155,DF155
1458	003210	070467				
1459					: ITEM 156	
1460	003212	051540	067375	071112	.WORD	EM156,DM156,DT156,DF156
1461	003220	070467				
1462					: ITEM 157	
1463	003222	051565	067644	071134	.WORD	EM157,DM157,DT157,DF157
1464	003230	070515				
1465					: ITEM 160	
1466	003232	051607	067555	071112	.WORD	EM160,DM160,DT160,DF160
1467	003240	070467				
1468					: ITEM 161	
1469	003242	051701	067375	071112	.WORD	EM161,DM161,DT161,DF161
1470	003250	070467				
1471					: ITEM 162	
1472	003252	051725	067644	071134	.WORD	EM162,DM162,DT162,DF162
1473	003260	070515				
1474					: ITEM 163	
1475	003262	051750	067375	071112	.WORD	EM163,DM163,DT163,DF163
1476	003270	070467				
1477					: ITEM 164	
1478	003272	051775	067745	071112	.WORD	EM164,DM164,DT164,DF164
1479	003300	070467				
1480					: ITEM 165	
1481	003302	052573	070034	071216	.WORD	EM165,DM165,DT165,DF165
1482	003310	070527				
1483					: ITEM 166	
1484	003312	052614	070034	071216	.WORD	EM166,DM166,DT166,DF166
1485	003320	070527				
1486					: ITEM 167	
1487	003322	052635	070034	071216	.WORD	EM167,DM167,DT167,DF167
1488	003330	070527				
1489					: ITEM 170	
1490	003332	052656	070034	071216	.WORD	EM170,DM170,DT170,DF170
1491	003340	070527				
1492					: ITEM 171	
1493	003342	052701	070034	071216	.WORD	EM171,DM171,DT171,DF171
1494	003350	070527				
1495					: ITEM 172	
1496	003352	052724	070034	071216	.WORD	EM172,DM172,DT172,DF172
1497	003360	070527				
1498					: ITEM 173	
1499	003362	052747	070124	071262	.WORD	EM173,DM173,DT173,DF173
1500	003370	070550				
1501					: ITEM 174	
1502	003372	052772	070124	071262	.WORD	EM174,DM174,DT174,DF174
1503	003400	070550				
1504					: ITEM 175	
1505	003402	053015	070124	071262	.WORD	EM175,DM175,DT175,DF175
1506	003410	070550				
1507					: ITEM 176	
1508	003412	047106	067375	071112	.WORD	EM176,DM176,DT176,DF176

1509	003420	070467				
1510					; ITEM 177	
1511	003422	047131	067465	071112	.WORD	EM177, DH177, DT177, DF177
1512	003430	070467				
1513					; ITEM 200	
1514	003432	053040	070034	071216	.WORD	EM200, DH200, DT200, DF200
1515	003440	070527				
1516					; ITEM 201	
1517	003442	053115	070034	071216	.WORD	EM201, DH201, DT201, DF201
1518	003450	070527				
1519					; ITEM 202	
1520	003452	053216	070034	071216	.WORD	EM202, DH202, DT202, DF202
1521	003460	070527				
1522					; ITEM 203	
1523	003462	053317	070034	071216	.WORD	EM203, DH203, DT203, DF203
1524	003470	070527				
1525					; ITEM 204	
1526	003472	053477	070034	071216	.WORD	EM204, DH204, DT204, DF204
1527	003500	070527				
1528					; ITEM 205	
1529	003502	053554	070034	071216	.WORD	EM205, DH205, DT205, DF205
1530	003510	070527				
1531					; ITEM 206	
1532	003512	053653	070034	071216	.WORD	EM206, DH206, DT206, DF206
1533	003520	070527				
1534					; ITEM 207	
1535	003522	053754	070034	071216	.WORD	EM207, DH207, DT207, DF207
1536	003530	070527				
1537					; ITEM 210	
1538	003532	054053	070034	071216	.WORD	EM210, DH210, DT210, DF210
1539	003540	070527				
1540					; ITEM 211	
1541	003542	054152	070034	071216	.WORD	EM211, DH211, DT211, DF211
1542	003550	070527				
1543					; ITEM 212	
1544	003552	054260	070034	071216	.WORD	EM212, DH212, DT212, DF212
1545	003560	070527				
1546					; ITEM 213	
1547	003562	054361	070034	071216	.WORD	EM213, DH213, DT213, DF213
1548	003570	070527				
1549					; ITEM 214	
1550	003572	054506	070034	071216	.WORD	EM214, DH214, DT214, DF214
1551	003600	070527				
1552					; ITEM 215	
1553	003602	052051	067745	071112	.WORD	EM215, DH215, DT215, DF215
1554	003610	070467				
1555					; ITEM 216	
1556	003612	052202	067644	071134	.WORD	EM216, DH216, DT216, DF216
1557	003620	070515				
1558					; ITEM 217	
1559	003622	052224	067555	071112	.WORD	EM217, DH217, DT217, DF217
1560	003630	070467				
1561					; ITEM 220	
1562	003632	052274	067375	071112	.WORD	EM220, DH220, DT220, DF220
1563	003640	070467				
1564					; ITEM 221	

E03

MAINDEC-11-FPP34-A POP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 30
 DFFPCA.P11 31-OCT-76 17:16 ERROR POINTER TABLE

1565	003642	052320	067745	071112	.WORD	EM221,DH221,DT221,DF221
1566	003650	070467				
1567					; ITEM 222	
1568	003652	052452	067644	071134	.WORD	EM222,DH222,DT222,DF222
1569	003660	070515				
1570					; ITEM 223	
1571	003662	052475	067555	071112	.WORD	EM223,DH223,DT223,DF223
1572	003670	070467				
1573					; ITEM 224	
1574	003672	052546	067375	071112	.WORD	EM224,DH224,DT224,DF224
1575	003700	070467				
1576					; ITEM 225	
1577	003702	054633	067555	071112	.WORD	EM225,DH225,DT225,DF225
1578	003710	070606				
1579					; ITEM 226	
1580	003712	054656	067375	071112	.WORD	EM226,DH226,DT226,DF226
1581	003720	070606				
1582					; ITEM 227	
1583	003722	054702	070221	071162	.WORD	EM227,DH227,DT227,DF227
1584	003730	070616				
1585					; ITEM 230	
1586	003732	054732	067555	071112	.WORD	EM230,DH230,DT230,DF230
1587	003740	070606				
1588					; ITEM 231	
1589	003742	054756	067375	071112	.WORD	EM231,DH231,DT231,DF231
1590	003750	070606				
1591					; ITEM 232	
1592	003752	055003	070221	071162	.WORD	EM232,DH232,DT232,DF232
1593	003760	070616				
1594					; ITEM 233	
1595	003762	055034	067555	071112	.WORD	EM233,DH233,DT233,DF233
1596	003770	070606				
1597					; ITEM 234	
1598	003772	055060	067375	071112	.WORD	EM234,DH234,DT234,DF234
1599	004000	070606				
1600					; ITEM 235	
1601	004002	055105	070221	071162	.WORD	EM235,DH235,DT235,DF235
1602	004010	070616				
1603					; ITEM 236	
1604	004012	055136	067555	071112	.WORD	EM236,DH236,DT236,DF236
1605	004020	070606				
1606					; ITEM 237	
1607	004022	055163	067375	071112	.WORD	EM237,DH237,DT237,DF237
1608	004030	070606				
1609					; ITEM 240	
1610	004032	055211	070221	071162	.WORD	EM240,DH240,DT240,DF240
1611	004040	070616				
1612					; ITEM 241	
1613	004042	055243	067555	071112	.WORD	EM241,DH241,DT241,DF241
1614	004050	070606				
1615					; ITEM 242	
1616	004052	055270	067375	071112	.WORD	EM242,DH242,DT242,DF242
1617	004060	070606				
1618					; ITEM 243	
1619	004062	055316	070221	071162	.WORD	EM243,DH243,DT243,DF243
1620	004070	070616				

F03

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 31
 DFFPCA.P11 31-OCT-76 17:16 ERROR POINTER TABLE

1621					: ITEM 244	
1622	004072	055350	067555	071112	.WORD	EM244, DH244, DT244, DF244
1623	004100	070606				
1624					: ITEM 245	
1625	004102	055374	067375	071112	.WORD	EM245, DH245, DT245, DF245
1626	004110	070606				
1627					: ITEM 246	
1628	004112	055421	067745	071112	.WORD	EM246, DH246, DT246, DF246
1629	004120	070606				
1630					: ITEM 247	
1631	004122	055452	070221	071162	.WORD	EM247, DH247, DT247, DF247
1632	004130	070616				
1633					: ITEM 250	
1634	004132	055503	067555	071112	.WORD	EM250, DH250, DT250, DF250
1635	004140	070606				
1636					: ITEM 251	
1637	004142	055530	067375	071112	.WORD	EM251, DH251, DT251, DF251
1638	004150	070606				
1639					: ITEM 252	
1640	004152	055556	067745	071112	.WORD	EM252, DH252, DT252, DF252
1641	004160	070606				
1642					: ITEM 253	
1643	004162	055610	070221	071162	.WORD	EM253, DH253, DT253, DF253
1644	004170	070616				
1645					: ITEM 254	
1646	004172	055642	067745	071112	.WORD	EM254, DH254, DT254, DF254
1647	004200	070606				
1648					: ITEM 255	
1649	004202	055676	070221	071162	.WORD	EM255, DH255, DT255, DF255
1650	004210	070616				
1651					: ITEM 256	
1652	004212	055732	067555	071112	.WORD	EM256, DH256, DT256, DF256
1653	004220	070606				
1654					: ITEM 257	
1655	004222	055760	067375	071112	.WORD	EM257, DH257, DT257, DF257
1656	004230	070606				
1657					: ITEM 260	
1658	004232	056007	070034	071216	.WORD	EM260, DH260, DT260, DF260
1659	004240	070622				
1660					: ITEM 261	
1661	004242	056044	070034	071216	.WORD	EM261, DH261, DT261, DF261
1662	004250	070622				
1663					: ITEM 262	
1664	004252	056103	070034	071216	.WORD	EM262, DH262, DT262, DF262
1665	004260	070622				
1666					: ITEM 263	
1667	004262	056203	070034	071216	.WORD	EM263, DH263, DT263, DF263
1668	004270	070622				
1669					: ITEM 264	
1670	004272	056231	070034	071216	.WORD	EM264, DH264, DT264, DF264
1671	004300	070622				
1672					: ITEM 265	
1673	004302	056326	070034	071216	.WORD	EM265, DH265, DT265, DF265
1674	004310	070622				
1675					: ITEM 266	
1676	004312	056417	070034	071216	.WORD	EM266, DH266, DT266, DF266

1677	004320	070622				
1678					; ITEM 267	
1679	004322	056532	070034	071216	.WORD	EM267, DH267, DT267, DF267
1680	004330	070622				
1681					; ITEM 270	
1682	004332	056627	070034	071216	.WORD	EM270, DH270, DT270, DF270
1683	004340	070622				
1684					; ITEM 271	
1685	004342	056670	070034	071216	.WORD	EM271, DH271, DT271, DF271
1686	004350	070622				
1687					; ITEM 272	
1688	004352	056736	070034	071216	.WORD	EM272, DH272, DT272, DF272
1689	004360	070622				
1690					; ITEM 273	
1691	004362	057027	070034	071216	.WORD	EM273, DH273, DT273, DF273
1692	004370	070643				
1693					; ITEM 274	
1694	004372	057064	070034	071216	.WORD	EM274, DH274, DT274, DF274
1695	004400	070643				
1696					; ITEM 275	
1697	004402	057123	070034	071216	.WORD	EM275, DH275, DT275, DF275
1698	004410	070643				
1699					; ITEM 276	
1700	004412	057223	070034	071216	.WORD	EM276, DH276, DT276, DF276
1701	004420	070643				
1702					; ITEM 277	
1703	004422	057320	070034	071216	.WORD	EM277, DH277, DT277, DF277
1704	004430	070643				
1705					; ITEM 300	
1706	004432	057374	070034	071216	.WORD	EM300, DH300, DT300, DF300
1707	004440	070643				
1708					; ITEM 301	
1709	004442	057471	070034	071416	.WORD	EM301, DH301, DT301, DF301
1710	004450	070664				
1711					; ITEM 302	
1712	004452	057515	070034	071416	.WORD	EM302, DH302, DT302, DF302
1713	004460	070664				
1714					; ITEM 303	
1715	004462	057543	070124	071470	.WORD	EM303, DH303, DT303, DF303
1716	004470	070710				
1717					; ITEM 304	
1718	004472	057571	070034	071416	.WORD	EM304, DH304, DT304, DF304
1719	004500	070664				
1720					; ITEM 305	
1721	004502	057660	070034	071416	.WORD	EM305, DH305, DT305, DF305
1722	004510	070664				
1723					; ITEM 306	
1724	004512	057763	070034	071416	.WORD	EM306, DH306, DT306, DF306
1725	004520	070664				
1726					; ITEM 307	
1727	004522	060150	070034	071416	.WORD	EM307, DH307, DT307, DF307
1728	004530	070664				
1729					; ITEM 310	
1730	004532	060252	070034	071416	.WORD	EM310, DH310, DT310, DF310
1731	004540	070664				
1732					; ITEM 311	

1733	004542	060355	070034	071416	.WORD	EM311, DH311, DT311, DF311
1734	004550	070664				
1735					; ITEM 312	
1736	004552	060456	070034	071416	.WORD	EM312, DH312, DT312, DF312
1737	004560	070664				
1738					; ITEM 313	
1739	004562	060560	070034	071416	.WORD	EM313, DH313, DT313, DF313
1740	004570	070664				
1741					; ITEM 314	
1742	004572	060661	070034	071416	.WORD	EM314, DH314, DT314, DF314
1743	004600	070664				
1744					; ITEM 315	
1745	004602	060762	070034	071416	.WORD	EM315, DH315, DT315, DF315
1746	004610	070664				
1747					; ITEM 316	
1749	004612	061063	070034	071416	.WORD	EM316, DH316, DT316, DF316
1749	004620	070664				
1750					; ITEM 317	
1751	004622	061164	070034	071416	.WORD	EM317, DH317, DT317, DF317
1752	004630	070664				
1753					; ITEM 320	
1754	004632	061265	070034	071416	.WORD	EM320, DH320, DT320, DF320
1755	004640	070664				
1756					; ITEM 321	
1757	004642	061366	070034	071416	.WORD	EM321, DH321, DT321, DF321
1758	004650	070664				
1759					; ITEM 322	
1760	004652	061467	070034	071542	.WORD	EM322, DH322, DT322, DF322
1761	004660	070734				
1762					; ITEM 323	
1763	004662	061524	070034	071542	.WORD	EM323, DH323, DT323, DF323
1764	004670	070734				
1765					; ITEM 324	
1766	004672	061563	070124	071606	.WORD	EM324, DH324, DT324, DF324
1767	004700	070755				
1768					; ITEM 325	
1769	004702	061622	070034	071542	.WORD	EM325, DH325, DT325, DF325
1770	004710	070734				
1771					; ITEM 326	
1772	004712	061622	070034	071542	.WORD	EM326, DH326, DT326, DF326
1773	004720	070734				
1774					; ITEM 327	
1775	004722	061763	070034	071542	.WORD	EM327, DH327, DT327, DF327
1776	004730	070734				
1777					; ITEM 330	
1778	004732	062065	070034	071542	.WORD	EM330, DH330, DT330, DF330
1779	004740	070734				
1780					; ITEM 331	
1781	004742	062170	070034	071542	.WORD	EM331, DH331, DT331, DF331
1782	004750	070734				
1783					; ITEM 332	
1784	004752	063444	070034	071542	.WORD	EM332, DH332, DT332, DF332
1785	004760	070734				
1786					; ITEM 333	
1787	004762	061524	070034	071542	.WORD	EM333, DH333, DT333, DF333
1788	004770	070734				

1789					; ITEM 334	
1790	004772	062273	070034	071542	.WORD	EM334, DH334, DT334, DF334
1791	005000	070734				
1792					; ITEM 335	
1793	005002	062367	070034	071542	.WORD	EM335, DH335, DT335, DF335
1794	005010	070734				
1795					; ITEM 336	
1796	005012	062471	070034	071542	.WORD	EM336, DH336, DT336, DF336
1797	005020	070734				
1798					; ITEM 337	
1799	005022	062545	070034	071542	.WORD	EM337, DH337, DT337, DF337
1800	005030	070734				
1801					; ITEM 340	
1802	005032	062647	070034	071542	.WORD	EM340, DH340, DT340, DF340
1803	005040	070734				
1804					; ITEM 341	
1805	005042	062751	070034	071542	.WORD	EM341, DH341, DT341, DF341
1806	005050	070734				
1807					; ITEM 342	
1808	005052	063055	070034	071542	.WORD	EM342, DH342, DT342, DF342
1809	005060	070734				
1810					; ITEM 343	
1811	005062	063157	070034	071542	.WORD	EM343, DH343, DT343, DF343
1812	005070	070734				
1813					; ITEM 344	
1814	005072	063261	070034	071542	.WORD	EM344, DH344, DT344, DF344
1815	005100	070734				
1816					; ITEM 345	
1817	005102	063536	070034	071542	.WORD	EM345, DH345, DT345, DF345
1818	005110	070734				
1819					; ITEM 346	
1820	005112	063636	070034	071542	.WORD	EM346, DH346, DT346, DF346
1821	005120	070734				
1822					; ITEM 347	
1823	005122	063734	070034	071542	.WORD	EM347, DH347, DT347, DF347
1824	005130	070776				
1825					; ITEM 350	
1826	005132	063760	070034	071542	.WORD	EM350, DH350, DT350, DF350
1827	005140	070776				
1828					; ITEM 351	
1829	005142	064006	067705	071162	.WORD	EM351, DH351, DT351, DF351
1830	005150	070616				
1831					; ITEM 352	
1832	005152	064112	070034	071542	.WORD	EM352, DH352, DT352, DF352
1833	005160	070776				
1834					; ITEM 353	
1835	005162	064216	070034	071542	.WORD	EM353, DH353, DT353, DF353
1836	005170	070776				
1837					; ITEM 354	
1838	005172	064322	070034	071542	.WORD	EM354, DH354, DT354, DF354
1839	005200	070776				
1840					; ITEM 355	
1841	005202	064426	070034	071542	.WORD	EM355, DH355, DT355, DF355
1842	005210	070776				
1843					; ITEM 356	
1844	005212	064532	067555	071046	.WORD	EM356, DH356, DT356, DF356

1845	005220	070606							
1846									
1847	005222	064630	070261	071070	; ITEM 357	.WORD	EM357, DH357, DT357, DF357		
1848	005230	070606							
1849									
1850	005232	064726	067705	071162	; ITEM 360	.WORD	EM360, DH360, DT360, DF360		
1851	005240	070616							
1852									
1853	005242	067156	067375	071416	; ITEM 361	.WORD	EM361, DH361, DT361, DF361		
1854	005250	070606							
1855									
1856	005252	000000	000000	000000	; ITEM 362	.WORD	EM362, DH362, DT362, DF362		
1857	005260	000000							
1858									
1859	005262	000000	000000	000000	; ITEM 363	.WORD	EM363, DH363, DT363, DF363		
1860	005270	000000							
1861									
1862	005272	000000	000000	000000	; ITEM 364	.WORD	EM364, DH364, DT364, DF364		
1863	005300	000000							
1864									
1865	005302	000000	000000	000000	; ITEM 365	.WORD	EM365, DH365, DT365, DF365		
1866	005310	000000							
1867									
1868	005312	000000	000000	000000	; ITEM 366	.WORD	EM366, DH366, DT366, DF366		
1869	005320	000000							
1870									
1871	005322	000000	000000	000000	; ITEM 367	.WORD	EM367, DH367, DT367, DF367		
1872	005330	000000							
1873									
1874	005332	000000	000000	000000	; ITEM 370	.WORD	EM370, DH370, DT370, DF370		
1875	005340	000000							
1876									
1877	005342	000000	000000	000000	; ITEM 371	.WORD	EM371, DH371, DT371, DF371		
1878	005350	000000							
1879									
1880	005352	000000	000000	000000	; ITEM 372	.WORD	EM372, DH372, DT372, DF372		
1881	005360	000000							
1882									
1883	005362	000000	000000	000000	; ITEM 373	.WORD	EM373, DH373, DT373, DF373		
1884	005370	000000							
1885									
1886	005372	000000	000000	000000	; ITEM 374	.WORD	EM374, DH374, DT374, DF374		
1887	005400	000000							
1888									
1889	005402	000000	000000	000000	; ITEM 375	.WORD	EM375, DH375, DT375, DF375		
1890	005410	000000							
1891									
1892	005412	000000	000000	000000	; ITEM 376	.WORD	EM376, DH376, DT376, DF376		
1893	005420	000000							
1894									
1895	005422	000000	000000	000000	; ITEM 377	.WORD	EM377, DH377, DT377, DF377		
1896	005430	000000							
1897									
1898	005432	000000	000000	000000	; ITEM 400	.WORD	EM400, DH400, DT400, DF400		
1899	005440	000000							
1900									

; ITEM 401

K03

MAINDEC-11-FPP34-A
DFFPCA.P11

31-OCT-76

POP 11/34 FPP DIAGNOSTIC
17:16

ERROR POINTER TABLE

MACY11 27(1006)

31-OCT-76

17:35

PAGE 36

1901	005442	065021	067555	071112	.WORD	EM401, DH401, DT401, DF401
1902	005450	070606				
1903					; ITEM 402	
1904	005452	065044	067375	071112	.WORD	EM402, DH402, DT402, DF402
1905	005460	070606				
1906					; ITEM 403	
1907	005462	065066	067705	071162	.WORD	EM403, DH403, DT403, DF403
1908	005470	070616				
1909					; ITEM 404	
1910	005472	065220	070221	071162	.WORD	EM404, DH404, DT404, DF404
1911	005500	070616				
1912					; ITEM 405	
1913	005502	065250	067555	071112	.WORD	EM405, DH405, DT405, DF405
1914	005510	070606				
1915					; ITEM 406	
1916	005512	065274	067375	071112	.WORD	EM406, DH406, DT406, DF406
1917	005520	070606				
1918					; ITEM 407	
1919	005522	065317	067705	071162	.WORD	EM407, DH407, DT407, DF407
1920	005530	070616				
1921					; ITEM 410	
1922	005532	065452	070221	071162	.WORD	EM410, DH410, DT410, DF410
1923	005540	070616				
1924					; ITEM 411	
1925	005542	065503	067555	071112	.WORD	EM411, DH411, DT411, DF411
1926	005550	070606				
1927					; ITEM 412	
1928	005552	065527	067375	071112	.WORD	EM412, DH412, DT412, DF412
1929	005560	070606				
1930					; ITEM 413	
1931	005562	065552	067705	071162	.WORD	EM413, DH413, DT413, DF413
1932	005570	070616				
1933					; ITEM 414	
1934	005572	065705	070221	071162	.WORD	EM414, DH414, DT414, DF414
1935	005600	070616				
1936					; ITEM 415	
1937	005602	065736	067555	071112	.WORD	EM415, DH415, DT415, DF415
1938	005610	070606				
1939					; ITEM 416	
1940	005612	065763	067375	071112	.WORD	EM416, DH416, DT416, DF416
1941	005620	070606				
1942					; ITEM 417	
1943	005622	066007	067705	071162	.WORD	EM417, DH417, DT417, DF417
1944	005630	070616				
1945					; ITEM 420	
1946	005632	066055	070221	071162	.WORD	EM420, DH420, DT420, DF420
1947	005640	070616				
1948					; ITEM 421	
1949	005642	066107	067555	071112	.WORD	EM421, DH421, DT421, DF421
1950	005650	070606				
1951					; ITEM 422	
1952	005652	066134	067375	071112	.WORD	EM422, DH422, DT422, DF422
1953	005660	070606				
1954					; ITEM 423	
1955	005662	066160	067705	071162	.WORD	EM423, DH423, DT423, DF423
1956	005670	070616				

1957					; ITEM 424	
1958	005672	066226	070221	071162	.WORD	EM424, DM424, DT424, DF424
1959	005700	070616				
1960					; ITEM 425	
1961	005702	066260	067555	071112	.WORD	EM425, DM425, DT425, DF425
1962	005710	070606				
1963					; ITEM 426	
1964	005712	066304	067375	071112	.WORD	EM426, DM426, DT426, DF426
1965	005720	070606				
1966					; ITEM 427	
1967	005722	066327	067705	071162	.WORD	EM427, DM427, DT427, DF427
1968	005730	070616				
1969					; ITEM 430	
1970	005732	066462	070221	071162	.WORD	EM430, DM430, DT430, DF430
1971	005740	070616				
1972					; ITEM 431	
1973	005742	066513	067705	071162	.WORD	EM431, DM431, DT431, DF431
1974	005750	070616				
1975					; ITEM 432	
1976	005752	066566	067555	071112	.WORD	EM432, DM432, DT432, DF432
1977	005760	070606				
1978					; ITEM 433	
1979	005762	066613	067375	071112	.WORD	EM433, DM433, DT433, DF433
1980	005770	070606				
1981					; ITEM 434	
1982	005772	066637	067705	071162	.WORD	EM434, DM434, DT434, DF434
1983	006000	070616				
1984					; ITEM 435	
1985	006002	066773	070221	071162	.WORD	EM435, DM435, DT435, DF435
1986	006010	070616				
1987					; ITEM 436	
1988	006012	067025	067705	071162	.WORD	EM436, DM436, DT436, DF436
1989	006020	070616				
1990					; ITEM 437	
1991	006022	067102	067555	071112	.WORD	EM437, DM437, DT437, DF437
1992	006030	070606				
1993					; ITEM 440	
1994	006032	067130	067555	071112	.WORD	EM440, DM440, DT440, DF440
1995	006040	070606				
1996					; ITEM 441	
1997	006042	067201	070351	071652	.WORD	EM441, DM441, DT441, DF441
1998	006050	071017				
1999					; ITEM 442	
2000	006052	067235	070417	071670	.WORD	EM442, DM442, DT442, DF442
2001	006060	071017				
2002					; ITEM 443	
2003	006062	067267	070417	071670	.WORD	EM443, DM443, DT443, DF443
2004	006070	071017				
2005						
2006						
2007					.SBTTL	ACT11 HOOKS
2008						
2009					;;*****	
2010					;HOOKS REQUIRED BY ACT11	
2011	006072				SSVPC=.	;SAVE PC
2012	000046				.=46	

M03

MAINDEC-11-FPP34-A
DFFPCA.P11 31-OCT-76

PDP 11/34 FPP DIAGNOSTIC
17:16 ACT11 HOOKS

MACY11 27(1006) 31-OCT-76 17:35 PAGE 38

```

2013 000046 037354 SENDAD ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SECP
2014 000052 030052 .=52
2015 000052 000000 .WORD 0 ;;2)SET LOC.52 TO ZERO
2016 006072 006072 .=$$VPC ;; RESTORE PC
2017 .SBTTL APT PARAMETER BLOCK
2018
2019 ;;*****
2020 ;;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
2021 ;;*****
2022 006072 .SX= ;;SAVE CURRENT LOCATION
2023 000024 .=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
2024 000024 200 ;;FOR APT START UP
2025 000044 .=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
2026 000044 $APTHDR ;;POINT TO APT HEADER BLOCK
2027 006072 .=.SX ;;RESET LOCATION COUNTER
2028
2029 ;;*****
2030 ;;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
2031 ;;INTERFACE SPEC.
2032
2033 $APTHD:
2034 $SHIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
2035 $MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
2036 $STMT: .WORD 10 ;;RUN TIM OF LONGEST TEST
2037 $PASTM: .WORD 40 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
2038 $UNITM: .WORD 0 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
2039 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
2040
2041 006106
2042
2043 START:
2044 .SBTTL INITIALIZE THE COMMON TAGS
2045 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2046 MOV $CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
2047 CLR (R6)+ ;;CLEAR MEMORY LOCATION
2048 CMP $SWR,R6 ;;DONE?
2049 BNE -6 ;;LOOP BACK IF NO
2050 MOV $STACK,SP ;;SETUP THE STACK POINTER
2051 ;;INITIALIZE A FEW VECTORS
2052 MOV $$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2053 MOV #340,@#IOTVEC+2 ;;LEVEL 7
2054 MOV $ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2055 MOV #340,@#EMTVEC+2 ;;LEVEL 7
2056 MOV $STRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2057 MOV #340,@#TRAPVEC+2;LEVEL 7
2058 MOV $PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2059 MOV #340,@#PWRVEC+2 ;;LEVEL 7
2060 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
2061 CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
2062 CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2063 MOVB #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
2064 ;;INITIALIZE THE "T-BIT" TRAP VECTOR. THEN LOAD LOCATION "$RTRN", IN
2065 ;;THE "END-OF-PASS" ($EOP) ROUTINE, WITH A "RTI" OR "RTT".
2066 MOV $RTRN,@#TBITVEC ;;SET "T" BIT VECTOR TO $RTRN
2067 MOV #340,@#TBITVEC+2 ;;LEVEL 7
2068 MOV $RTI,$RTRN ;;SET $RTRN TO A RTI
2069 MOV #655,@#RESVEC ;;TRY TO DO A RTT
2070 CLR -(SP) ;;DUMMY PS

```

```

2069 006264 012746 006272      MOV      #64$, -(SP)      ;; AND PC
2070 006270 000006      RTT      ;; TRY THE RTT
2071 006272 012767 000006 031120 64$:  MOV      #RTT, $RTRN    ;; RTT IS LEGAL--SET $RTRN TO A RTT
2072 006300 000402      BR       66$
2073 006302 062706 000010      65$:  ADD      #10, SP      ;; RTT ILLEGAL--CLEAN OFF THE STACK
2074 006306 012737 000012 000010 66$:  MOV      #RESVEC+2, @#RESVEC ;; RESTORE TRAP CATCHER
2075 006314 005067 031106      CLR      $TBIT          ;; CLEAR "T" BIT SWITCH
2076 006320 012767 006320 172560      MOV      #., $LPADR     ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
2077 006326 012767 006326 172554      MOV      #., $LPERR     ;; SETUP THE ERROR LOOP ADDRESS
2078      ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2079      ;; EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
2080 006334 013746 000004      MOV      @#ERRVEC, -(SP) ;; SAVE ERROR VECTOR
2081 006340 012737 006374 000004      MOV      #67$, @#ERRVEC ;; SET UP ERROR VECTOR
2082 006346 012767 177570 172564      MOV      #DSWR, SWR     ;; SETUP FOR A HARDWARE SWITCH REGISTER
2083 006354 012767 177570 172560      MOV      #DDISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
2084 006362 022777 177777 172550      CMP      #-1, @SWR     ;; TRY TO REFERENCE HARDWARE SWR
2085 006370 001012      BNE      69$           ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
2086      ;; AND THE HARDWARE SWR IS NOT = -1
2087 006372 000403      BR       68$           ;; BRANCH IF NO TIMEOUT
2088 006374 012716 006402      67$:  MOV      #68$, (SP)    ;; SET UP FOR TRAP RETURN
2089 006400 000002      RTI
2090 006402 012767 000176 172530 68$:  MOV      #SWREG, SWR    ;; POINT TO SOFTWARE SWR
2091 006410 012767 000174 172524      MOV      #DISPREG, DISPLAY
2092 006416 012637 000004      69$:  MOV      (SP)+, @#ERRVEC ;; RESTORE ERROR VECTOR
2093
2094 006422 005067 172676      CLR      $PASS         ;; CLEAR PASS COUNT
2095 006426 132767 000200 172703      BITB    #APTSIZE, $ENVM ;; TEST USER SIZE UNDER APT
2096 006434 001403      BEQ      70$           ;; YES, USE NON-APT SWITCH
2097 006436 012767 001340 172474      MOV      #SSWREG, SWR  ;; NO, USE APT SWITCH REGISTER
2098 006444
2099      70$:
2100      .SBTTL  TYPE PROGRAM NAME
2101      ;; TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2102 006444 005227 177777      INC      #-1           ;; FIRST TIME?
2103 006450 001052      BNE      71$           ;; BRANCH IF NO
2104 006452 022737 037354 000042      CMP      #SENDAD, @#42 ;; ACT-11?
2105 006460 001446      BEQ      71$           ;; BRANCH IF YES
2106 006462 104401 006530      TYPE    72$           ;; TYPE ASCIZ STRING
2107      .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2108 006466 005737 000042      TST     @#42          ;; ARE WE RUNNING UNDER XXDP/ACT?
2109 006472 001012      BNE      73$           ;; BRANCH IF YES
2110 006474 126727 172636 000001      CMPB    $ENV, #1     ;; ARE WE RUNNING UNDER APT?
2111 006502 001406      BEQ     73$           ;; BRANCH IF YES
2112 006504 026727 172430 000176      CMP     SWR, #SWREG  ;; SOFTWARE SWITCH REG SELECTED?
2113 006512 001005      BNE     74$           ;; BRANCH IF NO
2114 006514 104405      GTSWR   ;; GET SOFT-SWR SETTINGS
2115 006516 000403      BR     74$
2116 006520 112767 000001 172406 73$:  MOVB    #1, $AUTOB    ;; SET AUTO-MODE INDICATOR
2117 006526 000423      74$:  BR     71$
2118      ;; 72$: .ASCIZ <CRLF>*FP11A, 11/34 FPP, DIAGNOSTIC PART 3*<CRLF>
2119 006576      71$:
2120
2121 006576      LOOP:
2122
2123
2124

```

2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180

```

*****
*TEST 1      STF WITH ILLEGAL ACCUMULATOR TEST
*
*THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE C.
*
*****
ST1:  SCOPE
0001:
      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      CLR           RC          ;SET THE FPS.
      LDFPS         RO
0002: 000004
      MOV           #000T,2#FPVECT ;SET UP FOR FP TRAPS.
      MOV           #15,2#STMP2
15:   STF           ACC,AC7      ;THIS TEST INSTRUCTION SHOULD
                                   ;CAUSE A TRAP.
      ;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
0002:
      STFPS        RO          ;GET FPS.
      MOV          RO,2#STMP3   ;GET FEC.
      STST         RC
      MOV          RO,2#STMP4
35:   ERROR        1          ;STF WITH ILLEGAL ACCUMULATOR, MODE
                                   ;0, DIDN'T TRAP. ST 765 TO ST 537.
      BR           000DONE
      ;TRAP TO 000T, HERE, WHEN THE EXPECTED ERROR OCCURS.
000T: MOV           (SP),RO     ;MAKE SURE THE ERROR OCCURRED
      CMP          #0002,RO     ;AT THE CORRECT ADDRESS.
      BEQ          0003        ;BRANCH IF TRAP ADDRESS CORRECT.
      JMP          2#FPSPUR     ;IF INCORRECT GO REPORT SPURIOUS
                                   ;FP TRAP.
0003: STFPS        R4          ;GET FPS.
      STST         R5          ;GET FEC.
      MOV          R4,2#STMP3   ;SAVE DATA IN CASE OF ERROR.
      MOV          R5,2#STMP4
      MOV          #100000,R2   ;EXPECTED FPS
      MOV          #2,R3        ;EXPECTED FEC
      MOV          R2,2#STMP5
      MOV          R3,2#STMP6
      CMP          (SP)+,(SP)+ ;RESET THE STACK.
      CMP          R2,R4        ;WAS FPS CORRECT?
      BEQ          0004        ;BRANCH IF YES.
      ;OTHERWISE REPORT FPS INCORRECTLY
      ;SET AFTER USE OF ILLEGAL ACC.
15:   ERROR        2
      BR           000DONE
0004: CMP          R3,R5        ;WAS THE FEC CORRECT?
      BEQ          000DONE     ;BRANCH IF CORRECT.
006576 000004
006600
00660C 104413
006602 005000
006604 17C100
006606 012737 006644 000244
006614 012737 006622 001236
006622 174007
006624
006624 170200
006626 010037 001240
006632 170300
006634 010037 001242
006640 104001
006642 000434
006644 011600
006646 022700 006624
006652 001402
006654 000137 042564
006660 170204
006662 170305
006664 010437 001240
006670 010537 001242
006674 012702 100000
006700 012703 000002
006704 010237 001244
006710 010337 001246
006714 022626
006716 020204
006720 001402
006722 104002
006724 000403
006726 020305
006730 001401

```


C04

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
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236

006732 104003
006734 104412
006736 000004
006740 104413
006742 012700 177777
006746 012701 007076
006752 012702 000014
006756 010021
006760 077202
006762 012700 000200
006766 170100
006770 012700 007126
006774 172410
006776 012700 007112
007002 005002
007004 170102
007006 012737 007020 001236
007014 010037 001240
007020 174010
007022 022700 007112
007026 001404
007030 010037 001242
007034 104004
007036 000456
007040 012700 007112
007044 012701 007126
007050 022021
007052 001031
007054 022011
007056 001027

15: ERROR 3 ; OTHERWISE REPORT INCORRECT FEC
; AFTER USE OF ILLEGAL ACC.
000DONE: RSETUP ; GO INITIALIZE THE FPS AND STACK; AND
; SEE IF THE USER HAS EXPRESSED
; THE DESIRE TO CHANGE THE SOFTWARE
; VIRTUAL CONSOLE SWITCH REGISTER (HAS
; THE USER TYPED CONTROL G?).
:*****
: *TEST 2 FDST MODE 1, FLOATING MODE, TEST
: *
: *THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.
: *
:*****
↑ST2: SCOPE
PPP1: LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
MOV #1, R0 ; SET UP A BACKGROUND PATTERN IN THE
MOV #PPPBF0, R1 ; INPUT BUFFER.
MOV #14, R2
PPP2: MOV R0, (R1)+
SOB R2, PPP2
MOV #200, R0 ; SET FD MODE.
LDFPS R0
MOV #PPPTP1, R0 ; PUT TEST DATA INTO ACC.
LDD (R0), ACC
MOV #PPPBF1, R0 ; FDST ADDRESS.
CLR R2 ; CLEAR THE FPS.
LDFPS R2
MOV #PPP3, 2#STMP2
MOV R0, 2#STMP3
PPP3: STF ACC, (R0) ; TEST INSTRUCTION.
CMP #PPPBF1, R0 ; WAS R0 MODIFIED DURING EXECUTION?
BEQ PPP4 ; BRANCH IF R0 NOT MODIFIED, CORRECT.
MOV R0, 2#STMP4 ; OTHERWISE REPORT ERROR, R0 MODIFIED.
15: ERROR 4 ; GO TO NEXT TEST.
BR PPPDONE
PPP4: MOV #PPPBF1, R0 ; CHECK THE DATA IN THE OUTPUT BUFFER.
MOV #PPPTP1, R1
CMP (R0)+, (R1)+ ; BRANCH IF INCORRECT.
BNE PPP10
CMP (R0)+, (R1) ; BRANCH IF INCORRECT..
BNE PPP10

```

2237 007060 022720 177777      CMP      #-1,(R0)+      ;WAS FLOATING MODE USED?
2238 007064 001034      BNE      PPP15      ;BRANCH IF NOT.
2239 007066 022710 177777      CMP      #-1,(R0)
2240 007072 001031      BNE      PPP15
2241 007074 000437      BR       PPPDONE ;GO TO NEXT TEST.
2242
2243 007076 177777 177777 177777 PPPBFD: .WORD  -1,-1,-1,-1,-1,-1
2244 007104 177777 177777 177777
2245
2246 007112 177777 177777 177777 PPPBF1: .WORD  -1,-1,-1,-1,-1,-1
2247 007120 177777 177777 177777
2248
2249 007126 123456 023456      PPPTP1: .WORD  123456,23456
2250 007132 034567 045671      .WORD  34567,45671
2251
2252      ;REPORT DATA IN OUT PUT BUFFER INCORRECT.
2253 007136 012737 007126 001242 PPP10: MOV      #PPPTP1,2#STMP4
2254 007144 012737 007112 001240      MOV      #PPPBFD,2#STMP3
2255 007152 104005      IS:      ERROR      5      ;BAD DATA.
2256 007154 000407      BR       PPPDONE
2257
2258      ;REPORT FLOATING MODE NOT USED, BUT FD FAILED.
2259 007156 012737 007126 001242 PPP15: MOV      #PPPTP1,2#STMP4
2260 007164 012737 007112 001240      MOV      #PPPBFD,2#STMP3
2261 007172 104006      IS:      ERROR      6      ;ST 707 TO 245 INTO 244 (BUT FD).
2262
2263 007174      PPPDONE:
2264 007174 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
2265      ;SEE IF THE USER HAS EXPRESSED
2266      ;THE DESIRE TO CHANGE THE SOFTWARE
2267      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2268      ;THE USER TYPED CONTROL G?).
2269
2270
2271
2272
2273      ;*****
2274      ;*TEST 3      FDST MODE 2 TEST
2275      ;*
2276      ;*THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.
2277      ;*
2278      ;*****
2279 007176 000004      ST3:      SCOPE
2280
2281      ;FIRST TEST STF.
2282 007200      Q001:
2283 007200 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2284
2285 007202 012700 177777      MOV      #-1,R0      ;SET UP THE OUTPUT BUFFER.
2286 007206 012701 007340      MOV      #QQQBF0,R1
2287 007212 012702 000014      MOV      #14,R2
2288 007216 010021      QQQ2:      MOV      R0,(R1)+
2289 007220 077202      SOB      R2,QQQ2
2290
2291 007222 012700 000200      MOV      #200,R0      ;SET FD MODE.
2292 007226 170100      LDFPS    R0

```

E04

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 43
 OFFPCA.P11 31-OCT-76 17:16 T3 FDST MODE 2 TEST

```

2293 007230 012700 007370      MOV      #QQQTP1,RO      ;SETUP ACO.
2294 007234 172410      LDD      (RO),ACO
2295
2296 007236 012700 007354      MOV      #QQQBF1,RO      ;FDST ADDRESS.
2297 007242 005002      CLR      R2
2298 007244 170102      LDFPS   R2              ;SET FPS.
2299 007246 012737 007254 001236      MOV      #QQQ3,2#STMP2
2300
2301 007254 174020      QQQ3:   STF      ACO,(RO)+      ;TEST INSTRUCTION.
2302
2303 007256 022700 007360      CMP      #QQQBF1+4,RO      ;WAS RO INCREMENTED BY 4 PROPERLY?
2304
2305 007262 001407      BEQ      QQQ4              ;BRANCH IF RO CORRECT.
2306 007264 019037 001242      MOV      RO,2#STMP4      ;REPORT RO INCORRECT AFTER FDST MODE 2.
2307 007270 012737 007360 001240      MOV      #QQQBF1+4,2#STMP3
2308 007276 104007      15:     ERROR      7              ;BAD CONSTANT USED OR DIDN'T GO 527 TO 642
2309 007300 000526      BR
2310 007302 012700 007354      QQQ4:   MOV      #QQQBF1,RO      ;WAS THE OUTPUT DATA CORRECT?
2311 007306 012701 007370      MOV      #QQQTP1,R1
2312 007312 022021      CMP      (RO)+,(R1)+
2313 007314 001031      BNE      QQQ10            ;BRANCH IF INCORRECT.
2314 007316 022021      CMP      (RO)+,(R1)+
2315 007320 001027      BNE      QQQ10            ;BRANCH IF INCORRECT.
2316 007322 022027 177777      CMP      (RO)+,#-1        ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
2317 007326 001024      BNE      QQQ10            ;BRANCH IF INCORRECT.
2318 007330 022027 177777      CMP      (RO)+,#-1
2319 007334 001021      BNE      QQQ10            ;BRANCH IF INCORRECT.
2320 007336 000430      BR      QQQ20
2321 007340 177777 177777 177777      QQQBF0: .WORD      -1,-1,-1,-1,-1,-1
2322 007346 177777 177777 177777
2323 007354 177777 177777 177777      QQQBF1: .WORD      -1,-1,-1,-1,-1,-1
2324 007362 177777 177777 177777
2325 007370 076543      QQQTP1: 76543
2326 007372 065432      65432
2327 007374 054321      54321
2328 007376 043210      43210
2329      ;REPORT OUTPUT DATA INCORRECT:
2330 007400 012737 007370 001240      QQQ10:  MOV      #QQQTP1,2#STMP3
2331 007406 012737 007354 001242      MOV      #QQQBF1,2#STMP4
2332 007414 104010      15:     ERROR      10              ;BAD DATA
2333 007416 000457      BR      QQQDONE
2334
2335      ;NOW TEST STD MODE 2.
2336
2337      QQQ20:
2338 007420      LPERR
2339 007422 104413      MOV      #QQQBF0,RO      ;SET UP THE LOOP ON ERROR ADDRESS.
2340 007426 010001 007340      MOV      RO,R1           ;SET UP DEFAULT INPUT DATA BUFFER.
2341 007430 012702 000014      MOV      #14,R2
2342 007434 010021      QQQ22:  MOV      RO,(R1)+
2343 007436 077202      SOB     R2,QQQ22
2344 007440 012700 000200      MOV      #200,RO        ;ENTER FLOATING DOUBLE MODE.
2345 007444 170100      LDFPS   RO
2346 007446 012700 007370      MOV      #QQQTP1,RO      ;LOAD ACO.
2347 007452 172410      LDD      (RO),ACO
2348 007454 012700 007354      MOV      #QQQBF1,RO      ;SET DESTINATION ADDRESS.

```

F04

```

2349 007460 012737 007466 001236      MOV      #QQQ23,2#STMP2
2350 007466 174020      QQQ23:  STD      ACO,(R0)+      ;TEST INSTRUCTION.
2351 007470 022700 007364      CMP      #QQQBF1+10,RC  ;WAS R0 INCREMENTED BY 10 CORRECTLY?
2352 007474 001407      BEQ      QQQ24          ;BRANCH IF CORRECT.
2353 007476 010037 001242      MOV      R0,2#STMP4     ;REPORT R0 INCORRECTLY INCREMENTED.
2354 007502 012737 007364 001240      MOV      #QQQBF1+10,2#STMP3
2355 007510 104011      1$:     ERROR      11      ;DO NOT INCREM BY 10 BAD CONSTANT
2356 007512 000421      BR       QQQDONE
2357 007514 012700 007354      QQQ24:  MOV      #QQQBF1,R0      ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
2358 007520 012701 007370      MOV      #QQQTP1,R1
2359 007524 012702 000004      MOV      #4,R2
2360 007530 022021      1$:     CMP      (R0)+,(R1)+
2361 007532 001002      BNE      QQQ25          ;BRANCH IF INCORRECT.
2362 007534 077203      SOB      R2,1$
2363 007536 000407      BR       QQQDONE
2364                                     ;REPORT DATA INCORRECT.
2365 007540 012737 007370 001240      QQQ25:  MOV      #QQQTP1,2#STMP3
2366 007546 012737 007354 001242      MOV      #QQQBF1,2#STMP4
2367 007554 104012      1$:     ERROR      12      ;BAD DATA
2368 007556
2369 007556 104412      QQQDONE: RSETUP        ;GO INITIALIZE THE FPS AND STACK; AND
2370                                     ;SEE IF THE USER HAS EXPRESSED
2371                                     ;THE DESIRE TO CHANGE THE SOFTWARE
2372                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2373                                     ;THE USER TYPED CONTROL G?).
2374
2375 ;*****
2376 ;*TEST 4          FDS* MODE 2, WITH GR7, TEST
2377 ;*
2378 ;*THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.
2379 ;*
2380 ;*****
2381 007560 000004      1ST4:   SCOPE
2382
2383 RRR1:
2384 007562 104413      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
2385 007564 012700 007642      MOV      #RRR3,R0      ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
2386 007570 012701 007710      MOV      #RRRTP1,R1
2387 007574 012702 000004      MOV      #4,R2
2388 007600 012021      1$:     MOV      (R0)+,(R1)+
2389 007602 077202      SOB      R2,1$
2390 007604 012700 000200      MOV      #200,R0       ;ENTER FLOATING DOUBLE MODE.
2391 007610 170100      LDFPS      R0
2392 007612 012700 007720      MOV      #RRRTP2,R0    ;SET UP ACO.
2393 007616 172410      LDD      (R0),ACO
2394 007620 012737 007740 000004      MOV      #RRR10,2#ERRVECT ;SET UP FOR AN ODD ADDRESS.
2395 007626 012737 007640 001236      MOV      #RRR2,2#STMP2
2396 007634 005001      CLR      R1
2397 007636 005004      CLR      R4
2398 ;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
2399 ;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
2400 ;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
2401 ;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
2402 ;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
2403 ;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
2404 007640 174027      RRR2:  STD      ACO,(R7)+      ;TEST INSTRUCTION.

```

G04

```

2405 007642 005201 RRR3: INC R1 ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4.
2406 007644 005201 INC R1
2407 007646 005201 INC R1
2408 007650 005201 INC R1
2409 007652 012700 007730 MOV #RRREXP,R0 ;SEE IF THE DATA WAS OUTPLT CORRECTLY.
2410 007656 012702 007642 MOV #RRR3,R2
2411 007662 012703 000004 MOV #4,R3
2412 007666 022022 RRR4: CMP (R0)+,(R2)+ ;BRANCH IF INCORRECT.
2413 007670 001051 BNE RRR25
2414 007672 077303 SOB R3,RRR4
2415 007674 005704 TST R4 ;MAKE SURE R4 IS 0.
2416 007676 001056 BNE RRR15 ;BRANCH IF R4 IS INCORRECT.
2417 007700 022701 000003 CMP #3,R1 ;SEE IF R1 IS CORRECT.
2418 007704 001053 BNE RRR15 ;BRANCH IF R1 IS INCORRECT.
2419 007706 000474 BR RRRDONE
2420 ;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3.
2421 007710 005201 RRRTP1: INC R1
2422 007712 005201 INC R1
2423 007714 005201 INC R1
2424 007716 005201 INC R1
2425 ;THIS IS THE DATA PUT IN ACO BEFORE EXECUTION OF THE STD.
2426 007720 005204 RRRTP2: INC R4
2427 007722 005204 INC R4
2428 007724 005204 INC R4
2429 007726 005204 INC R4
2430 ;THIS IS THE EXPECTED DATA AT RRR3 AFTER EXECUTION OF THE STD.
2431 007730 005204 RRREXP: INC R4
2432 007732 005201 INC R1
2433 007734 005201 INC R1
2434 007736 005201 INC R1
2435 ;IF A FAILURE IN THE FDST FLOWS RESULTS IN AN ODD ADDRESS TRAP THROUGH
2436 ;4 TO HERE:
2437 007740 011602 RRR10: MOV (SP),R2 ;SEE IF THE TRAP WAS BECAUSE OF AN ODD ADDRESS.
2438 007742 032702 000001 BIT #1,R2
2439 007746 001005 BNE RRR11 ;BRANCH IF YES.
2440 007750 020227 007644 CMP R2,#RRR3+2 ;SEE IF THE TRAP OCCURRED AT THE TEST INSTRUCTION.
2441 007754 001412 BEQ RRR12 ;BRANCH IF YES.
2442 007756 000137 042620 JMP @CPSPUR ;OTHERWISE REPORT A SPURIOUS TRAP THROUGH VECTOR 4.
2443 ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
2444 007762 010237 001236 RRR11: MOV R2,@STMP2
2445 007766 012737 007644 001240 MOV #RRR3+2,@STMP3
2446 007774 022626 CMP (SP)+,(SP)+
2447 007776 104013 1S: ERROR 13 ;BAD CONSTANT #2 + PC ODD ADDR.
2448 010000 000437 BR RRRDONE
2449 010002 010237 001236 RRR12: MOV R2,@STMP2
2450 010006 022626 CMP (SP)+,(SP)+
2451 010010 104014 1S: ERROR 14 ;ODD ADDRESS TRAP
2452 010012 000432 BR RRRDONE ;WRONG MODE USED.
2453
2454 ;REPORT DATA INCORRECT:
2455 010014 012737 007642 001240 RRR25: MOV #RRR3,@STMP3
2456 010022 012737 007730 001242 MOV #RRREXP,@STMP4
2457 010030 104015 1S: ERROR 15 ;BAD DATA BUT GR7 FAIL
2458 010032 000422 BR RRRDONE
2459
2460 ;REPORT PC INCORRECT MODIFIED DURING THE EXECUTION OF FDST IMMEDIATE

```


H04

2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516

010034 012737 007644 001240
010042 005704
010044 001404
010046 012737 007642 001242
010054 000410
010056 012702 007644
010062 062701 177775
010066 006301
010070 160102
010072 010237 001242
010076
010076 104016
010100
010100 104412

```
;MODE. THE PC SHOULD HAVE BEEN INCREMENTED BY 2 BUT IT WASN'T.  
;USE R1 AND R4 TO COMPUTE THE ACTUAL ACTION THAT WAS TAKEN ON THE PC.  
RRR15: MOV #RRR3+2,2*STMP3  
TST R4 ;IS R4 CLEAR.  
BEQ 1$  
MOV #RRR3,2*STMP4  
BR 2$  
1$: MOV #RRR3+2,R2  
ADD #-3,R1  
ASL R1  
SUB R1,R2  
MOV R2,2*STMP4  
2$:  
3$: ERROR 16 ;BAD CONSTANT PC+  
RRRDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HPS  
;THE USER TYPED CONTROL G?).  
:*****  
:TEST 5 FDST MODE 4 TEST  
:*****  
:THIS IS A TEST OF STD WITH FDST MODE 4.  
:*****  
TST5: SCOPE  
SSS1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #-1,RO ;SET UP THE OUTPUT BUFFER.  
MOV #SSSBFO,R1  
MOV #10,R2  
1$: MOV RO,(R1)+  
SOB R2,1$  
MOV #200,RO ;ENTER FLOATING DOUBLE MODE.  
LDFPS RO  
MOV #SSSTP1,RO ;SET UP ACO.  
LDD (RO),ACO  
MOV #SSS10,2*ERRVECT ;SET UP FOR A TRAP TO 4.  
MOV #SSS2,2*STMP2  
MOV #SSSA1,RO ;SET UP THE DESTINATION ADDRESS.  
SSS2: STD ACO,-(RO) ;TEST INSTRUCTION.  
INC R1  
CMP RO,#SSSBFO ;SEE IF RO WAS DECREMENTED PROPERLY.  
BNE SSS15 ;BRANCH IF RO IS INCORRECT.  
MOV #SSSBFO,RO ;WAS THE OUTPUT DATA CORRECT?  
MOV #SSSTP1,R1  
1$: MOV #4,R2  
CMP (RO)+,(R1)+ ;BRANCH IF INCORRECT.  
BNE SSS20  
SOB R2,1$  
MOV #-1,RO ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?  
MOV #SSSA1,R1
```

2517 010226 012702 000004
2518 010232 020021
2519 010234 001056
2520 010236 077203
2521 010240 000463
2522
2523
2524 010242 177777
2525 010244 177777
2526 010246 177777
2527 010250 177777
2528 010252 177777
2529 010254 177777
2530 010256 177777
2531 010260 177777
2532
2533
2534 010262 147250
2535 010264 036147
2536 010266 025036
2537 010270 147250
2538 010272 177777
2539 010274 177777
2540 010276 177777
2541 010300 177777
2542
2543
2544 010302 011600
2545 010304 020027 010164
2546 010310 001405
2547 010312 020027 010166
2548 010316 001402
2549 010320 000137 042620
2550
2551 010324 010037 001236
2552 010330 104017
2553 010332 000426
2554
2555
2556 010334 010037 001242
2557 010340 012737 010242 001240
2558 010346 104020
2559 010350 000417
2560
2561
2562 010352 012737 010242 001240
2563 010360 012737 010262 001242
2564 010366 104021
2565 010370 000407
2566 010372 012737 010252 001242
2567 010400 012737 010272 001240
2568 010406 104022
2569 010410
2570 010410 104412
2571
2572

MOV #4,R2
25: CMP RO,(R1)+
BNE SSS25 ;BRANCH IF INCORRECT.
SOB R2,25
BR SSSDONE

;THIS IS THE OUTPUT DATA BUFFER.
SSSBFO: -1
-1
-1
-1
SSSA1: -1
-1
-1
-1

;THIS IS THE TEST DATA LOADED INTO ACO:
SSSTP1: 147250
36147
25036
147250
SSSTP2: -1
-1
-1
-1

;IF AN ODD ADDRESS TRAP OCCURS COME HERE:
SSS10: MOV (SP),RO ;SEE IF THE TRAP ACCURRED ON THE TEST INSTRUCTION.
CMP RO,#SSS2+2
BEQ SSS11 ;BRANCH IF YES.
CMP RO,#SSS2+4
BEQ SSS11 ;BRANCH IF YES.
JMP @CPSPUR ;OTHERWISE GO REPORT A SPURIOUS TRAP THROUGH 4.
;REPORT FAILURE IN FDST FLOWS RESULTED IN AN ODD ADDRESS.
SSS11: MOV RO,@STMP2
25: ERROR 17 ;FDST FORK X ODD AD RES.
BR SSSDONE

;REPORT RO INCORRECTLY DECREMENTED.
SSS15: MOV RO,@STMP4
MOV #SSSBFO,@STMP3
15: ERROR 20 ;RO NOT DECRE PROP
BR SSSDONE

;REPORT OUTPUT DATA INCORRECT:
SSS20: MOV #SSSBFO,@STMP3
MOV #SSSTP1,@STMP4
15: ERROR 21 ;BAD DATA
BR SSSDONE
SSS25: MOV #SSSA1,@STMP4
MOV #SSSTP2,@STMP3
15: ERROR 22 ;DATA BAD OUTSIDE TARGET AREA
SSSDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE

;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

2573
2574
2575
2576
2577
2578
2579
2580
2581
2582 0:0412 000004
2583
2584 010414
2585 010414 104413
2586 010416 012701 010534
2587 010422 012700 177777
2588 010426 012702 000013
2589 010432 010021
2590 010434 077202
2591 010436 012737 010534 010550
2592 010444 012700 000200
2593 010450 170100
2594 010452 012700 010560
2595 010456 172410
2596 010460 012737 010570 000004
2597 010466 016737 000006 001236
2598 010474 012700 010550
2599
2600 010500 174030
2601
2602 010502 020027 010552
2603 010506 001046
2604 010510 012701 010534
2605 010514 012702 010560
2606 010520 012703 000004
2607 010524 022122
2608 010526 001045
2609 010530 077303
2610 010532 000452
2611
2612
2613 010534 177777
2614 010536 177777
2615 010540 177777
2616 010542 177777
2617 010544 177777
2618 010546 177777
2619 010550 010534
2620 010552 177777
2621 010554 177777
2622 010556 177777
2623 010560 101213
2624 010562 141516
2625 010564 071727
2626 010566 037475
2627
2628

;TEST 6 FDST MODE 3 TEST
;THIS IS A TEST OF FDST MODE 3 USING STD.

TTT6: SCOPE
TTT1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #TTTBFO,R1 ;SET UP THE OUTPUT DATA BUFFER.
MOV #-1,R0
MOV #13,R2
15: MOV RO,(R1)+
SOB R2,15
MOV #TTTBFO,2#TTTA2
MOV #200,RO ;ENTER DOUBLE FLOATING MODE.
LDFPS RO
MOV #TTTTP1,RO ;SET UP ACO.
LDD (RO),ACO
MOV #TTT10,2#ERRVECT ;SET UP FOR TRAPS TO 4.
MOV TTT2,2#STMP2
MOV #TTTA2,RO ;SET UP THE DESTINATION ADDRESS.
TTT2: STD ACO,2(RO)+ ;TEST INSTRUCTION.
CMP RO,#TTTA2+2 ;SEE IF RO WAS INCREMENTED CORRECTLY.
BNE TTT15 ;BRANCH IF INCORRECT.
MOV #TTTBFO,R1 ;CHECK THE OUTPUT DATA BUFFER.
MOV #TTTTP1,R2
TTT3: MOV #4,R3
CMP (R1)+,(R2)+
BNE TTT20 ;BRANCH IF NOT CORRECT.
SOB R3,TTT3
BR TTTDONE

;THIS IS THE OUTPUT DATA BUFFER:

TTTBFO: -1
-1
-1
-1
-1
-1
TTTA1: -1
TTTA2: TTTBFO
TTTA3: -1
-1
-1
TTTTP1: 101213
141516
071727
037475

;TRAP THROUGH VECTOR 4 TO HERE.

K04

```

2629 010570 011602          TTT10: MOV      (SP),R2          ;SEE IF THE TRAP ADDRESS IS THAT OF THE TEST INSTRUCTION
2630 010572 020227 010502    CMP      R2,#TTT2+2          ;
2631 010576 001405          BEQ      TTT11                ;BRANCH IF YES.
2632 010600 020227 010504    CMP      R2,#TTT2+4          ;
2633 010604 001402          BEQ      TTT11                ;BRANCH IF YES.
2634 010606 000137 042620    JMP      @#CPSPUR            ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
2635
2636          ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
2637 010612 010237 001236    TTT11: MOV      R2,@#STMP2
2638 010616 022626          CMP      (SP)+,(SP)+
2639 010620 104023          1S:      ERROR   23          ;BET FDST X ODD ADR
2640 010622 000416          BR       TTTDONE
2641
2642          ;REPORT RO INCORRECT:
2643 010624 010037 001242    TTT15: MOV      RO,@#STMP4
2644 010630 012737 010552 001240  MOV      @TTTA2+2,@#STMP3
2645 010636 104024          1S:      ERROR   24          ;RO NOT INCREMENT PROPERLY
2646 010640 000407          BR       TTTDONE
2647
2648          ;REPORT INCORRECT OUTPUT DATA:
2649 010642 012737 010534 001240  TTT20: MOV      @TTTBFO,@#STMP3
2650 010650 012737 010560 001242  MOV      @TTTTP1,@#STMP4
2651 010656 104025          1S:      ERROR   25          ;BAD DATA
2652 010660          TTTDONE:
2653 010660 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
2654          ;SEE IF THE USER HAS EXPRESSED
2655          ;THE DESIRE TO CHANGE THE SOFTWARE
2656          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2657          ;THE USER TYPED CONTROL G?).
2658
2659          ;*****
2660          ;*TEST 7          FDST MODE 5 TEST
2661          ;*
2662          ;*THIS IS A TEST OF FDST MODE 5 USING STD.
2663          ;*
2664          ;*****
2665 010662 000004          TST7:  SCOPE
2666
2667 010664          UUU1:
2668 010664 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
2669 010666 012701 011004    MOV      @UUUBFO,R1          ;SET JP THE OUTPUT DATA BUFFER.
2670 010672 012700 177777    MOV      #-1,R0
2671 010676 012702 000013    MOV      #13,R2
2672 010702 010021          1S:      MOV      RO,(R1)+
2673 010704 077202          SOB      R2,1S
2674 010706 012737 011004 011016  MOV      @UUUBFO,@#UUUA1
2675 010714 012700 000200          MOV      #200,R0          ;ENTER DOUBLE FLOATING MODE.
2676 010720 170100          LDFPS      RO
2677 010722 012700 011030    MOV      @UUUTP1,R0          ;SET UP ACO.
2678 010726 172410          LDD      (RO),ACO
2679 010730 012737 011040 000004  MOV      @UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
2680 010736 016737 000006 001236  MOV      UUU2,@#STMP2
2681 010744 012700 011020          MOV      @UUUA2,R0          ;SET UP THE DESTINATION ADDRESS.
2682 010750 174050          UUU2:  STD      ACO,@-(RO)      ;TEST INSTRUCTION.
2683 010752 020027 011016    CMP      RO,@UUUA2-2        ;WAS RO DECRIMENTED PROPERLY?
2684 010756 001046          BNE      UUU15              ;BRANCH IF RO IS INCORRECT.

```

```

2685 010760 012701 011004      MOV      #UUUBFO,R1      ;WAS THE DATA OUTPUT CORRECTLY?
2686 010764 012702 011030      MOV      #UUUTP1,R2
2687 010770 012703 000004      MOV      #4,R3
2688 010774 022122      UUU3:   CMP      (R1)+,(R2)+
2689 010776 001045      BNE     UUU20           ;BRANCH IF DATA IS INCORRECT.
2690 011000 077303      SOB     R3,UUU3
2691 011002 000452      BR      UUUDONE
2692
2693      ;THIS IS THE OUTPUT DATA BUFFER
2694 011004 177777      UUUBFO: -1
2695 011006 177777      -1
2696 011010 177777      -1
2697 011012 177777      -1
2698 011014 177777      -1
2699 011016 011004      UUA1:  UUUBFO
2700 011020 177777      UUA2:  -1
2701 011022 177777      UUA3:  -1
2702 011024 177777      -1
2703 011026 177777      -1
2704 011030 020212      UUUTP1: 20212
2705 011032 023242      23242
2706 011034 026273      26273
2707 011036 031323      031323
2708
2709      ;IF A TRAP TO 4 OCCURS COME HERE.
2710 011040 011602      UUU10: MOV      (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2711 011042 020227 010752      CMP      R2,#UUU2+2
2712 011046 001405      BEQ     UUU11           ;BRANCH IF YES.
2713 011050 020227 010754      CMP      R2,#UUU2+4
2714 011054 001402      BEQ     UUU11           ;BRANCH IF YES.
2715 011056 000137 042620      JMP     @#CPSPUR       ;OTHERWISE REPORT A SPURIOUS TRAP TO 4.
2716      ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
2717 011062 010237 001236      UUU11: MOV      R2,@#STMP2
2718 011066 022626      CMP      (SP)+,(SP)+
2719 011070 104026      IS:    ERROR 26         ;BET FDST X ODD ADR
2720 011072 000416      BR      UUUDONE
2721
2722      ;REPORT RD INCORRECT.
2723 011074 010037 001242      UUU15: MOV      RD,@#STMP4
2724 011100 012737 011022 001240      MOV      #UUUA2+2,@#STMP3
2725 011106 104027      IS:    ERROR 27         ;RD NOT INCREMENT PROPERLY
2726 011110 000407      BR      UUUDONE
2727
2728      ;REPORT BAD DATA.
2729 011112 012737 011034 001242      UUU20: MOV      #UUUBFO,@#STMP4
2730 011120 012737 011030 001240      MOV      #UUUTP1,@#STMP3
2731 011126 104030      IS:    ERROR 30         ;BAD DATA
2732 011130      UUUDONE:
2733 011130 104412      RSETUP           ;GO INITIALIZE THE FPS AND STACK; AND
2734      ;SEE IF THE USER HAS EXPRESSED
2735      ;THE DESIRE TO CHANGE THE SOFTWARE
2736      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2737      ;THE USER TYPED CONTROL G?).
2738
2739      ;*****
2740      ;*TEST 10      FDST MODE 6, INDEX MODE, TEST

```


MO4

```

2741
2742
2743
2744
2745 011132 000004
2746
2747 011134
2748 011134 104413
2749 011136 012700 000200
2750 011142 170100
2751 011144 012701 011254
2752 011150 012700 177777
2753 011154 012702 000004
2754 011160 010021
2755 011162 077202
2756 011164 012737 011274 000004
2757 011172 012700 011264
2758 011176 172410
2759 011200 012737 011216 001236
2760 011206 012700 003353
2761 011212 012701 000001
2762 011216 174060 005701
2763
2764 011222 020027 003353
2765 011226 001040
2766 011230 012702 011254
2767 011234 012703 011264
2768 011240 012704 000004
2769 011244 022223
2770 011246 001037
2771 011250 077403
2772 011252 000444
2773 011254 177777
2774 011256 177777
2775 011260 177777
2776 011262 177777
2777 011264 030313
2778 011266 023334
2779 011270 035363
2780 011272 074041
2781
2782
2783 011274 011602
2784 011276 020227 011220
2785 011302 001405
2786 011304 020227 011222
2787 011310 001402
2788 011312 000137 042564
2789
2790 011316 010237 001236
2791 011322 022626
2792 011324 104031
2793 011326 000416
2794
2795
2796 011330 010037 001242

; *
; * THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.
; *
; *****
TST10: SCOPE

VVV1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #200,RO ;ENTER DOUBLE FLOATING MODE.
LDFPS RO
MOV #VVVBFO,R1 ;SET UP THE OUT PUT DATA BUFFER.
MOV #-1,RO
MOV #4,R2
IS: MOV RO,(R1)+
SOB R2,IS
MOV #VVV10,ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR.
MOV #VVVTP1,RO ;SET UP ACO.
LDD (RO),ACO
MOV #VVV2,STMP2
MOV #VVVBFO-5701,RO ;SET UP THE DESTINATION ADDRESS.
MOV #1,R1
VVV2: STD ACO,5701(RO) ;TEST INSTRUCTION.

CMP RO,#VVVBFO-5701 ;SEE IF RO WAS MODIFIED.
BNE VVV15 ;BRANCH IF INCORRECT.
MOV #VVVBFO,R2 ;WAS THE OUTPUT DATA CORRECT.
MOV #VVVTP1,R3
MOV #4,R4
IS: CMP (R2)+,(R3)+
BNE VVV20 ;BRANCH IF INCORRECT DATA.
SOB R4,IS
BR VVVDONE

VVVBFO: -1
-1
-1
-1

VVVTP1: 30313
23334
35363
74041

;COME HERE AFTER A TRAP THROUGH VECTOR 4.
VVV10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
CMP R2,#VVV2+2
BEQ VVV11 ;BRANCH IF YES.
CMP R2,#VVV2+4
BEQ VVV11 ;BRANCH IF YES.
JMP @#FPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
VVV11: MOV R2,STMP2
CMP (SP)+,(SP)+
IS: ERROR 31 ;FDST FORK X ODD ADD
BR VVVDONE

;REPORT RO MODIFIED.
VVV15: MOV RO,STMP4
  
```

```

2797 011334 012737 003353 001240      MOV      #VWBFO-5701, @#STMP3
2798 011342 104032      15:     ERROR      32      ;RO MODIFIED!
2799 011344 000407      BR       VVVDONE
2800
2801      ;REPORT INCORRECT DATA.
2802 011346 012737 011254 001240  VVV20:  MOV      #VVVBFO, @#STMP3
2803 011354 012737 011264 001242      MOV      #VVVTP1, @#STMP4
2804 011362 104033      15:     ERROR      33      ;BAD DATA
2805 011364      VVVVDONE:
2806 011364 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
2807      ;SEE IF THE USER HAS EXPRESSED
2808      ;THE DESIRE TO CHANGE THE SOFTWARE
2809      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2810      ;THE USER TYPED CONTROL G?).
2811
2812      ;*****
2813      ;*TEST 11      FDST MODE 7, INDEX DEFERRED MODE, TEST
2814      ;*
2815      ;*THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD.
2816      ;*
2817      ;*****
2818 011366 000004      TST11:  SCOPE
2819
2820      WWW1:
2821 011370      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2822 011372 104413      MOV      #200, RO      ;ENTER DOUBLE FLOATING MODE.
2823 011376 170100      LDFPS     RO
2824 011400 012701 011516      MOV      #WWWBFO, R1      ;SET UP THE OUTPUT DATA BUFFER.
2825 011404 012700 177777      MOV      #-1, RO
2826 011410 012702 000004      MOV      #4, R2
2827 011414 010021      15:     MOV      RO, (R1)+
2828 011416 077202      SOB      R2, 15
2829 011420 012737 011546 000004      MOV      #WWWIC, @#ERRVECT ;SET UP FOR TRAPS TO 4.
2830 011426 012700 011526      MOV      #WWWTP1, RO      ;SET UP ACO.
2831 011432 172410      LDD      (RO), ACO
2832 011434 012737 011460 001236      MOV      #WWW2, @#STMP2
2833 011442 012700 003635      MOV      #WWWBF1-5701, RO ;SET UP THE DESTINATION ADDRESS.
2834 011446 012701 000001      MOV      #1, R1
2835 011452 012737 011516 011536      MOV      #WWWBFO, @#WWWBF1
2836 011460 174070 005701      WWW2:   STD      ACO, @5701(RO) ;TEST INSTRUCTION.
2837
2838 011464 020027 003635      CMP      RO, #WWWBF1-5701 ;IS RO CORRECT?
2839 011470 001044      BNE      WWW15           ;BRANCH IF INCORRECT.
2840 011472 012702 011516      MOV      #WWWBFO, R2      ;WAS THE DATA OUTPUT CORRECTLY?
2841 011476 012703 011526      MOV      #WWWTP1, R3
2842 011502 012704 000004      MOV      #4, R4
2843 011506 022223      15:     CMP      (R2)+, (R3)+
2844 011510 001043      BNE      WWW20           ;BRANCH IF DATA IS INCORRECT.
2845 011512 077403      SOB      R4, 15
2846 011514 000450      BR       WWWDONE
2847 011516 177777      WWWBFO: -1
2848 011520 177777      -1
2849 011522 177777      -1
2850 011524 177777      -1
2851 011526 041424      WWWTP1: 41424
2852 011530 034445      34445

```

B05

2853 011532 046475
2854 011534 051525
2855 011536 177777
2856 011540 177777
2857 011542 177777
2858 011544 177777
2859
2860
2861 011546 011602
2862 011550 020227 011462
2863 011554 001405
2864 011556 020227 011464
2865 011562 001402
2866 011564 000137 042564
2867
2868 011570 010237 001236
2869 011574 022626
2870 011576 104034
2871 011600 000416
2872
2873
2874 011602 010037 001242
2875 011606 012737 003615 001240
2876 011614 104035
2877 011616 000407
2878
2879
2880 011620 012737 011516 001240
2881 011626 012737 011526 001242
2882 011634 104036
2883 011636
2884 011636 104412
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901 011642 104413
2902 011644 004767 000330
2903 011650 000000
2904 011652 000000
2905 011654 000000
2906 011656 000000
2907 011660 000000
2908 011662 000000
2909 011664 000000

```
46475
051525
MMBF1: -1
-1
-1
-1

;TRAP THROUGH 4 TO HERE.
MM10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
CMP R2,#MM12+2
BEQ MM11 ;BRANCH IF YES.
CMP R2,#MM12+4
BEQ MM11 ;BRANCH IF YES.
JMP J#FPPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
;REPORT FAILURE OF FDST FORK RESULTED IN AN ODD ADDRESS TRAP TO 4.
MM11: MOV R2,#STMP2
CMP (SP)+,(SP)+
IS: ERROR 34 ;FDST FORK X ODD ADD
BR MM10DONE

;REPORT RD MODIFIED.
MM15: MOV RD,#STMP4
MOV #MMBFO-5701,#STMP3
IS: ERROR 35 ;RD MODIFIED!
BR MM10DONE

;REPORT DATA INCORRECT
MM20: MOV #MMBFO,#STMP3
MOV #MM1TP1,#STMP4
IS: ERROR 36 ;BAD DATA
MM10DONE: RSETUP
;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

;*****
;TEST 12 STCFD TEST
;
;THIS IS A TEST OF THE STCFD INSTRUCTION.
;
;*****
TST12: SCOPE

;AC=C
XXX1: LPERR
JSR PC,STCFDS ;SET UP THE LOOP ON ERROR ADDRESS.
IS: 0 ;AC
0
0
0
2S: 0 ;RES
0
0
```

```

2909 011666 000000
2910 011670 000000
2911 011672 000000
2912 011674 177777
2913 011676 177777
2914 011700 047000
2915 011702 047004
2916 011704 177777
2917 011706 147004
2918 011710 104042
2919 011712 000401
2920 011714 104043
2921 011716
2922
2923
2924 011716 104413
2925 011720 004767 000254
2926 011724 017203
2927 011726 142536
2928 011730 047506
2929 011732 172031
2930 011734 017203
2931 011736 142536
2932 011740 000000
2933 011742 000000
2934 011744 017203
2935 011746 142536
2936 011750 047506
2937 011752 172031
2938 011754 040000
2939 011756 040000
2940 011760 177777
2941 011762 177777
2942 011764 104044
2943 011766 000401
2944 011770 104040
2945 011772
2946
2947
2948 011772 104413
2949 011774 004767 000200
2950 012000 050717
2951 012002 027374
2952 012004 075767
2953 012006 077071
2954 012010 050717
2955 012012 027374
2956 012014 000000
2957 012016 000000
2958 012020 000000
2959 012022 000000
2960 012024 000000
2961 012026 000000
2962 012030 047000
2963 012032 047000
2964 012034 177777

```

: ERROR RES.
: FPS BEFORE EXECUTION.
: FPS AFTER EXECUTION.
: FEC
: ERROR FPS.
;FDFL(---FDFLXST 767
;BUT EZBT X ST560 TO 061 INTO 261
: SET UP THE LOOP ON ERROR ADDRESS.
PC,STCFDS
;AC
: RES
: ERROR RES.
: FPS BEFORE EXECUTION.
: FPS AFTER EXECUTION.
: FEC
: ERROR FPS.
;X11(1,0)---0 X ST766
: SET UP THE LOOP ON ERROR ADDRESS.
PC,STCFDS
;AC
: RES
: ERROR RES.
: FPS BEFORE EXECUTION.
: FPS AFTER EXECUTION.
: FEC

```

2965 012036 174002          174002          ;ERROR FPS.
2966 012040 104045          ERROR          45          ;BUT OPIC X ST25:
2967 012042 000401          BR            65
2968 012044 104046          ERROR          46          ;BUT EZBT X ST421
2969 012046
2970
2971 012046
2972 012046 104413          LPERR
2973 012050 004767          JSR           PC,STCFDS          ;SET UP THE LOOP ON ERROR ADDRESS.
2974 012054 020212          15:          20212          ;AC
2975 012056 032425          32425
2976 012060 026272          26272
2977 012062 002123          02123
2978 012064 020212          25:          20212          ;RES
2979 012066 032425          32425
2980 012070 000000          0
2981 012072 000000          0
2982 012074 020212          35:          20212          ;ERROR RES.
2983 012076 032425          32425
2984 012100 100000          100000
2985 012102 000000          0
2986 012104 040000          45:          40000          ;FPS BEFORE EXECUTION.
2987 012106 040000          40000          ;FPS AFTER EXECUTION.
2988 012110 177777          -1
2989 012112 177777          -1
2990 012114 104047          55:          ERROR          47          ;ERROR FPS.
2991 012116 000401          BR            65          ;BUT FD IN ROUND X ST113
2992 012120 104040          ERROR          40
2993 012122
2994
2995 012122
2996 012122 104413          XXX5:        LPERR
2997 012124 004767          JSR           PC,STCFDS          ;SET UP THE LOOP ON ERROR ADDRESS.
2998 012130 121314          15:          121314          ;AC
2999 012132 151617          151617
3000 012134 101112          101112
3001 012136 131415          131415
3002 012140 121314          25:          121314          ;RES
3003 012142 151617          151617
3004 012144 000000          0
3005 012146 000000          0
3006 012150 021314          35:          21314          ;ERROR RES.
3007 012152 151617          151617
3008 012154 000000          0
3009 012156 000000          0
3010 012160 040000          45:          40000          ;FPS BEFORE EXECUTION.
3011 012162 040010          40010          ;FPS AFTER EXECUTION.
3012 012164 177777          -1
3013 012166 177777          -1
3014 012170 104050          55:          ERROR          50          ;ERROR FPS.
3015 012172 000401          BR            65          ;BUT ENBT X ST567 OR BAD SIGN ST460
3016 012174 104040          ERROR          40
3017 012176 000535          BR            XXXDONE
3018
3019
3020

```

3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076

: THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
: THE STCFD INSTRUCTION AND CHECK THE RESULTS. A CALL
: TO IT IS MADE THUS:

```
JSR      PC,@#STCFDS
ACARG:   .WORD  X,X,X,X      ;AC OPERAND
RES:     .WORD  X,X,X,X      ;EXPECTED RESULT
ERRES:   .WORD  X,X,X,X      ;ERROR RESULT
FPSB:    .WORD  X              ;FPS BEFORE EXECUTION
FPSA:    .WORD  X              ;FPS AFTER EXECUTION
FEC:     .WORD  X              ;EXPECTED FEC
EPFPS:   .WORD  X              ;ERROR FPS.
ERR1:    ERROR  X              ;DATA ERROR.
BR       CONT
ERR2:    ERROR  X              ;FPS ERROR.
CONT:                                         ;RETURN ADDRESS
```

: THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
: THE STCFD INSTRUCTION IS EXECUTED.
: THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
: COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
: TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
: COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
: TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
: REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
: STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
: ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
: THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
: WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
: RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
: REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

012200 012601
012202 012700 000200
012206 170100
012210 010100
012212 172410
012214 012700 177777
012220 012702 012462
012224 012703 000004
012230 010022
012232 077302
012234 016100 000030
012240 170100
012242 012737 012254 001236
012250 012700 012462
012254 176010
012256 170204
012260 170305
012262 010102
012264 010237 001240
012270 062702 000010
012274 010237 001244
012300 012737 012462 001242

```
STCFDS: MOV      (SP)+,R1      ;PICK UP THE POINTER TO THE OPERANDS.
        MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE.
        LDFPS   R0
        MOV      R1,R0        ;LOAD ACO.
        LDD     (R0),ACO
        MOV      #-1,R0       ;FILL THE OUTPUT BUFFER WITH -1'S.
        MOV      #STCFT,R2
        MOV      #4,R3
15:     MOV      R0,(R2)+
        SOB     R3,15
        MOV      30(R1),R0     ;LOAD THE FPS.
        LDFPS   R0
        MOV      #25,@#STMP2
        MOV      #STCFT,R0     ;SET UP THE DESTINATION ADDRESS.
25:     STCFD   ACO,(R0)      ;TEST INSTRUCTION.

        STFPS   R4            ;GET THE FPS.
        STST   R5            ;GET THE FEC.
        MOV     R1,R2        ;SAVE THE DATA IN CASE OF ERROR.
        MOV     R2,@#STMP3
        ADD    #10,R2
        MOV     R2,@#STMP5
        MOV     #STCFT,@#STMP4
```


F05

```

3077 012306 010437 001250      MOV      R4,2#STMP7
3078 012312 016137 000032 001252      MOV      32(R1),2#STMP10
3079
3080 012320 010102              MOV      R1,R2          ;CHECK THE RESULT.
3081 012322 062702 000010      ADD      #10,R2
3082 012326 012703 012462      MOV      #STCFT,R3
3083 012332 012700 000004      MOV      #4,R0
3084 012336 022223              35:     CMP      (R2)+,(R3)+
3085 012340 001014              BNE     15$            ;BRANCH IF INCORRECT.
3086 012342 077003              SOB     R0,3$
3087
3088 012344 016102 000032      MOV      32(R1),R2
3089 012350 020204              CMP      R2,R4          ;IS THE FPS CORRECT?
3090 012352 001025              BNE     20$            ;BRANCH IF FPS INCORRECT.
3091 012354 005702              TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
3092 012356 100003              BPL     4$              ;GO AHEAD AND CHECK THE FEC.
3093 012360 026105 000036      CMP      36(R1),R5
3094 012364 001027              BNE     25$            ;BRANCH IF FEC IS INCORRECT.
3095 012366 000161 000046      45:     JMP      46(R1)       ;RETURN.
3096
3097              ;RESULT INCORRECT:
3098 012372 010102 15$:     MOV      R1,R2          ;SEE IF ERROR WAS ANTICIPATED.
3099 012374 062702 000020      ADD      #20,R2
3100 012400 012703 012462      MOV      #STCFT,R3
3101 012404 012700 000004      MOV      #4,R0
3102 012410 022223              16$:     CMP      (R2)+,(R3)+
3103 012412 001003              BNE     17$            ;BRANCH IF NOT ANTICIPATED.
3104 012414 077003              SOB     R0,16$
3105 012416 000161 000040      JMP      40(R1)       ;IF ERROR WAS ANTICIPATED RETURN.
3106              ;OTHERWISE REPORT RESULT INCORRECT HERE.
3107 012422 17$:
3108 012422 104037 18$:     ERROR   37            ;DATA ERROR
3109 012424 000760              BR      4$
3110
3111              ;FPS INCORRECT:
3112 012426 020461 000034      20$:     CMP      R4,34(R1)   ;WAS THE ERROR ANTICIPATED.
3113 012432 001002              BNE     21$            ;BRANCH IF NOT ANTICIPATED.
3114 012434 000161 000044      JMP      44(R1)       ;IF IT WAS ANTICIPATED RETURN.
3115
3116              ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
3117 012440 21$:
3118 012440 104040 22$:     ERROR   40            ;FPS X
3119 012442 000751              BR      4$
3120
3121              ;REPORT FEC INCORRECT:
3122 012444 016137 000036 001256 25$:     MOV      36(R1),2#STMP12
3123 012452 010537 001254      MOV      R5,2#STMP11
3124 012456 104041      26$:     ERROR   41            ;FEC X
3125 012460 000742              BR      4$
3126 012462 177777 177777 177777 STCFT:  -1,-1,-1,-1
3127 012470 177777
3128 012472      XXXDONE:
3129 012472 104412      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3130              ;SEE IF THE USER HAS EXPRESSED
3131              ;THE DESIRE TO CHANGE THE SOFTWARE
3132              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS

```



```

3189 012624 104052          ERROR 52
3190 012626          6S:
3191          ;YYY3:
3192 012626          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3193 012626 104413          JSR          PC,STCDF5
3194 012630 004767 000200          77777          ;ACO
3195 012634 077777          -1
3196 012636 177777          100000
3197 012640 100000          0
3198 012642 000000          0
3199 012644 000000          2S:          ;RES
3200 012646 000000          0
3201 012650 177777          -1
3202 012652 177777          -1
3203 012654 077777          3S:          ;ERROR RES.
3204 012656 177777          -1
3205 012660 177777          -1
3206 012662 177777          -1
3207 012664 040200          4S:          ;FPS BEFORE EXECUTION.
3208 012666 040206          ;FPS AFTER EXECUTION.
3209 012670 177777          -1          ;FEC
3210 012672 040204          40204          ;ERROR FPS.
3211 012674 104055          5S:          ERROR 55
3212 012676 000401          BR          6S
3213 012700 104056          ERROR 56          ;BUT EZBT X ST421 TO 062 INTO 262
3214 012702          6S:
3215          ;YYY4:
3216 012702          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3217 012702 104413          JSR          PC,STCDF5
3218 012704 004767 000124          77777          ;ACO
3219 012710 077777          -1
3220 012712 177777          100000
3221 012714 100000          0
3222 012716 000000          0
3223 012720 000000          2S:          ;RES
3224 012722 000000          0
3225 012724 177777          -1
3226 012726 177777          -1
3227 012730 077777          3S:          ;ERROR RES.
3228 012732 177777          -1
3229 012734 177777          -1
3230 012736 177777          -1
3231 012740 040200          4S:          ;FPS BEFORE EXECUTION.
3232 012742 040206          ;FPS AFTER EXECUTION.
3233 012744 177777          -1          ;FEC
3234 012746 140206          140206          ;ERROR FPS.
3235 012750 104055          5S:          ERROR 55
3236 012752 000401          BR          6S
3237 012754 104057          ERROR 57          ;BUT FIV ST262 TO 123 INTO 103
3238 012756          6S:
3239          ;YYY5:
3240 012756          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3241 012756 104413          JSR          PC,STCDF5
3242 012760 004767 000050          77777          ;ACO
3243 012764 177777          -1
3244 012766 177777          -1

```

```

3245 012770 100000
3246 012772 000000
3247 012774 100000
3248 012776 000000
3249 013000 177777
3250 013002 177777
3251 013004 000000
3252 013006 000000
3253 013010 177777
3254 013012 177777
3255 013014 047200
3256 013016 147216
3257 013020 000010
3258 013022 047206
3259 013024 104060
3260 013026 000400
3261 013030 104061
3262 013032 000535
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 013034 012601
3296 013036 012700 000200
3297 013042 170100
3298 013044 010100
3299 013046 172410
3300 013050 012700 177777
    
```

```

100000
0
25: 100000 :RES
0
-1
-1
35: 0 :ERROR RES.
0
-1
-1
45: 47200 :FPS BEFORE EXECUTION.
147216 :FPS AFTER EXECUTION.
10 :FEC
47206 :ERROR FPS.
55: ERROR 60 :BUT FIV ST262 FAIL TO 103 INT 123
BR 65
ERROR 61 :BUT FLAG ST 147 X TO ST 361 INTO 365
BR YYYDONE
    
```

;THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
 ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
 ;TO IT IS MADE THUS:

```

JSR PC, #STCFDS
ACARG: .WORD X,X,X,X ;AC OPERAND
RES: .WORD X,X,X,X ;EXPECTED RESULT
ERRES: .WORD X,X,X,X ;ERROR RESULT
FPSB: .WORD X ;FPS BEFORE EXECUTION
FPSA: .WORD X ;FPS AFTER EXECUTION
FEC: .WORD X ;EXPECTED FEC
ERFPS: .WORD X ;ERROR FPS.
ERR1: ERROR X ;DATA ERROR.
BR CONT
ERR2: ERROR X ;FPS ERROR.
CONT: ;RETURN ADDRESS
    
```

;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
 ;THE STCDF INSTRUCTION IS EXECUTED.
 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
 ;STCDF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

```

STCFDS: MOV (SP)+, R1 ;PICK UP THE POINTER TO THE OPERANDS.
MOV #200, R0 ;ENTER DOUBLE FLOATING MODE.
LDFPS R0
MOV R1, R0 ;LOAD ACO.
LDD (R0), ACO
MOV #-1, R0 ;FILL THE OUTPUT BUFFER WITH -1'S.
    
```

```

3301 013054 012702 013316      MOV      #STCDT,R2
3302 013060 012703 000004      MOV      #4,R3
3303 013064 010022      15:     MOV      RO,(R2)+
3304 013066 077302      SOB      R3,15
3305 013070 016100 000030      MOV      30(R1),RO      ;LOAD THE FPS.
3306 013074 170100      LDFPS   RO
3307 013076 012737 013110 001236      MOV      #25,2#STMP2
3308 013104 012700 013316      MOV      #STCDT,RO      ;SET UP THE DESTINATION ADDRESS.
3309 013110 176010      25:     STCDF   ACO,(RO)      ;TEST INSTRUCTION.
3310
3311 013112 170204      STFPS   R4      ;GET THE FPS.
3312 013114 170305      STST   R5      ;GET THE FEC.
3313 013116 010102      MOV      R1,R2      ;SAVE THE DATA IN CASE OF ERROR.
3314 013120 010237 001240      MOV      R2,2#STMP3
3315 013124 062702 000010      ADD      #10,R2
3316 013130 010237 001244      MOV      R2,2#STMP5
3317 013134 012737 013316 001242      MOV      #STCDT,2#STMP4
3318 013142 010437 001250      MOV      R4,2#STMP7
3319 013146 016137 000032 001252      MOV      32(R1),2#STMP10
3320
3321 013154 010102      MOV      R1,R2      ;CHECK THE RESULT.
3322 013156 062702 000010      ADD      #10,R2
3323 013162 012703 013316      MOV      #STCDT,R3
3324 013166 012700 000004      MOV      #4,RO
3325 013172 022223      35:     CMP      (R2)+,(R3)+
3326 013174 001014      BNE     155
3327 013176 077003      SOB     RO,35      ;BRANCH IF INCORRECT.
3328
3329 013200 016102 000032      MOV      32(R1),R2
3330 013204 020204      CMP     R2,R4      ;IS THE FPS CORRECT?
3331 013206 001025      BNE     205      ;BRANCH IF FPS INCORRECT.
3332 013210 005702      TST    R2      ;IF EXPECTED FPS IS NEGATIVE, THEN
3333 013212 100003      BPL     45      ;GO AHEAD AND CHECK THE FEC.
3334 013214 026105 000034      CMP     34(R1),R5
3335 013220 001027      BNE     255      ;BRANCH IF FEC IS INCORRECT.
3336 013222 000161 000046      45:     JMP     46(R1)      ;RETURN.
3337
3338      ;RESULT INCORRECT:
3339 013226 010102      155:    MOV     R1,R2      ;SEE IF ERROR WAS ANTICIPATED.
3340 013230 062702 000020      ADD     #20,R2
3341 013234 012703 013316      MOV     #STCDT,R3
3342 013240 012700 000004      MOV     #4,RO
3343 013244 022223      165:    CMP     (R2)+,(R3)+
3344 013246 001003      BNE     175      ;BRANCH IF NOT ANTICIPATED.
3345 013250 077003      SOB     RO,165
3346 013252 000161 000040      JMP     40(R1)      ;IF ERROR WAS ANTICIPATED RETURN.
3347      ;OTHERWISE REPORT RESULT INCORRECT HERE.
3348 013256      175:
3349 013256 104051      185:    ERROR  51      ;DATA ERROR
3350 013260 000760      BR      45
3351
3352      ;FPS INCORRECT:
3353 013262 020461 000034      205:    CMP     R4,34(R1)      ;WAS THE ERROR ANTICIPATED.
3354 013266 001002      BNE     215      ;BRANCH IF NOT ANTICIPATED.
3355 013270 000161 000044      JMP     44(R1)      ;IF IT WAS ANTICIPATED RETURN.
3356

```

```

3357 ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
3358 013274 215:
3359 013274 104052 225: ERROR 52 ;FPS X
3360 013276 000751 BR 45
3361
3362 ;REPORT FEC INCORRECT:
3363 013300 016137 000036 001256 255: MOV 36(R1),2#STMP12
3364 013306 010537 001254 MOV R5,2#STMP11
3365 013312 104053 265: ERROR 53 ;FEC X
3366 013314 000742 BR 45
3367 013316 177777 177777 177777 STCDT: -1,-1,-1,-1
3368 013324 177777
3369 013326 YYYDONE:
3370 013326 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3371 ;SEE IF THE USER HAS EXPRESSED
3372 ;THE DESIRE TO CHANGE THE SOFTWARE
3373 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3374 ;THE USER TYPED CONTROL G?).
3375 ;*****
3376 ;*TEST 14 STCFD WITH ILLEGAL ACCUMULATOR TEST
3377 ;*
3378 ;*THIS TEST STCFD WITH ILLEGAL AC 6.
3379 ;*
3380 ;*****
3381 013330 000004 †ST14: SCOPE
3382
3383 ZZZ1:
3384 013332 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3385 013332 104413 MOV #40000,R0 ;DISSABLE INTERRUPTS.
3386 013334 012700 040000 LDFPS R0
3387 013342 012737 013350 001236 MOV #ZZZ2,2#STMP2
3388 013350 176006 ZZZ2: STCFD ACO,AC6 ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.
3389
3390 013352 170204 STFPS R4 ;GET FPS.
3391 013354 170305 STST R5 ;GET FEC.
3392 013356 020427 140000 CMP R4,#140000 ;IS FPS CORRECT?
3393 013362 001004 BNE ZZZ10 ;BRANCH IF INCORRECT FPS.
3394 013364 022705 000002 CMP #2,R5 ;IS FEC CORRECT?
3395 013370 001010 BNE ZZZ15 ;BRANCH IF INCORRECT.
3396 013372 000415 BR ZZZDONE
3397
3398 ;REPORT FPS INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
3399 013374 010437 001242 ZZZ10: MOV R4,2#STMP4
3400 013400 012737 140000 001240 MOV #140000,2#STMP3
3401 013406 104062 15: ERROR 62 ;BUT FDST ST767 X TO 567 INTO 577
3402 013410 000406 BR ZZZDONE
3403
3404 ;REPORT FEC INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
3405 013412 010537 001242 ZZZ15: MOV R5,2#STMP4
3406 013416 012737 000002 001240 MOV #2,2#STMP3
3407 013424 104063 15: ERROR 63 ;FEC<---2 ST577 X
3408 013426 ZZZDONE:
3409 013426 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3410 ;SEE IF THE USER HAS EXPRESSED
3411 ;THE DESIRE TO CHANGE THE SOFTWARE
3412 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS

```

LOS

;THE USER TYPED CONTROL G?).

3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468

013430 C00004
013432
013432 104413
013434 012700 013620
013440 012701 013610
013444 012702 000004
013450 012021
013452 077202
013454 012700 013610
013460 012701 000213
013464 170101
013466 012737 013474 001236
013474 170410
013476 170205
013500 012702 000004
013504 012701 013610
013510 005721
013512 001010
013514 077203
013516 022705 000204
013522 001014
013524 020027 013610
013530 001020
013532 000442
013534 012737 013610 001240
013542 012737 013630 001242
013550 104064
013552 000432
013554 010437 001242
013560 012737 000204 001240
013566 104065
013570 000423
013572 010037 001242
013576 012737 013610 001240
013604 104066
013606 000414
013610 073475
013612 067707
013614 127347

```
*****  
*TEST 15 CLRD TEST  
*  
*THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.  
*  
*****  
TST15: SCOPE  
AAB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #AABTP1,R0 ;SET UP OUTPUT BUFFER  
MOV #AABBFO,R1  
MOV #4,R2  
1$: MOV (R0)+,(R1)+  
SOB R2,1$  
MOV #AABBFO,R0 ;SET UP DESTINATION OPERAND ADDRESS.  
MOV #213,R1 ;SET UP FPS.  
LDFPS R1  
2$: MOV #2$,@#STMP2  
CLRD (R0) ;TEST INSTRUCTION.  
  
STFPS R5 ;GET FPS.  
MOV #4,R2 ;SEE IF RESULT CLEAR, 0.  
3$: MOV #AABBFO,R1  
TST (R1)+  
BNE AAB2 ;BRANCH IF RESULT INCORRECT, NOT 0.  
SOB R2,3$  
CMP #204,R5 ;SEE IF FPS IS CORRECT.  
BNE AAB3 ;BRANCH IF INCORRECT.  
CMP R0,#AABBFO ;SEE IF R0 IS CORRECT.  
BNE AAB4 ;BRANCH IF R0 IS INCORRECT.  
BR AABDONE  
  
;RESULT NOT 0, REPORT ERROR.  
AAB2: MOV #AABBFO,@#STMP3  
MOV #AABTP2,@#STMP4  
1$: ERROR 64 ;BAD DATA = 0 X 11+ZERO ST770 X  
BR AABDONE  
  
;REPORT FPS INCORRECT:  
AAB3: MOV R4,@#STMP4  
MOV #204,@#STMP3  
1$: ERROR 65 ;BAD FPS  
BR AABDONE  
  
;REPORT R0 INCORRECT.  
AAB4: MOV R0,@#STMP4  
MOV #AABBFO,@#STMP3  
1$: ERROR 66  
BR AABDONE  
  
;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.  
AABBFO: 73475  
67707  
127347
```


M05

3469 013616 056770
3470
3471 013620 073475
3472 013622 067707
3473 013624 127347
3474 013626 056770
3475
3476 013630 000000
3477 013632 000000
3478 013634 000000
3479 013636 000000
3480 013640
3481 013640 104412
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493 013642 000004
3494 013644
3495 013644 104413
3496 013646 012700 040200
3497 013652 170100
3498 013654 012737 013662 001236
3499 013662 170407
3500
3501 013664 170204
3502 013666 170305
3503 013670 020427 140200
3504 013674 001004
3505 013676 022705 000002
3506 013702 001010
3507 013704 000415
3508
3509
3510 013706 010437 001242
3511 013712 012737 140200 001240
3512 013720 104067
3513 013722 000406
3514
3515
3516 013724 010537 001242
3517 013730 012737 000002 001240
3518 013736 104070
3519 013740
3520 013740 104412
3521
3522
3523
3524

56770
;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
AABTP1: 73475
67707
127347
56770
;THIS IS THE EXPECTED DATA, RESULT:
AABTP2: 0
0
0
0
AABDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
;*****
;*TEST 16 CLRD WITH ILLEGAL ACCUMULATOR TEST
;*
;*THIS IS A TEST OF CLRD WITH ILLEGAL AC7.
;*
;*****
+ST16: SCOPE
CCB1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #40200,R0 ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
LDFPS R0
MOV #CCB2,@#STMP2
CCB2: CLRD AC7 ;TEST INSTRUCTION.
STFPS R4 ;GET FPS.
STST R5 ;GET FEC.
CMP R4,#140200 ;IS THE FPS CORRECT?
BNE CCB10 ;BRANCH IF FPS IS INCORRECT.
CMP #2,R5 ;IS THE FEC CORRECT?
BNE CCB15 ;BRANCH IF FEC IS INCORRECT.
BR CCBDONE
;REPORT INCORRECT FPS:
CCB10: MOV R4,@#STMP4
MOV #140200,@#STMP3
IS: ERROR 67 ;BUT FOST ST 700X TO 607 INTO 677
BR CCBDONE
;REPORT INCORRECT FEC:
CCB15: MOV R5,@#STMP4
MOV #2,@#STMP3
IS: ERROR 70 ;FEC<---2 ST 677 X
CCBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

N05

MAINDEC-11-FPF34-A
OFFPCA.P11

PDP 11 34 FPP DIAGNOSTIC
31-OCT-76 17:16

MACY11 27(1006) 31-OCT-76 17:35 PAGE 65
T16 CLRD WITH ILLEGAL ACCUMULATOR TEST

3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580

013742 000004
013744
013744 104413
013746 012700 040200
013752 170100
013754 012737 013762 001236
013762 170707
013764 170204
013766 170305
013770 022704 140200
013774 001004
013776 022705 000002
014002 001010
014004 000415
014006 012737 140200 001240
014014 010437 001242
014020 104176
014022 000406
014024 012737 000002 001240
014032 010537 001242
014036 104177
014040
014040 104412
014042 000004

```
*****
*TEST 17      NEGf, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
*
*THIS IS A TEST OF THE SPECIAL
*DEST FLOWS USING THE NEGf INST
*WITH MODE ZERO AND ILLEGAL
*AC7.
*****
TST17: SCOPE
VVB1:
LPERR          :SET UP THE LOOP ON ERROR ADDRESS.
MOV #40200,R0  ;SET UP THE FPS, FID=1 AND FD=1.
LDFPS R0
MOV #VVB2,2*STMP2
VVB2: NEGf AC7 ;TEST INSTRUCTION.
STFPS R4      ;GET FPS.
STST R5       ;GET FEC.
CMP #140200,R4 ;IS FPS CORRECT?
BNE VVB10     ;BRANCH IF FPS IS INCORRECT.
CMP #2,R5     ;IS FEC CORRECT?
BNE VVB15     ;BRANCH IF FEC IS INCORRECT.
BR VVBDONE
;REPORT INCORRECT FPS:
VVB10: MOV #140200,2*STMP3
MOV R4,2*STMP4
IS: ERROR 176 ;FPS BAD
BR VVBDONE
;REPORT FEC INCORRECT:
VVB15: MOV #2,2*STMP3
MOV R5,2*STMP4
IS: ERROR 177 ;FEC BAD
VVBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
*****
*TEST 20      NEGf, ABSF AND TSTF SOURCE MODE 0 TEST
*
*THIS IS A TEST THE NEGf, ABSF AND TSTF
*SOURCE FLOWS. THE NEGf INSTRUCTION
*IS USED TO TEST MODE 0
*****
TST20: SCOPE
```

B06

```

014204 104413
014205 012700 000200
014206 170100
014207 012700 014216
014208 172410
014209 005000
014210 170100
014211 012700 014226
014212 172410
014213 012700 000201
014214 170100
014215 012737 014210 001236
014110 170700 000200
014112 170205
014114 012700 000200
014120 170100
014122 012700 014236
014126 174010
014130 012701 000004
014134 005720
014136 001005
014140 077103
014142 022705 000204
014146 001014
014150 000442
014152 012737 014226 001242
014160 012737 014246 001240
014166 012737 014236 001244
014174 104071
014176 000427
014200 012737 000204 001240
014206 010537 001242
014212 104072
014214 000420
014216 101112
014220 131415
014222 161710
014224 111213
014226 000000
014230 000000
014232 000000
014234 000000
014236 177777
  
```

DDB1:

 L PERR ;SET UP THE LOOP ON ERROR ADDRESS.

 MOV #200,RO ;SET FD MODE.

 LDFPS RO

 MOV #DDBTP1,RO ;SET JP ACC.

 LDO (RO),ACC ;SET ACC = 0

 CLR RO ;CLEAR THE FPS.

 LDFPS RO

 MOV #DDBTP2,RO ;LOAD ACC TO BE A FLOATING 0.

 LDF (RO),ACC ;SET ACC=ZERO

 ;FLOAT

 ;SET FD MODE.

 MOV #201,RO

 LDFPS RO

 MOV #DDB2,2*STMP2

DDB2:

 NEG0 ACC ;TEST INSTRUCTION.

 STFPS R5 ;GET FPS.

 MOV #200,RO ;SET FD MODE.

 LDFPS RO

 MOV #DDBBFO,RO ;GET THE RESULT OUT OF ACC.

 STD ACC,(RO) ;SEE IF THE RESULT IS CORRECT.

IS:

 MOV #4,R1

 TST (RO)+

 BNE DDB5 ;BRANCH IF THE RESULT IS INCORRECT.

 SOB R1,IS

 CMP #204,R5 ;IS THE FPS CORRECT?

 BNE DDB6 ;BRANCH IF THE FPS IS INCORRECT.

 BR DDBDONE

;RESULT INCORRECT, REPORT FAILURE:

 DDB5: MOV #DDBTP2,2*STMP4 ;EXPECT DO

 MOV #DDBTP3,2*STMP3 ;PREV FO IMPURE

 MOV #DDBBFO,2*STMP5 ;GOT

IS:

 ERROR 71

 BR DDBDONE

;REPORT FPS INCORRECT:

 DDB6: MOV #204,2*STMP3

 MOV R5,2*STMP4

IS:

 ERROR 72

 BR DDBDONE

;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.

 DDBTP1: 101112

 131415

 161710

 111213

 DDBTP2: 0

 0

 0

 0

DDBBFO: -1

C06

3637 014240 177777
3638 014242 177777
3639 014244 177777
3640 014246 000000
3641 014250 000000
3642 014252 161710
3643 014254 111213
3644
3645 014256
3646 014256 104412
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660 014260 000004
3661
3662 014262
3663 014262 104413
3664 014264 012700 014372
3665 014270 012751 014422
3666 014274 012732 000004
3667 014300 012021
3668 014302 077202
3669 014304 012700 000200
3670 014310 170130
3671 014312 012700 014422
3672 014316 012737 014332 DC1236
3673 014324 012737 014432 000004
3674 014332 170710
3675
3676 014334 170205
3677 014336 012701 014422
3678 014342 012702 000004
3679 014346 005721
3680 014350 001046
3681 014352 077203
3682
3683 014354 020027 014422
3684 014360 001055
3685 014362 022705 000204
3686 014366 001061
3687 014370 000466
3688
3689
3690 014372 000177
3691 014374 167574
3692 014376 137271

-1
-1
-1
CCBTP3: 0
0
161710
111213

DCBDONE:
RSE*UP

:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER (HAS
:THE USER TYPED CONTROL G?).

:*****
:TEST 21 NEG. ABSF AND TSTF SOURCE MODE 1 TEST
:*****
:THIS IS A TEST THE NEG. ABSF AND TSTF
:SOURCE FLOWS. THE NEG. INSTRUCTION
:IS USED TO TEST MODE 1
:*****

TST21: SCOPE

EEB1: LPERR :SET UP THE LOOP ON ERROR ADDRESS.
MOV #EEBTP1,R0 :SET UP THE DATA BUFFER.
MOV #EEBBF1,R1
MOV #4,R2
IS: MOV (R0)+,(R1)+
SOB R2,IS
MOV #200,R0 ;SET FD MODE.
LDFPS R0
MOV #EEBBF1,R0 ;SET UP THE OPERAND ADDRESS.
MOV #EEB2,2*STMP2
MOV #EEB10,2*ERRVECT :SET UP VECTOR 4 IN CASE OF ERROR.
EEB2: NEG. (R0) :TEST INSTRUCTION.

STFPS R5 ;GET FPS.
MOV #EEBBF1,R1 ;SEE IF RESULT IS CORRECT.
MOV #4,R2
IS: TST (R1)+
BNE EEB15 ;BRANCH IF NOT CORRECT.
SOB R2,IS

CMP R0,#EEBBF1 ;IS R0 CORRECT?
BNE EEB20 ;BRANCH IF NOT CORRECT.
CMP #204,R5 ;IS THE FPS CORRECT?
BNE EEB25 ;BRANCH IF NOT CORRECT.
BR EEBDONE

:THESE ARE TEST DATA TABLES AND A BUFFER.

EEBTP1: 177
167574
137271

3693 014438 107675
3694 014438 000000
3695 014438 000000
3696 014438 000000
3697 014438 000000
3698 014410 177777
3699 014412 177777
3700 014414 177777
3701 014416 177777
3702 014420 177777
3703 014422 177777
3704 014424 177777
3705 014426 177777
3706 014428 177777
3707 014430 177777

EEBTP2: 0
0
0
0
0
EEBBFC: -1
-1
-1
-1
-1
EEBBF1: -1
-1
-1
-1
-1

3708 014432 011632
3709 014434 020227 014334
3710 014440 001405
3711 014442 020227 014336
3712 014446 001402
3713 014450 000137 042620
3714
3715 014454 022626
3716 014456 010237 001236
3717 014462 104107
3718 014464 000430
3719

: IF A TRAP TO 4 OCCURS COME HERE:
EEB10: MOV (SP) R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
CMP R2, #EEB2+2
BEQ 1\$;BRANCH IF YES.
CMP R2, #EEB2+4
BEQ 1\$;BRANCH IF YES.
JMP @CPSPUR ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
:REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP TO 4.
1\$: CMP (SP)+, (SP)+ ;RESET THE STACK.
MOV R2, @STMP2
2\$: ERROR 107 ;ODD ADRES
BR EEBDONE ;BUT FDSTX IN ST 771

3720
3721 014466 012737 014402 001242
3722 014474 012737 014372 001240
3723 014502 012737 014422 001244
3724 014510 104073
3725 014512 000415
3726
3727
3728 014514 012737 014422 001240
3729 014522 010037 001242
3730 014526 104074
3731 014530 000406
3732

:REPORT RESULT INCORRECT.
EEB15: MOV #EEBTP2, @STMP4
MOV #EEBTP1, @STMP3
MOV #EEBBF1, @STMP5
1\$: ERROR 73 ;BAD DATA X11#0 ST 312X
BR EEBDONE
:RO INCORRECT:
EEB20: MOV #EEBBF1, @STMP3
MOV RO, @STMP4
1\$: ERROR 74 ;RO BADX
BR EEBDONE

3733
3734 014532 010537 001240
3735 014536 012737 000204 001244
3736 014544 104075
3737

:REPORT FPS INCORRECT:
EEB25: MOV R5, @STMP3
MOV #204, @STMP5
1\$: ERROR 75 ;FPS X

3738 014546
3739 014546 104412
3740
3741
3742
3743
3744
3745
3746
3747
3748

EEBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
: *TEST 22 NEG. ABSF AND TSTF SOURCE MODE 2 TEST
: *
: *THIS IS A TEST THE NEG. ABSF AND TSTF

E06

3749
3750
3751
3752
3753 014550 000004
3754
3755 014552
3756 014552 104413
3757 014554 012700 014662
3758 014560 012701 014712
3759 014564 012702 000004
3760 014570 012021
3761 014572 077202
3762 014574 012700 000200
3763 014600 170100
3764 014602 012700 014712
3765 014606 012737 014622 001236
3766 014614 012737 014722 000004
3767
3768 014622 170620
3769
3770 014624 170205
3771 014626 012701 014712
3772 014632 012702 000004
3773 014636 005721
3774 014640 001046
3775 014642 077203
3776
3777 014644 020027 014722
3778 014650 001055
3779 014652 022705 000204
3780 014656 001061
3781 014660 000466
3782
3783
3784 014662 000177
3785 014664 167574
3786 014666 137271
3787 014670 107675
3788 014672 000000
3789 014674 000000
3790 014676 000000
3791 014700 000000
3792 014702 177777
3793 014704 177777
3794 014706 177777
3795 014710 177777
3796 014712 177777
3797 014714 177777
3798 014716 177777
3799 014720 177777
3800
3801
3802 014722 011602
3803 014724 020227 014624
3804 014730 001405

: *SOURCE FLOWS. THE ABSD INSTRUCTION
: *IS USED TO TEST MODE 2
: *
: *****
↑ST22: SCOPE

FFB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #FFBTP1,R0 ;SET UP THE DATA BUFFER.
MOV #FFBBF1,R1
MOV #4,R2
IS: MOV (R0)+,(R1)+
SOB R2,IS
MOV #200,R0 ;SET FD.
LDFPS R0
MOV #FFBBF1,R0 ;SET UP THE OPERAND ADDRESS.
MOV #FFB2,#STMP2
MOV #FFB10,#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

FFB2: ABSD (R0)+ ;TEST INSTRUCTION.

STFPS R5 ;GET FPS.
MOV #FFBBF1,R1 ;CHECK RESULT.
IS: MOV #4,R2
TST (R1)+
BNE FFB15 ;BRANCH IF INCORRECT.
SOB R2,IS

CMP R0,#FFBBF1+10 ;IS R0 CORRECT?
BNE FFB20 ;BRANCH IF INCORRECT.
CMP #204,R5 ;IS THE FPS CORRECT?
BNE FFB25 ;BRANCH IF INCORRECT.
BR FFBDONE

:THESE ARE TEST DATA TABLES AND DATA BUFFER.
FFBTP1: 177
167574
137271
107675
FFBTP2: 0
0
0
0
FFBBF0: -1
-1
-1
-1
FFBBF1: -1
-1
-1
-1
-1

:IF A TRAP TO 4 OCCURS COME HERE.
FFB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2,#FFB2+2
BEQ IS ;BRANCH IF YES.

F06

```

3805 014732 020227 014626      CMP      R2,#FFB2+4
3806 014736 001402                BEQ      1$          ;BRANCH IF YES.
3807 014740 000137 042620      JMP      @CPSPUR    ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3808                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3809 014744 022626      1$:      CMP      (SP)+,(SP)+
3810 014746 010237 001236      MOV      R2,@$TMP2
3811 014752 104076      2$:      ERROR   76          ;ODD ADRES
3812 014754 000430      BR       FFBDONE    ;BUT FDSTX IN ST 771
3813
3814                                ;REPORT RESULT INCORRECT:
3815 014756 012737 014672 001240      FFB15:  MOV      #FFBTP2,@$TMP3
3816 014764 012737 014662 001242      MOV      #FFBTP1,@$TMP4
3817 014772 012737 014712 001244      MOV      #FFBBF1,@$TMP5
3818 015000 104077      1$:      ERROR   77          ;BAD DATA X11*0 ST 312X
3819 015002 000415      BR       FFBDONE
3820
3821                                ;REPORT RO INCORRECT:
3822 015004 012737 014716 001240      FFB20:  MOV      #FFBBF1+4,@$TMP3
3823 015012 010037 001242      MOV      RO,@$TMP4
3824 015016 104100      1$:      ERROR   100         ;RO BADX
3825 015020 000406      BR       FFBDONE
3826
3827                                ;REPORT FPS INCORRECT:
3828 015022 010537 001240      FFB25:  MOV      R5,@$TMP3
3829 015026 012737 000204 001244      MOV      #204,@$TMP5
3830 015034 104101      1$:      ERROR   101         ;FPS X
3831
3832                                FFBDONE:
3833 015036 104412      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3834                                ;SEE IF THE USER HAS EXPRESSED
3835                                ;THE DESIRE TO CHANGE THE SOFTWARE
3836                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3837                                ;THE USER TYPED CONTROL G?).
3838
3839                                ;*****
3840                                ;*TEST 23      NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
3841                                ;*
3842                                ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
3843                                ;*SOURCE FLOWS. THE ABSD INSTRUCTION
3844                                ;*IS USED TO TEST MODE 4
3845                                ;*
3846                                ;*****
3847                                †ST23:  SCOPE
3848
3849                                GGB1:
3850 015042 104413      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3851 015044 012700 015152      MOV      #GGBTP1,RO    ;SET UP THE DATA BUFFER.
3852 015050 012701 015172      MOV      #GGBPFO,R1
3853 015054 012702 000004      MOV      #4,R2
3854 015060 012021      1$:      MOV      (RO)+,(R1)+
3855 015062 077202      SOB      R2,1$
3856 015064 012700 000200      MOV      #200,RO      ;SET FD.
3857 015070 170100      LDFPS      RO
3858 015072 012700 015202      MOV      #GGBBF1,RO    ;SET UP THE OPERAND ADDRESS.
3859 015076 012737 015112 001236      MOV      #GGB2,@$TMP2
3860 015104 012737 015212 000004      MOV      #GGB10,@ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

```


G06

```

3861 015112 170640      GGB2:  ABSD  -(R0)          ;TEST INSTRUCTION.
3862
3863 015114 170205      STFPS  RS          ;GET FPS.
3864 015116 012701 015172  MOV    #GGBBFO,R1      ;CHECK RESULT.
3865 015122 012702 000004  MOV    #4,R2
3866 015126 005721      IS:    TST    (R1)+
3867 015130 001046      BNE    GGB15          ;BRANCH IF INCORRECT.
3868 015132 077203      SOB    R2,15
3869
3870 015134 020027 015172  CMP    R0,#GGBBFO     ;IS R0 CORRECT?
3871 015140 001055      BNE    GGB20          ;BRANCH IF INCORRECT.
3872 015142 022705 000204  CMP    #204,R5        ;IS THE FPS CORRECT?
3873 015146 001061      BNE    GGB25          ;BRANCH IF INCORRECT.
3874 015150 000466      BR     GGBDONE
3875
3876      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3877 015152 000177      GGBTP1: 177
3878 015154 117273      117273
3879 015156 147576      147576
3880 015160 177071      177071
3881 015162 000000      GGBTP2: 0
3882 015164 000000      0
3883 015166 000000      0
3884 015170 000000      0
3885 015172 177777      GGBBFO: -1
3886 015174 177777      -1
3887 015176 177777      -1
3888 015200 177777      -1
3889 015202 177777      GGBBF1: -1
3890 015204 177777      -1
3891 015206 177777      -1
3892 015210 177777      -1
3893
3894      ;IF A TRAP TO 4 OCCURS COME HERE.
3895 015212 011602      GGB10: MOV    (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3896 015214 020227 015114  CMP    R2,#GGB2+2
3897 015220 001405      BEQ    15             ;BRANCH IF YES.
3898 015222 020227 015116  CMP    R2,#GGB2+4
3899 015226 001402      BEQ    15             ;BRANCH IF YES.
3900 015230 000137 042620  JMP    @#CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3901      ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3902 015234 022626      IS:    CMP    (SP)+,(SP)+
3903 015236 010237 001236  MOV    R2,@#STMP2
3904 015242 104102      2S:    ERROR  102      ;ODD ADRES
3905 015244 000430      BR     GGBDONE      ;BUT FDSTX IN ST 771
3906
3907      ;REPORT RESULT INCORRECT:
3908 015246 012737 015162 001240  GGB15: MOV    #GGBTP2,@#STMP3
3909 015254 012737 015152 001242  MOV    #GGBTP1,@#STMP4
3910 015262 012737 015172 001244  MOV    #GGBBFO,@#STMP5
3911 015270 104103      IS:    ERROR  103      ;BAD DATA X11*0 ST 312X
3912 015272 000415      BR     GGBDONE
3913
3914      ;REPORT R0 INCORRECT:
3915 015274 012737 015172 001240  GGB20: MOV    #GGBBFO1,@#STMP3
3916 015302 010037 001242  MOV    R0,@#STMP4
  
```

H06

3917 015306 104104
3918 015310 000406
3919
3920
3921 015312 010537 001240
3922 015316 012737 000204 001244
3923 015324 104105
3924
3925 015326
3926 015326 104412
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939 015330 000004
3940
3941 015332
3942 015332 104413
3943 015334 012700 015442
3944 015340 012701 015472
3945 015344 012702 000010
3946 015350 012021
3947 015352 077202
3948 015354 012700 000200
3949 015360 170100
3950 015362 012700 015502
3951 015366 012737 015402 001236
3952 015374 012737 015512 000004
3953
3954 015402 170630
3955
3956 015404 170205
3957 015406 012701 015472
3958 015412 012702 000004
3959 015416 005721
3960 015420 001052
3961 015422 077203
3962 015424 020027 015504
3963 015430 001061
3964 015432 022705 000204
3965 015436 001065
3966 015440 000472
3967
3968
3969 015442 000177
3970 015444 147576
3971 015446 177071
3972 015450 107576 015472 177777

15: ERROR 104 ;RO BADX
BR GGBDONE
:REPORT FPS INCORRECT:
GGB25: MOV R5,2#STMP3
MOV #204,2#STMP5
15: ERROR 105 ;FPS X
GGBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
:TEST 24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
:*****
:THIS IS A TEST THE NEGF, ABSF AND TSTF
:SOURCE FLOWS. THE ABSD INSTRUCTION
:IS USED TO TEST MODE 3
:*****
TST24: SCOPE
HMB1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #HMBTP1,R0 ;SET UP THE DATA BUFFER.
MOV #HMBBF0,R1
MOV #10,R2
15: MOV (R0)+,(R1)+
SOB R2,15
MOV #200,R0 ;SET FD.
LDFPS R0
MOV #HMBBF1,R0 ;SET UP THE OPERAND ADDRESS.
MOV #HMB2,2#STMP2
MOV #HMB10,2#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
HMB2: ABSD 2(R0)+ ;TEST INSTRUCTION.
STFPS R5 ;GET FPS.
MOV #HMBBF0,R1 ;CHECK RESULT.
MOV #4,R2
15: TST (R1)+
BNE HMB15 ;BRANCH IF INCORRECT.
SOB R2,15
CMP R0,#HMBBF1+2 ;IS RO CORRECT?
BNE HMB20 ;BRANCH IF INCORRECT.
CMP #204,R5 ;IS THE FPS CORRECT?
BNE HMB25 ;BRANCH IF INCORRECT.
BR HMBDONE
;THESE ARE TEST DATA TABLES AND DATA BUFFER.
HMBTP1: 177
147576
177071
107576,HMBBF0,-1,-1,-1

3973 015456 177777 177777
3974 015462 000000 000000 000000
3975 015470 000000
3976 015472 177777
3977 015474 177777
3978 015476 177777
3979 015500 177777
3980 015502 177777
3981 015504 177777
3982 015506 177777
3983 015510 177777

HHBTP2: 0.0.0.0
HHBBFO: -1
-1
-1
-1
HHBBF1: -1
-1
-1

3984
3985
3986 015512 011602
3987 015514 020227 015404
3988 015520 001405
3989 015522 020227 015406
3990 015526 001402
3991 015530 000137 042620
3992
3993 015534 022626
3994 015536 010237 001236
3995 015542 104106
3996 015544 000430

; IF A TRAP TO 4 OCCURS COME HERE.
HHB10: MOV (SP), R2 ; SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2, #HHB2+2
BEQ 1\$; BRANCH IF YES.
CMP R2, #HHB2+4
BEQ 1\$; BRANCH IF YES.
JMP @#CPSUR ; OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
; REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
1\$: CMP (SP)+, (SP)+
MOV R2, @#STMP2
2\$: ERROR 106 ; ODD ADRES
BR HHBDONE ; BUT FDSTX IN ST 771

3997
3998
3999 015546 012737 015462 001240
4000 015554 012737 015442 001242
4001 015562 012737 015472 001244
4002 015570 104110
4003 015572 000415

; REPORT RESULT INCORRECT:
HHB15: MOV #HHBTP2, @#STMP3
MOV #HHBTP1, @#STMP4
MOV #HHBBFO, @#STMP5
1\$: ERROR 110 ; BAD DATA X11*0 ST 3127
BR HHBDONE

4004
4005
4006 015574 012737 015504 001240
4007 015602 010037 001242
4008 015606 104111
4009 015610 000406

; REPORT RO INCORRECT:
HHB20: MOV #HHBBF1+2, @#STMP3
MOV RO, @#STMP4
1\$: ERROR 111 ; RO INCORRECT.
BR HHBDONE

4010
4011 015612 010537 001240
4012 015616 012737 000204 001244
4013 015624 104112

; REPORT FPS INCORRECT:
HHB25: MOV R5, @#STMP3
MOV #204, @#STMP5
1\$: ERROR 112 ; FPSX

4014
4015 015626
4016 015626 104412

HHBDONE: RSETUP ; GO INITIALIZE THE FPS AND STACK; AND
; SEE IF THE USER HAS EXPRESSED
; THE DESIRE TO CHANGE THE SOFTWARE
; VIRTUAL CONSOLE SWITCH REGISTER (HAS
; THE USER TYPED CONTROL G?).

4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028

; *TEST 25 NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
; *
; *THIS IS A TEST THE NEGF, ABSF AND TSTF
; *SOURCE FLOWS. THE NEGD INSTRUCTION
; *IS USED TO TEST MODE 5
; *

JOB

MAINDEC-11-FPP34-A PDP 11.34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 74
 DFFPCA.P11 31-OCT-76 17:16 T25 NEGF. ABSF AND TSTF SOURCE MODE 5 TEST

```

4029 015630 000004 TST25: SCOPE
4030
4031 015632 IIB1:
4032 015632 104413 LPERA ;SET UP THE LOOP ON ERROR ADDRESS.
4033 015634 012700 015742 MOV #IIBTP1,R0 ;SET UP THE DATA BUFFER.
4034 015640 012701 015772 MOV #IIBBF0,R1
4035 015644 012702 000010 MOV #10,R2
4036 015650 012021 IS: MOV (R0)+,(R1)+
4037 015652 077202 SOB R2,IS
4038 015654 012700 000200 MOV #200,R0 ;SET FD.
4039 015660 170100 LDFPS R0
4040 015662 012700 016004 MOV #IIBBF1+2,R0 ;SET UP THE OPERAND ADDRESS.
4041 015666 012737 015702 001236 MOV #IIB2,#STMP2
4042 015674 012737 016012 000004 MOV #IIB10,#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
4043
4044 015702 170750 IIB2: NEGD 2-(R0) ;TEST INSTRUCTION.
4045
4046 015704 170205 STFPS R5 ;GET FPS.
4047 015706 012701 015772 MOV #IIBBF0,R1 ;CHECK RESULT.
4048 015712 012702 000004 MOV #4,R2
4049 015716 005721 IS: TST (R1)+
4050 015720 001052 BNE IIB15 ;BRANCH IF INCORRECT.
4051 015722 077203 SOB R2,IS
4052 015724 020027 016002 CMP R0,#IIBBF1 ;IS R0 CORRECT?
4053 015730 001061 BNE IIB20 ;BRANCH IF INCORRECT.
4054 015732 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
4055 015736 001065 BNE IIB25 ;BRANCH IF INCORRECT.
4056 015740 000472 BR IIBDONE
4057
4058 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4059 015742 000176 IIBTP1: 176
4060 015744 177074 177074
4061 015746 127374 127374
4062 015750 157677 015772 177777 157677.IIBBF0,-1,-1,-1
4063 015756 177777 177777
4064 015762 000000 IIBTP2: 0
4065 015764 000000 0
4066 015766 000000 0
4067 015770 000000 0
4068 015772 177777 IIBBF0: -1
4069 015774 177777 -1
4070 015776 177777 -1
4071 016000 177777 -1
4072 016002 177777 IIBBF1: -1
4073 016004 177777 -1
4074 016006 177777 -1
4075 016010 177777 -1
4076
4077 ;IF A TRAP TO 4 OCCURS COME HERE.
4078 016012 011602 IIB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4079 016014 020227 015704 CMP R2,#IIB2+2
4080 016020 001405 BEQ IS ;BRANCH IF YES.
4081 016022 020227 015706 CMP R2,#IIB2+4
4082 016026 001402 BEQ IS ;BRANCH IF YES.
4083 016030 000137 042620 JMP 2#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4084 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
  
```

K06

```

4085 016034 023625 1S: CMP (SP)+,(SP)+
4086 016036 0. J237 001236 MOV R2,#STMP2
4087 016042 104113 2S: ERROR 113 ;OOD ADRES
4088 016044 000430 BR IIBDONE ;BUT FDSTX IN ST 771
4089
4090 ;REPORT RESULT INCORRECT:
4091 016046 012737 015762 001240 IIB15: MOV #IIBTP2,#STMP3
4092 016054 012737 015747 001242 MOV #IIBTP1,#STMP4
4093 016062 012737 015772 001244 MOV #IIBBFO,#STMP5
4094 016070 104114 1S: ERROR 114 ;BAD DATA X11*0 ST 3127
4095 016072 000415 BR IIBDONE
4096
4097 ;REPORT RO INCORRECT:
4098 016074 012737 016002 001240 IIB20: MOV #IIBBF1,#STMP3
4099 016102 010037 001242 MOV RO,#STMP4
4100 016106 104115 1S: ERROR 115 ;RO BADX
4101 016110 000406 BR IIBDONE
4102
4103 ;REPORT FPS INCORRECT:
4104 016112 010537 001240 IIB25: MOV R5,#STMP3
4105 016116 012737 000204 001244 MOV #204,#STMP5
4106 016124 104116 1S: ERROR 116 ;FPSX
4107 IIBDONE:
4108 016126 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4109 ;SEE IF THE USER HAS EXPRESSED
4110 ;THE DESIRE TO CHANGE THE SOFTWARE
4111 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4112 ;THE USER TYPED CONTROL G?).
4113
4114 ;*****
4115 ;*TEST 26 NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
4116 ;*
4117 ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
4118 ;*SOURCE FLOWS. THE ABSD INSTRUCTION
4119 ;*IS USED TO TEST MODE 6
4120 ;*
4121 ;*****
4122 016130 000004 TST26: SCOPE
4123
4124 JJB1:
4125 016132 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4126 016134 012700 016244 MOV #JJBTP1,RO ;SET UP THE DATA BUFFER.
4127 016140 012701 016266 MOV #JJBFO,R1
4128 016144 012702 000004 MOV #4,R2
4129 016150 012021 1S: MOV (RO)+,(R1)+
4130 016152 077202 SOB R2,1S
4131 016154 012700 000200 MOV #200,RO ;SET FO.
4132 016160 170100 LDFPS RO
4133 016162 012700 016257 MOV #JJBFO-7,RO ;SET UP THE OPERAND ADDRESS.
4134 016166 012737 016202 001236 MOV #JJB2,#STMP2
4135 016174 012737 016306 000004 MOV #JJB10,#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
4136
4137 016202 170660 000007 JJB2: ABSD 7(RO) ;TEST INSTRUCTION.
4138
4139 016206 170205 STFPS R5 ;GET FPS.
4140 016210 012701 016266 MOV #JJBFO,R1 ;CHECK RESULT.

```

L06

```

4141 016214 012702 000004          MOV      #4,R2
4142 016220 005721          1$:     TST      (R1)+
4143 016222 001047          BNE     JJB15          ;BRANCH IF INCORRECT.
4144 016224 077203          SOB     R2,1$
4145 016226 020027 016257          CMP     R0,#JJBFO-7    ;IS R0 CORRECT?
4146 016232 001043          BNE     JJB15          ;BRANCH IF INCORRECT.
4147 016234 022705 000204          CMP     #204,R5        ;IS THE FPS CORRECT?
4148 016240 001053          BNE     JJB20          ;BRANCH IF INCORRECT.
4149 016242 000467          BR      JJB DONE

;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4152 016244 000177          JJBTP1: 177
4153 016246 161524          161524
4154 016250 131273          131273
4155 016252 107174 000000          107174.
4156 016256 000000          JJBTP2: 0
4157 016260 000000          0
4158 016262 000000          0
4159 016264 000000          0
4160 016266 177777          JJBFO:  -1
4161 016270 177777          -1
4162 016272 177777          -1
4163 016274 177777          -1
4164 016276 177777          JJBFI:  -1
4165 016300 177777          -1
4166 016302 177777          -1
4167 016304 177777          -1

;IF A TRAP TO 4 OCCURS COME HERE.
4170 016306 011602          JJB10:  MOV     (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4171 016310 020227 016204          CMP     R2,#JJB2+2
4172 016314 001405          BEQ     1$              ;BRANCH IF YES.
4173 016316 020227 016206          CMP     R2,#JJB2+4
4174 016322 001402          BEQ     1$              ;BRANCH IF YES.
4175 016324 000137 042620          JMP     @*CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4176          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
4177 016330 022626          1$:     CMP     (SP)+,(SP)+
4178 016332 010237 001236          MOV     R2,@*STMP2
4179 016336 104117          2$:     ERROR   117          ;ODD ADRES
4180 016340 000430          BR      JJB DONE        ;BUT FDSTX IN ST 771

;REPORT RESULT INCORRECT:
4183 016342 012737 016256 001240          JJB15:  MOV     #JJBTP2,@*STMP3
4184 016350 012737 016244 001242          MOV     #JJBTP1,@*STMP4
4185 016356 012737 016266 001244          MOV     #JJBFO,@*STMP5
4186 016364 104120          1$:     ERROR   120          ;BAD DATA X11*0 ST 3127
4187 016366 000415          BR      JJB DONE

;REPORT R0 INCORRECT:
4190 016370 012737 016257 001240          JJB20:  MOV     #JJBFO-7,@*STMP3
4191 016376 010037 001242          MOV     R0,@*STMP4
4192 016402 104124          1$:     ERROR   124          ;R0 BADX
4193 016404 000406          BR      JJB DONE

;REPORT FPS INCORRECT:
4195 016406 010537 001240          JJB25:  MOV     R5,@*STMP3
4196 016412 012737 000204 001244          MOV     #204,@*STMP5
  
```

M06

```

4197 016420 104122      1S:  ERROR  1P2      ;FPSX
4198 016422
4199 016422 104412      JJBDONE:
4200                                     ;GO INITIALIZE THE FPS AND STACK; AND
4201                                     ;SEE IF THE USER HAS EXPRESSED
4202                                     ;THE DESIRE TO CHANGE THE SOFTWARE
4203                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4204                                     ;THE USER TYPED CONTROL G?).
4205
4206
4207
4208
4209
4210
4211
4212 016424 000004      ;*****
4213                                     ;*TEST 27      NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227 016476 170770 000007      ;*****
4228                                     ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
4229                                     ;*SOURCE FLOWS.  THE ABSD INSTRUCTION
4230                                     ;*IS USED TO TEST MODE 6
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242 016540 000177      ;*****
4243 016542 167574      ;*TEST 27      NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
4244 016544 137271
4245 016546 107675 016570 177777
4246 016554 177777 177777
4247 016560 000000
4248 016562 000000
4249 016564 000000
4250 016566 000000
4251 016570 177777
4252 016572 177777

      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
      MOV      #KKBTP1,RJ      ;SET UP THE DATA BUFFER.
      MOV      #KKB8F0,R1
      MOV      #10,R2
      1S:  MOV      (R0)+,(R1)+
      SOB      R2,1S
      MOV      #200,R0      ;SET FD.
      LDFPS    RC
      MOV      #KKB8F1-7,R0      ;SET UP THE OPERAND ADDRESS.
      MOV      #KKB2,@#STMP2
      MOV      #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

      KKB2:  NEG0      @7(R0)      ;TEST INSTRUCTION.

      STFPS    R5      ;GET FPS.
      MOV      #KKB8F0,R1      ;CHECK RESULT.
      MOV      #4,R2
      1S:  TST      (R1)+
      BNE      KKB15      ;BRANCH IF INCORRECT.
      SOB      R2,1S
      CMP      R0,#KKB8F1-7      ;IS R0 CORRECT?
      BNE      KKB20      ;BRANCH IF INCORRECT.
      CMP      #204,R5      ;IS THE FPS CORRECT?
      BNE      KKB20      ;BRANCH IF INCORRECT.
      BR      KKBDONE

      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
      KKBTP1: 177
               167574
               137271
               107675,KKB8F0,-1,-1,-1

      KKBTP2: 0
               0
               0
               0

      KKB8F0: -1
               -1
  
```



```

4253 016574 177777 -1
4254 016576 177777 -1
4255 016600 177777 KKBBF1: -1
4256 016602 177777 -1
4257 016604 177777 -1
4258 016606 177777 -1
4259
4260 ;IF A TRAP TO 4 OCCURS COME HERE.
4261 016610 011602 KKBB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4262 016612 020227 016500 CMP R2,#KKB2+2
4263 016616 001405 BEQ 1$ ;BRANCH IF YES.
4264 016620 020227 016502 CMP R2,#KKB2+4
4265 016624 001402 BEQ 1$ ;BRANCH IF YES.
4266 016626 000137 042620 JMP 2$CPSUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4267 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
4268 1$: CMP (SP)+(SP)+
4269 016632 022626 001236 MOV R2,2$STMP2
4270 016634 010237 2$: ERROR 123 ;ODD ADRES
4271 016640 104123 BR KKBDONE ;BUT FDSTX IN ST 771
4272 016642 000430
4273 ;REPORT RESULT INCORRECT:
4274 016644 012737 016560 001240 KKBB15: MOV #KKBTP2,2$STMP3
4275 016652 012737 016540 001242 MOV #KKBTP1,2$STMP4
4276 016660 012737 016570 001244 MOV #KKB BBFC,2$STMP5
4277 016666 104124 1$: ERROR 124 ;BAD DATA X11*0 ST 3127
4278 016670 000415 BR KKBDONE
4279
4280 ;REPORT RO INCORRECT:
4281 016672 012737 016571 001240 KKBB20: MOV #KKBBF1-7,2$STMP3
4282 016700 010037 001242 MOV RO,2$STMP4
4283 016704 104125 1$: ERROR 125 ;RO BADX
4284 016706 000406 BR KKBDONE
4285 ;REPORT FPS INCORRECT:
4286 016710 010537 001240 KKBB25: MOV R5,2$STMP3
4287 016714 012737 000204 001244 MOV #204,2$STMP5
4288 016722 104126 1$: ERROR 126 ;FPSX
4289
4290 KKBDONE:
4291 016724 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4292 ;SEE IF THE USER HAS EXPRESSED
4293 ;THE DESIRE TO CHANGE THE SOFTWARE
4294 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4295 ;THE USER TYPED CONTROL G?).
4296 ;*****
4297 ;*TEST 30 NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
4298 ;*
4299 ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
4300 ;*SOURCE FLOWS. THE NEGD INSTRUCTION
4301 ;*IS USED TO TEST MODE 6
4302 ;*
4303 ;*****
4304 016726 000004 TST30: SCOPE
4305 016730 LLB1:
4306 016730 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4307 016732 012700 017030 MOV #LLBTP1,RO ;SET UP THE DATA BUFFER.
4308 016736 012701 017050 MOV #LLBBFO,R1

```

```

4309 016742 012702 000004      MOV      R4,R2
4310 016746 012021      IS:     MOV      (R0)+,(R1)+
4311 016750 077202      SOB      R2,IS
4312 015752 012700 000200      MOV      R2,IS
4313 016756 170100      MOV      @200,R0      ;SET FD.
4314 016760 012737 016774 001236      LDFPS   R0
4315 016766 012737 017070 000304      MOV      @LLB2,@$TMP2
4316 016774 170767 000050      MOV      @LLB10,@$ERPVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
4317 017000 170205      LLB2:   NEGCC  LLBBFD      ;TEST INSTRUCTION.
4318 017002 012701 017050      STFPS   R5      ;GET FPS.
4319 017006 012702 000004      MOV      @LLBBFD,R1      ;CHECK RESULT.
4320 017012 005721      IS:     MOV      R4,R2
4321 017014 001043      TST     (R1)+
4322 017016 077203      BNE     LLB15      ;BRANCH IF INCORRECT.
4323 017020 022705 000204      SOB      R2,IS
4324 017024 001052      CMP     @204,R5      ;IS THE FPS CORRECT?
4325 017026 000457      BNE     LLB25      ;BRANCH IF INCORRECT.
4326 017026 000457      BR      LLBDONE
4327 017030 000127      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4328 017032 137475      LLBTP1: 127
4329 017034 147372      137475
4330 017036 117057      147372
4331 017040 000000      LLBTP2: 0
4332 017042 000000      0
4333 017044 000000      0
4334 017046 000000      0
4335 017050 177777      LLBBFD: -1
4336 017052 177777      -1
4337 017054 177777      -1
4338 017056 177777      -1
4339 017060 177777      LLBBF1: -1
4340 017062 177777      -1
4341 017064 177777      -1
4342 017066 177777      -1
4343 017070 011602      ;IF A TRAP TO 4 OCCURS COME HERE.
4344 017072 020227 016776      LLB10:  MOV      (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4345 017076 001405      CMP     R2,@LLB2+2
4346 017100 020227 017000      BEQ     IS      ;BRANCH IF YES.
4347 017104 001402      CMP     R2,@LLB2+4
4348 017106 000137 042620      BEQ     IS      ;BRANCH IF YES.
4349 017112 022626      JMP     @CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4350 017114 010237 001236      ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
4351 017120 104127      IS:     CMP     (SP)+,(SP)+
4352 017122 000421      MOV     R2,@$TMP2
4353 017124 012737 017040 001240      2S:     ERROR  127      ;ODD ADRES
4354 017132 012737 017030 001242      BR      LLBDONE      ;BUT FDSTX IN ST 771
4355 017140 012737 017050 001244      ;REPORT RESULT INCORRECT:
4356 017146 104130      LLB15:  MOV     @LLBTP2,@$TMP3
4357 017146 012737 017030 001242      MOV     @LLBTP1,@$TMP4
4358 017146 012737 017050 001244      MOV     @LLBBFD,@$TMP5
4359 017146 104130      IS:     ERROR  130      ;BAD DATA X11#0 ST 3127

```

4365 017150 000406
4366 017152 010537 001240
4368 017156 012737 000204 001244
4369 017164 104131
4370 017166
4371 017166 104412
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385 017170 000004
4386 017172
4387 017172 104413
4388 017174 012700 017272
4389 017200 012701 017322
4390 017204 012702 000010
4391 017210 012021
4392 017212 077202
4393 017214 012700 000200
4394 017220 170100
4395 017222 012737 017236 001236
4396 017230 012737 017342 000004
4397 017236 170677 000070
4398
4399
4400
4401 017242 170205
4402 017244 012701 017322
4403 017250 012702 000004
4404 017254 005721
4405 017256 001047
4406 017260 077203
4407 017262 022705 000204
4408 017266 001056
4409 017270 000463
4410
4411
4412 017272 000137
4413 017274 045607
4414 017276 101230
4415 017300 045607 017322 177777
4416 017306 177777 177777
4417 017312 000000
4418 017314 000000
4419 017316 000000
4420 017320 000000

```
BR LLBDONE  
:REPORT FPS INCORRECT:  
LLB25: MOV R5,2#STMP3  
MOV #204,2#STMP5  
IS: ERROR 131 ;FPSX  
LLBDONE:  
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
;THE USER TYPED CONTROL G?).  
:*****  
:TEST 31 NEG. ABSF AND TSTF SOURCE MODE 7, GR7, TEST  
:*****  
:THIS IS A TEST THE NEG. ABSF AND TSTF  
:SOURCE FLOWS. THE ABSD INSTRUCTION  
:IS USED TO TEST MODE 7  
:*****  
TST31: SCOPE  
MMB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #MMBTP1,R0 ;SET UP THE DATA BUFFER.  
MOV #MMBBF0,R1  
MOV #10,R2  
IS: MOV (R0)+,(R1)+  
SOB R2,IS  
MOV #200,R0 ;SET FD.  
LDFPS R0  
MOV #MMB2,2#STMP2  
MOV #MMB10,2#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.  
MMB2: ABSD 2#MMBBF1 ;TEST INSTRUCTION.  
STFPS R5 ;GET FPS.  
MOV #MMBBF0,R1 ;CHECK RESULT.  
IS: MOV #4,R2  
TST (R1)+  
BNE MMB15 ;BRANCH IF INCORRECT.  
SOB R2,IS  
CMP #204,R5 ;IS THE FPS CORRECT?  
BNE MMB25 ;BRANCH IF INCORRECT.  
BR MMBDONE  
:THESE ARE TEST DATA TABLES AND DATA BUFFER.  
MMBTP1: 137  
045607  
101230  
45607,MMBBF0,-1,-1,-1  
MMBTP2: 0  
0  
0  
0
```

017322 177777
017324 177777
017326 177777
017330 177777
017332 177777
017334 177777
017336 177777
017340 177777

MMBBFC: -1
-1
-1
-1
MMBBF1: -1
-1
-1
-1

017342 011602
017344 020227 017240
017350 001405
017352 020227 017242
017356 001402
017360 000137 042620
017364 022626
017366 010237 001236
017372 104132
017374 000421

; IF A TRAP TO 4 OCCURS COME HERE.
MMB10: MOV (SP), R2 ; SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2, #MMB2+2
BEQ 15 ; BRANCH IF YES.
CMP R2, #MMB2+4
BEQ 15 ; BRANCH IF YES.
JMP @#CPSUR ; OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
; REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
15: CMP (SP)+, (SP)+
MOV R2, @#STMP2
25: ERROR 132 ; ODD ADRES
BR MMBDONE ; BUT FDSTX IN ST 771

017376 012737 017312 001240
017404 012737 017272 001242
017412 012737 017322 001244
017420 104133
017422 000406

; REPORT RESULT INCORRECT:
MMB15: MOV #MMBTP2, @#STMP3
MOV #MMBTP1, @#STMP4
MOV #MMBBFO, @#STMP5
15: ERROR 133 ; BAD DATA X11*0 ST 3127
BR MMBDONE

017424 010537 001240
017430 012737 000204 001244
017436 104124

; REPORT FPS INCORRECT:
MMB25: MOV R5, @#STMP3
MOV #204, @#STMP5
15: ERROR 134 ; FPSX

017440
017440 104412

MMBDONE: RSETUP ; GO INITIALIZE THE FPS AND STACK; AND
; SEE IF THE USER HAS EXPRESSED
; THE DESIRE TO CHANGE THE SOFTWARE
; VIRTUAL CONSOLE SWITCH REGISTER (HAS
; THE USER TYPED CONTROL G?).

017442 000004

; TEST 32 SPECIAL DEST, MODE 0, TEST
; *
; *THIS IS A TEST OF THE NEG. ABSF AND TSTF DESTINATION FLOWS
; *MODE 0 USING THE NEG.0 INSTR.
; *

TST32: SCOPE

017444
017444 104413
017446 012700 000200
017452 170100
017454 012700 017542
017460 172410
017462 012737 017470 001236

NNB1: LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
MOV #200, R0 ; SET FD.
LDFPS R0
MOV #NNBTP1, R0 ; SET UP ACC.
LDD (R0), ACC
MOV #NNB2, @#STMP2

E07

```

4477 017470 170700 NNB2: NEG0 ACO ;TEST INSTRUCTION.
4478 017472 170205 STFPS R5 ;GET FPS.
4479 017474 012700 000200 MOV #200,R0 ;SET FD.
4480 017500 170100 LDFPS R0
4481 017502 012700 017562 MOV #NNBBFO,R0 ;GET THE RESULT.
4482 017506 174010 STD ACO,(R0)
4483 017510 012700 017562 MOV #NNBBFO,R0 ;IS THE RESULT CORRECT?
4484 017514 012701 017552 MOV #NNBTP2,R1
4485 017520 012702 000004 MOV #4,R2
4486 017524 022021 15: CMP (R0)+,(R1)+
4487 017526 031021 BNE NNB10 ;BRANCH IF INCORRECT.
4488 017530 077203 SOB R2,15
4489 017532 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4490 017536 001033 BNE NNB15 ;BRANCH IF INCORRECT.
4491 017540 000440 BR NNBDONE

```

;THESE ARE DATA TABLES AND A DATA BUFFER.

```

4492 017542 013572 NNBTP1: 013572
4493 017544 046013 46013
4494 017546 057246 57246
4495 017550 013570 013570
4496 017552 113572 NNBTP2: 113572
4497 017554 046013 46013
4498 017556 057246 57246
4499 017560 013570 013570
4500 017562 000000 NNBFFO: 0
4501 017564 000000 0
4502 017566 000000 0
4503 017570 000000 0

```

```

4504 017572 012737 017562 001240 ;REPORT RESULT INCORRECT:
4505 017600 012737 017552 001242 NNB10: MOV #NNBBFO,#STMP3
4506 017606 023737 017542 017562 MOV #NNBTP2,#STMP4
4507 017614 001002 CMP #NNBTP1,#NNBBFO
4508 017616 104135 15: BNE NNB11
4509 017620 000410 BR ERROR 135 ;E10*200X ST 336
4510 BR NNBDONE

```

```

4511 ;REPORT RESULT INCORRECT:
4512 NNB11:
4513 15: ERROR 136 ;BAD DATA NEGF
4514 BR NNBDONE

```

```

4515 ;REPORT FPS INCORRECT:
4516 NNB15: MOV R5,#STMP4
4517 017626 010537 001242 MOV #210,#STMP3
4518 017632 012737 000210 001240 15: ERROR 137 ;FPSX
4519 017640 104137

```

```

4520 NNBDONE:
4521 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4522 ;SEE IF THE USER HAS EXPRESSED
4523 ;THE DESIRE TO CHANGE THE SOFTWARE
4524 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4525 ;THE USER TYPED CONTROL G?).

```

::*****

F07

```

4533      : *TEST 33            SPECIAL DEST, MODE 1, TEST
4534      : *
4535      : *THIS IS A TEST OF THE NEG, ABS, AND TST DESTINATION FLOWS
4536      : *MODE 1 USING THE NEG, ABS INSTR.
4537      : *
4538      : *****
4539      017644    000004    ST33:    SCOPE
4540
4541      017646                    COB1:
4542      017646    104413            LPERR                    ;SET UP THE LOOP ON ERROR ADDRESS.
4543      017650    012701    0:7760    MOV        #00BTP1,R1            ;SET UP THE DATA BUFFER.
4544      017654    012700    017770    MOV        #00BTP2,R0
4545      017660    012707    000004    MOV        #4,R2
4546      017664    01202            1S:    MOV        (R0)+,(R1)+
4547      017666    077202            SOB        R2,1S
4548      017670    012710    017760    MOV        #00BTP1,R0
4549      017674    042710    100000    BIC        #100000,(R0)        ;MAKE OPERAND POSITIVE.
4550      017700    012737    017714    001236    MOV        #00B2,2#STMP2
4551      017706    012701    000200            MOV        #200,R1            ;SET FD.
4552      017712    170101            LDFPS     R1
4553
4554      017714    170710            COB2:    NEG        (R0)            ;TEST INSTRUCTION.
4555      017716    170205            STFPS     R5            ;GET FPS.
4556      017720    012701    017760    MOV        #00BTP1,R1            ;IS THE RESULT CORRECT.
4557      017724    012702    017770    MOV        #00BTP2,R2
4558      017730    012703    000004    MOV        #4,R3
4559      017734    022122            1S:    CMP        (R1)+,(R2)+
4560      017736    001020            BNE        00B10            ;BRANCH IF INCORRECT.
4561      017740    077303            SOB        R3,1S
4562      017742    022700    017760    CMP        #00BTP1,R0            ;IS R0 CORRECT.
4563      017746    001024            BNE        00B15            ;BRANCH IF INCORRECT.
4564      017750    022705    000210    CMP        #210,R5            ;IS THE FPS CORRECT?
4565      017754    001030            BNE        00B20            ;BRANCH IF INCORRECT.
4566      017756    000435            BR        00BDONE
4567
4568                                ; THESE ARE DATA TABLES AND A DATA BUFFER.
4569    017760    023245    00BTP1: 023245
4570    017762    026720                    26720
4571    017764    122324                    122324
4572    017766    052672                    52672
4573    017770    123245    00BTP2: 123245
4574    017772    026720                    26720
4575    017774    122324                    122324
4576    017776    052672                    52672
4577
4578                                ; REPORT RESULT INCORRECT:
4579    020000    012737    017760    001240    00B10: MOV        #00BTP1,2#STMP3
4580    020006    012737    017770    001242            MOV        #00BTP2,2#STMP4
4581    020014    104140            1S:    ERROR     140            ;BAD DATA
4582    020016    000415            BR        00BDONE
4583
4584                                ; REPORT R0 INCORRECT:
4585    020020    012737    017760    001240    00B15: MOV        #00BTP1,2#STMP3
4586    020026    010037    001242            MOV        R0,2#STMP4
4587    020032    104141            1S:    ERROR     141            ;SPEC DESTX
4588    020034    000406            BR        00BDONE            ;ROX
  
```

G07

4599
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

020036 012737 005210 001240
020044 010537 001242
020050 104142
020052
020052 104412

020054 000004
020056
020056 104413

020060 012701 020170
020064 012700 020200
020070 012702 000004
020074 012021
020076 077202
020100 012700 020170
020104 042710 100000
020110 012737 020124 001236
020116 012701 000200
020122 170101

020124 170720

020126 170205
020130 012701 020170
020134 012702 020200
020140 012703 000004
020144 022122
020146 001020
020150 077303
020152 022700 020200
020156 001024
020160 022705 000210
020164 001030
020166 000435

020170 023245
020172 026720
020174 122324
020176 052672
020200 123245
020202 026720

```
:REPORT FPS INCORRECT:  
00B20: MOV #210,2#STMP3  
MOV R5,2#STMP4  
1S: ERROR !42  
  
00BDONE:  
RSETUP  
;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
;THE USER TYPED CONTROL G?).  
:*****  
:TEST 34 SPECIAL DEST, MODE 2, TEST  
:*****  
:THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS  
:MODE 2 USING THE NEGD INSTR.  
:*****  
TST34: SCOPE  
PPB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #PPBTP1,R1 ;SET UP THE DATA BUFFER.  
MOV #PPBTP2,R0  
MOV #4,R2  
1S: MOV (R0)+,(R1)+  
SOB R2,1$  
MOV #PPBTP1,R0  
BIC #100000,(R0) ;MAKE OPERAND POSITIVE.  
MOV #PPB2,2#STMP2  
MOV #200,R1 ;SET FD.  
LDFPS R1  
  
PPB2: NEGD (R0)+ ;TEST INSTRUCTION.  
  
STFPS R5 ;GET FPS.  
MOV #PPBTP1,R1 ;IS THE RESULT CORRECT.  
MOV #PPBTP2,R2  
MOV #4,R3  
1S: CMP (R1)+,(R2)+  
BNE PPB10 ;BRANCH IF INCORRECT.  
SOB R3,1$  
CMP #PPBTP1+10,R0 ;IS R0 CORRECT.  
BNE PPB15 ;BRANCH IF INCORRECT.  
CMP #210,R5 ;IS THE FPS CORRECT?  
BNE PPB20 ;BRANCH IF INCORRECT.  
BR PPBDONE  
  
: THESE ARE DATA TABLES AND A DATA BUFFER.  
PPBTP1: 023245  
26720  
122324  
52672  
PPBTP2: 123245  
26720
```


H07

```

4645 020204 122324 122324
4646 020206 052672 52672
4647
4648
4649 020210 012737 020170 001240 :REPORT RESULT INCORRECT:
4650 020216 012737 020200 001242 PPB10: MOV #PPBTP1,2#STMP3
4651 020224 104143 1S: ERROR 143 ;BAD DATA
4652 020226 000415 BR PPBDONE
4653
4654
4655 020230 012737 020200 001240 :REPORT RD INCORRECT:
4656 020236 010037 001242 PPB15: MOV #PPBTP1+10,2#STMP3
4657 020242 104144 1S: ERROR 144 ;SPEC DESTX ROX
4658 020244 000406 BR PPBDONE
4659
4660
4661 020246 012737 000210 001240 :REPORT FPS INCORRECT:
4662 020254 010537 001242 PPB20: MOV #210,2#STMP3
4663 020260 104145 1S: ERROR 145
4664
4665 020262 PPBDONE:
4666 020262 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4667 ;SEE IF THE USER HAS EXPRESSED
4668 ;THE DESIRE TO CHANGE THE SOFTWARE
4669 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4670 ;THE USER TYPED CONTROL G?).
4671
4672 :*****
4673 :*TEST 35 SPECIAL DEST, MODE 4, TEST
4674 :*
4675 :*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4676 :*MODE 4 USING THE NEGD INSTR.
4677 :*
4678 :*****
4679 :*ST35: SCOPE
4680 QQB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4681 MOV #QQBTP1,R1 ;SET UP THE DATA BUFFER.
4682 MOV #QQBTP2,R0
4683 MOV #4,R2
4684 1S: MOV (R0)+,(R1)+
4685 SOB R2,1S
4686 MOV #QQBTP1+10,R0
4687 BIC #100000,-10(R0) ;MAKE OPERAND POSITIVE.
4688 MOV #QQB2,2#STMP2
4689 MOV #200,R1 ;SET FD.
4690 LDFPS R1
4691
4692 QQB2: NEGD -(R0) ;TEST INSTRUCTION.
4693
4694 STFPS R5 ;GET FPS.
4695 MOV #QQBTP1,R1 ;IS THE RESULT CORRECT.
4696 MOV #QQBTP2,R2
4697 MOV #4,R3
4698 1S: CMP (R1)+,(R2)+
4699 BNE QQB10 ;BRANCH IF INCORRECT.
4700 SOB R3,1S

```

```

4701 020364 022700 020402      CMP      #QOBTP1,R0      ;IS R0 CORRECT.
4702 020370 001030      BNE      QOB15         ;BRANCH IF INCORRECT.
4703 020372 022705 000210      CMP      #210,R5       ;IS THE FPS CORRECT?
4704 020376 001034      BNE      QOB20         ;BRANCH IF INCORRECT.
4705 020400 000441      BR       QOBDONE
4706
4707      ;THESE ARE DATA TABLES AND A DATA BUFFER.
4708 020402 023245      QOBTP1: 023245
4709 020404 026720      26720
4710 020406 122324      122324
4711 020410 052672      52672
4712 020412 177777 177777 177777      .WORD   -1,-1,-1,-1
4713 020420 177777
4714 020422 123245      QOBTP2: 123245
4715 020424 026720      26720
4716 020426 122324      122324
4717 020430 052672      52672
4718
4719      ;REPORT RESULT INCORRECT:
4720 020432 012737 020402 001240      QOB10:  MOV      #QOBTP1,@#STMP3
4721 020440 012737 020422 001242      MOV      #QOBTP2,@#STMP4
4722 020446 104146      IS:     ERROR   146      ;BAD DATA
4723 020450 000415      BR       QOBDONE
4724
4725      ;REPORT R0 INCORRECT:
4726 020452 012737 020402 001240      QOB15:  MOV      #QOBTP1,@#STMP3
4727 020460 010037 001242      MOV      R0,@#STMP4
4728 020464 104147      IS:     ERROR   147      ;SPEC DESTX R0X
4729 020466 000406      BR       QOBDONE
4730
4731      ;REPORT FPS INCORRECT:
4732
4733 020470 012737 000210 001240      QOB20:  MOV      #210,@#STMP3
4734 020476 010537 001242      MOV      R5,@#STMP4
4735 020502 104150      IS:     ERROR   150
4736
4737 020504      QOBDONE:
4738 020504 104412      PSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
4739      ;SEE IF THE USER HAS EXPRESSED
4740      ;THE DESIRE TO CHANGE THE SOFTWARE
4741      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4742      ;THE USER TYPED CONTROL G?).
4743
4744      ;*****
4745      ;*TEST 36      SPECIAL DEST, MODE 3, TEST
4746      ;*
4747      ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4748      ;*MODE 3 USING THE NEGD INSTR.
4749      ;*
4750      ;*****
4751 020506 000004      TST36:  SCOPE
4752
4753 020510      RRB1:
4754 020510 104413      LPERR
4755 020512 012701 020630      MOV      #RRBTP1,R1      ;SET UP THE LOOP ON ERROR ADDRESS.
4756 020516 012700 020640      MOV      #RRBTP2,R0      ;SET UP THE DATA BUFFER.

```

```

4757 020522 012702 000004
4758 020526 012021 1S: MOV #4,R2
4759 020530 077202 SOB (R0)+,(R1)+
4760 020532 012700 020650 MOV #RRBTP3,R0
4761 020536 012710 020630 MOV #RRBTP1,(R0)
4762 020542 042737 100000 020630 BIC #100000,#RRBTP1 ;MAKE THE OPERAND POSITIVE.
4763 020550 012737 020564 00123E MOV #RRB2,#STMP2
4764 020556 012701 000200 MOV #200,R1 ;SET FD.
4765 020562 170101 LDFPS R1
4766
4767 020564 170730 RRB2: NEG0 #2(R0)+ ;TEST INSTRUCTION.
4768
4769 020566 170205 STFPS R5 ;GET FPS.
4770 020570 012701 020630 MOV #RRBTP1,R1 ;IS THE RESULT CORRECT.
4771 020574 012702 020640 MOV #RRBTP2,R2
4772 020600 012703 000004 MOV #4,R3
4773 020604 022122 1S: CMP (R1)+,(R2)+
4774 020606 001021 BNE RRB10 ;BRANCH IF INCORRECT.
4775 020610 077303 SOB R3,1S
4776 020612 022700 020652 CMP #RRBTP3+2,R0 ;IS R0 CORRECT.
4777 020616 001025 BNE RRB15 ;BRANCH IF INCORRECT.
4778 020620 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4779 020624 001031 BNE RRB20 ;BRANCH IF I' CORRECT.
4780 020626 000436 BR RRBDONE
4781
4782 ;THESE ARE DATA TABLES AND A DATA BUFFER.
4783 020630 023245 RRBTP1: 023245
4784 020632 026720 26720
4785 020634 122324 122324
4786 020636 052672 52672
4787 020640 123245 RRBTP2: 123245
4788 020642 026720 26720
4789 020644 123324 123324
4790 020646 052672 52672
4791 020650 020630 RRBTP3: RRBTP1
4792
4793 ;REPORT RESULT INCORRECT:
4794 020652 012737 020630 001240 RRB10: MOV #RRBTP1,#STMP3
4795 020660 012737 020640 001242 MOV #RRBTP2,#STMP4
4796 020666 104150 1S: ERROR 150 ;BAD DATA
4797 020670 000415 BR RRBDONE
4798
4799 ;REPORT R0 INCORRECT:
4800 020672 012737 020652 001240 RRB15: MOV #RRBTP3+2,#STMP3
4801 020700 010037 001242 MOV R0,#STMP4
4802 020704 104152 1S: ERROR 152 ;SPEC DESTX ROX
4803 020706 000406 BR RRBDONE
4804
4805 ;REPORT FPS INCORRECT:
4806 020710 012737 000210 001240 RRB20: MOV #210,#STMP3
4807 020716 010537 001242 MOV R5,#STMP4
4808 020722 104153 1S: ERROR 153
4809
4810 RRBDONE:
4811 020724 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4812 ;SEE IF THE USER HAS EXPRESSED

```

K07

MAINDEC-11-FPP34-A
DFFPCA.P11

PDP 11 34 FPP DIAGNOSTIC
31-OCT-76 17:16

MACY11 27(1006) 31-OCT-76 17:35 PAGE 88
T36 SPECIAL DEST, MODE 3, TEST

;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824 020726 000004
4825 020730
4826 020730 104413
4827 020732 012701 021052
4828 020736 012700 021062
4829 020742 012702 000004
4830 020746 012021
4831 020750 077202
4832 020752 012700 021074
4833 020756 012760 021052 177776
4834 020764 042737 100000 021052
4835 020772 012737 021006 001236
4836 021000 012701 000200
4837 021004 170101
4838
4839 021006 170750
4840
4841 021010 170205
4842 021012 012701 021052
4843 021016 012702 021062
4844 021022 012703 000004
4845 021026 022122
4846 021030 001021
4847 021032 077303
4848 021034 022700 021072
4849 021040 001025
4850 021042 022705 000210
4851 021046 001031
4852 021050 000436
4853
4854
4855 021052 023245
4856 021054 026720
4857 021056 122324
4858 021060 052672
4859 021062 123245
4860 021064 026270
4861 021066 122324
4862 021070 052672
4863 021072 021052
4864
4865
4866 021074 012737 021052 001240
4867 021102 012737 021062 001242
4868 021110 104154

;TEST 37 SPECIAL DEST, MODE 5, TEST

;*THIS IS A TEST OF THE NEG, ABSF AND TSTF DESTINATION FLOWS
;*MODE 5 USING THE NEG, INSTR.

TST37: SCOPE
SSB1:

LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #SSBTP1,R1 ;SET UP THE DATA BUFFER.
MOV #SSBTP2,R0
MOV #4,R2
1\$: MOV (R0)+,(R1)+
SOB R2,1\$
MOV #SSBTP3+2,R0
MOV #SSBTP1,-2(R0)
BIC #100000,#SSBTP1 ;MAKE THE OPERAND POSITIVE.
MOV #SSB2,#STMP2
MOV #200,R1 ;SET FD.
LDFPS R1

SSB2: NEG @-(R0) ;TEST INSTRUCTION.

STFPS R5 ;GET FPS.
MOV #SSBTP1,R1 ;IS THE RESULT CORRECT.
MOV #SSBTP2,R2

1\$: MOV #4,R3
CMP (R1)+,(R2)+
BNE SSB10 ;BRANCH IF INCORRECT.
SOB R3,1\$
CMP #SSBTP3,R0 ;IS R0 CORRECT.
BNE SSB15 ;BRANCH IF INCORRECT.
CMP #210,R5 ;IS THE FPS CORRECT?
BNE SSB20 ;BRANCH IF INCORRECT.
BR SSB00NE

;THESE ARE DATA TABLES AND A DATA BUFFER.

SSBTP1: 023245
26720
122324
52672
SSBTP2: 123245
26270
122324
52672
SSBTP3: SSBTP1

;REPORT RESULT INCORRECT:
SSB10: MOV #SSBTP1,#STMP3
MOV #SSBTP2,#STMP4
1\$: ERROR 154 ;BAD DATA

```

4869 021112 000415 BR SSBDONE
4870
4871
4872 021114 012737 021072 001240 ;REPORT RO INCORRECT:
4873 021122 010037 001242 SSB15: MOV #SSBTP3,2*STMP3
4874 021126 104155 1S: MOV RO,2*STMP4
4875 021130 000406 1S: ERROR 155 ;SPEC DESTX ROX
4876 BR SSBDONE
4877
4878 021132 012737 000210 001240 ;REPORT FPS INCORRECT:
4879 021140 010537 001242 SSB20: MOV #210,2*STMP3
4880 021144 104156 1S: MOV R5,2*STMP4
4881
4882 021146 SSBDONE:
4883 021146 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4884 ;SEE IF THE USER HAS EXPRESSED
4885 ;THE DESIRE TO CHANGE THE SOFTWARE
4886 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4887 ;THE USER TYPED CONTROL G?).
4888
4889 ;*****
4890 ;*TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST
4891 ;*
4892 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4893 ;*MODE 2 USING THE NEGF INSTR.
4894 ;*
4895 ;*****
4896 TST40: SCOPE
4897 TTBI:
4898 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4899 MOV #TTBTP1,R1 ;SET UP THE DATA BUFFER.
4900 MOV #TTBTP2,R0
4901 MOV #4,R2
4902 1S: MOV (R0)+,(R1)+
4903 SOB R2,1S
4904 MOV #TTBTP1,R0
4905 BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
4906 MOV #TTB2,2*STMP2
4907 MOV #000,R1 ;SET FD.
4908 LDFPS R1
4909 TTBI: NEGF (R0)+ ;TEST INSTRUCTION.
4910
4911 STFPS R5 ;GET FPS.
4912 MOV #TTBTP1,R1 ;IS THE RESULT CORRECT.
4913 MOV #TTBTP2,R2
4914 MOV #4,R3
4915 1S: CMP (R1)+,(R2)+
4916 BNE TTB10 ;BRANCH IF INCORRECT.
4917 SOB R3,1S
4918 CMP #TTBTP1+4,R0 ;IS RO CORRECT.
4919 BNE TTB15 ;BRANCH IF INCORRECT.
4920 CMP #010,R5 ;IS THE FPS CORRECT?
4921 BNE TTB20 ;BRANCH IF INCORRECT.
4922 BR TTBDONE
4923
4924 ;THESE ARE DATA TABLES AND A DATA BUFFER.

```

M07

```

4925 021264 023245 TTBT1: 023245
4926 021266 026720 26720
4927 021270 122324 122324
4928 021272 052672 52672
4929 021274 123245 TTBT2: 123245
4930 021276 026720 26720
4931 021300 122324 122324
4932 021302 052672 52672
4933
4934
4935 021304 012737 021264 001240 ;REPORT RESULT INCORRECT:
4936 021312 012737 021274 001242 TTBT10: MOV #TTBT1,2#STMP3
4937 021320 104150 15: MOV #TTBT2,2#STMP4 ;BAD DATA
4938 021322 000415 BR 150 TTBDONE
4939
4940
4941 021324 012737 021270 001240 ;REPORT RO INCORRECT:
4942 021332 010037 001242 TTBT15: MOV #TTBT1+4,2#STMP3
4943 021336 104160 15: MOV RO,2#STMP4 ;SPEC DESTX ROX
4944 021340 000406 BR 160 TTBDONE
4945
4946
4947 021342 012737 000010 001240 ;REPORT FPS INCORRECT:
4948 021350 010537 001242 TTBT20: MOV #010,2#STMP3
4949 021354 104161 15: MOV R5,2#STMP4
4950
4951 021356 TTBDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4952 021356 104412 ;SEE IF THE USER HAS EXPRESSED
4953 ;THE DESIRE TO CHANGE THE SOFTWARE
4954 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4955 ;THE USER TYPED CONTROL G?).
4956
4957 ;*****
4958 ;*TEST 41 SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST ;
4959 ;*
4960 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4961 ;*MODE 2(IMMEDIATE) USING THE NEG0 INSTR.
4962 ;*
4963 ;*****
4964 021360 000004 TST41: SCOPE
4965 021362 UUB1:
4966 021362 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4967 021364 012700 021510 MOV #UUBTP2,R0
4968 021370 012701 021436 MOV #UUBTP1,R1 ;SET UP THE DATA BUFFER.
4969 021374 012702 000004 MOV #4,R2
4970 021400 012021 15: MOV (R0)+,(R1)+
4971 021402 077202 SOB R2,15
4972 021404 012700 021436 MOV #UUBTP1,R0
4973 021410 042737 100000 021436 BIC #100000,2#UUBTP1 ;MAKE THE OPERAND POSITIVE.
4974 021416 012737 021434 001236 MOV #UUB2,2#STMP2
4975 021424 012701 000200 MOV #200,R1 ;SET FD.
4976 021430 170101 LDFPS R1
4977 021432 005001 CLR R1
4978
4979 021434 170727 UUB2: NEG0 (R7)+ ;TEST INSTRUCTION.
4980 021436 005201 005201 005201 UUBTP1: 5201,5201,5201,5201

```

N07

```

4981 021444 005201
4982
4983 021446 170205
4984 021450 012703 021436
4985 021454 012702 021510
4986 021460 012704 000004
4987 021464 022322
4988 021466 001014
4989 021470 077403
4990 021472 022701 000003
4991 021476 0C1027
4992 021500 022705 00C210
4993 021504 0C1015
4994 021506 000436
4995
4996
4997 021510 105201
4998 021512 005201
4999 021514 005201
5000 021516 005201
5001
5002
5003 021520 012737 021436 001240
5004 021526 012737 021510 001242
5005 021534 104162
5006 021536 000422
5007
5008
5009 021540 012737 000210 001240
5010 021546 010537 0C1242
5011 021552 104163
5012 021554 000413
5013
5014
5015 021556 162701 000003
5016 021562 006301
5017 021564 012702 021440
5018 021570 010237 001240
5019 021574 16C102
5020 021576 010237 001242
5021 021602 104164
5022
5023 021604
5024 021604 104412
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036 021606 000004

;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
STFPS R5 ;GET FPS.
MOV #UUBTP1,R3 ;IS THE RESULT CORRECT.
MOV #UUBTP2,R2
MOV #4,R4
1$: CMP (R3)+,(R2)+
BNE UUB10 ;BRANCH IF INCORRECT.
SOB R4,1$
CMP #3,R1 ;WAS R1 INCREMENTED CORRECTLY.
BNE UUB15 ;BRANCH IF INCORRECT.
CMP #210,R5 ;IS THE FPS CORRECT?
BNE UUB20 ;BRANCH IF INCORRECT.
BR ULBDONE

;THESE ARE DATA TABLE.
UUBTP2: 105201
5201
5201
5201

;REPORT RESULT INCORRECT:
UUB10: MOV #UUBTP1,2#STMP3
MOV #UUBTP2,2#STMP4
1$: ERROR 162 ;BAD DATA
BR UUBDONE

;REPORT FPS INCORRECT:
UUB20: MOV #210,2#STMP3
MOV R5,2#STMP4
1$: ERROR 163 ;FPS
BR UUBDCNE

;REPORT PC INCORRECTLY INCREMENTED DURING EXECUTION.
UUB15: SUB #3,R1
ASL R1
MOV #UUBTP1+2,R2
MOV R2,2#STMP3
SUB R1,R2
MOV R2,2#STMP4
1$: ERROR 164 ;PC BAD CONSTAND B GR7X

UUBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

;*****
;*TEST 42 SPECIAL DEST, MODE 6, TEST
;*
;*THIS IS A TEST OF THE NEGf ABSF AND TSTF DESTINATION FLOWS
;*MODE 6 USING THE NEGd INSTR.
;*
;*****
TST42: SCOPE

```



```

5073 021610 012701 021734
5074 021610 012701 021734
5075 021610 012701 021734
5076 021616 012701 021734
5077 021616 012701 021734
5078 021616 012701 021734
5079 021616 012701 021734
5080 021616 012701 021734
5081 021616 012701 021734
5082 021616 012701 021734
5083 021616 012701 021734
5084 021616 012701 021734
5085 021616 012701 021734
5086 021616 012701 021734
5087 021616 012701 021734
5088 021616 012701 021734
5089 021616 012701 021734
5090 021616 012701 021734
5091 021616 012701 021734
5092 021616 012701 021734
5093 021616 012701 021734
5094 021616 012701 021734
5095 021616 012701 021734
5096 021616 012701 021734
5097 021616 012701 021734
5098 021616 012701 021734
5099 021616 012701 021734
5100 021616 012701 021734
5101 021616 012701 021734
5102 021616 012701 021734
5103 021616 012701 021734
5104 021616 012701 021734
5105 021616 012701 021734
5106 021616 012701 021734
5107 021616 012701 021734
5108 021616 012701 021734
5109 021616 012701 021734
5110 021616 012701 021734
5111 021616 012701 021734
5112 021616 012701 021734
5113 021616 012701 021734
5114 021616 012701 021734
5115 021616 012701 021734
5116 021616 012701 021734
5117 021616 012701 021734
5118 021616 012701 021734
5119 021616 012701 021734
5120 021616 012701 021734
5121 021616 012701 021734
5122 021616 012701 021734
5123 021616 012701 021734
5124 021616 012701 021734
5125 021616 012701 021734
5126 021616 012701 021734
5127 021616 012701 021734
5128 021616 012701 021734
5129 021616 012701 021734
5130 021616 012701 021734
5131 021616 012701 021734
5132 021616 012701 021734
5133 021616 012701 021734
5134 021616 012701 021734
5135 021616 012701 021734
5136 021616 012701 021734
5137 021616 012701 021734
5138 021616 012701 021734
5139 021616 012701 021734
5140 021616 012701 021734
5141 021616 012701 021734
5142 021616 012701 021734
5143 021616 012701 021734
5144 021616 012701 021734
5145 021616 012701 021734
5146 021616 012701 021734
5147 021616 012701 021734
5148 021616 012701 021734
5149 021616 012701 021734
5150 021616 012701 021734
5151 021616 012701 021734
5152 021616 012701 021734
5153 021616 012701 021734
5154 021616 012701 021734
5155 021616 012701 021734
5156 021616 012701 021734
5157 021616 012701 021734
5158 021616 012701 021734
5159 021616 012701 021734
5160 021616 012701 021734
5161 021616 012701 021734
5162 021616 012701 021734
5163 021616 012701 021734
5164 021616 012701 021734
5165 021616 012701 021734
5166 021616 012701 021734
5167 021616 012701 021734
5168 021616 012701 021734
5169 021616 012701 021734
5170 021616 012701 021734
5171 021616 012701 021734
5172 021616 012701 021734
5173 021616 012701 021734
5174 021616 012701 021734
5175 021616 012701 021734
5176 021616 012701 021734
5177 021616 012701 021734
5178 021616 012701 021734
5179 021616 012701 021734
5180 021616 012701 021734
5181 021616 012701 021734
5182 021616 012701 021734
5183 021616 012701 021734
5184 021616 012701 021734
5185 021616 012701 021734
5186 021616 012701 021734
5187 021616 012701 021734
5188 021616 012701 021734
5189 021616 012701 021734
5190 021616 012701 021734
5191 021616 012701 021734
5192 021616 012701 021734
5193 021616 012701 021734
5194 021616 012701 021734
5195 021616 012701 021734
5196 021616 012701 021734
5197 021616 012701 021734
5198 021616 012701 021734
5199 021616 012701 021734
5200 021616 012701 021734

```

```

XXB1:  LPERR                                ;SET UP THE LOOP ON ERROR ADDRESS.
        MOV  #XXBTP1,R1                    ;SET UP THE DATA BUFFER.
        MOV  #XYBTP2,R0
        MOV  #4,R2
IS:     MOV  (R0)+,(R1)+
        SOB  R2,IS
        MOV  #XXBTP1-5201,R0
        BIC  #100000,#XXBTP1;MAKE OPERAND POSITIVE.
        MOV  #XXB2,#STMP2
        MOV  #200,R1                        ;SET FD.
        LDFPS R1

XXB2:   CLR  R1
        NEG  #5201(R0)                      ;TEST INSTRUCTION.

        STFPS R5                            ;GET FPS.
        TEST R1
        BNE  XXB25                          ;WAS THE PC CORRECT AFTER EXECUTION?
        MOV  #XXBTP1,R1                      ;IS THE RESULT CORRECT.
        MOV  #XXBTP2,R2
        MOV  #4,R3
IS:     CMP  (R1)+,(R2)+
        BNE  XXB10                          ;BRANCH IF INCORRECT.
        SOB  R3,IS
        CMP  #XXBTP1-5201,R0                ;IS R0 CORRECT.
        BNE  XXB15                          ;BRANCH IF INCORRECT.
        CMP  #210,R5                        ;IS THE FPS CORRECT?
        BNE  XXB20                          ;BRANCH IF INCORRECT.
        BR   XXBDONE

;THESE ARE DATA TABLES AND A DATA BUFFER.
XXBTP1: 023245
        26720
        122324
        52672
XXBTP2: 123245
        26720
        122324
        52672

;REPORT PC INCORRECT AFTER EXECUTION.
XXB25:  MOV  #XXB2+2,#STMP4
        MOV  #XXB2+4,#STMP3
IS:     ERROR 215                          ;PC NOT INCREMENTED BY 2.
        BR   XXBDONE

;REPORT RESULT INCORRECT:
XXB10:  MOV  #XXBTP1,#STMP3
        MOV  #XXBTP2,#STMP4
IS:     ERROR 216                          ;BAD DATA
        BR   XXBDONE

;REPORT R0 INCORRECT:
XXB15:  MOV  #XXBTP1-5201,#STMP3

```

022032 010037 001242
022036 104217
022030 000408

022032 012737 000210 001242
022040 010537 001242
022044 104220

022046
022046 104412

022050 000004

022052
022052 104413
022054 012701 022204
022060 012700 022214
022064 012702 000004
022070 012021
022072 077202
022074 012700 015023
022100 012760 022204 005201
022106 042737 100000 022204
022114 012737 022132 001236
022122 012701 000200
022126 170101

022130 005001
022132 170770 005201

022136 170205
022140 005701
022142 001031
022144 012701 022204
022150 012702 022214
022154 012703 000004
022160 022122
022162 001031
022164 077303
022166 022700 015023
022172 001035
022174 022705 000210
022200 001041

18: MOV RC, #STMP4
ERROR 217 ;SPEC DESTX RCX
BR XXBDONE

:REPORT FPS INCORRECT:
XXB20: MOV #210, #STMP3
MOV RS, #STMP4
18: ERROR 220

XXBDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

:*****
:TEST 43 SPECIAL DEST. MODE 7, TEST
:*****
:THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
:MODE 7 USING THE NEG0 INSTR.
:*****
TST43: SCOPE

YYB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #YYBTP1, R1 ;SET UP THE DATA BUFFER.
MOV #YYBTP2, R0
MOV #4, R2
18: MOV (R0)+, (R1)+
SOB R2, 18
MOV #YYBTP3-5201, R0
MOV #YYBTP1, 5201(R0)
BIC #100000, #YYBTP1 ;MAKE THE OPERAND POSITIVE.
MOV #YYB2, #STMP2
MOV #200, R1 ;SET FD.
LDFPS R1

YYB2: CLR R1
NEG0 #5201(R0) ;TEST INSTRUCTION.

STFPS R5 ;GET FPS.
TST R1 ;WAS THE PC CORRECT AFTER EXECUTION?
BNE YYB25
MOV #YYBTP1, R1 ;IS THE RESULT CORRECT.
MOV #YYBTP2, R2
MOV #4, R3
18: CMP (R1)+, (R2)+
BNE YYB10 ;BRANCH IF INCORRECT.
SOB R3, 18
CMP #YYBTP3-5201, R0 ;IS R0 CORRECT.
BNE YYB15 ;BRANCH IF INCORRECT.
CMP #210, RS ;IS THE FPS CORRECT?
BNE YYB20 ;BRANCH IF INCORRECT.

5149 022202 000446
5150
5151
5152 022204 023245
5153 022206 026720
5154 022210 123324
5155 022212 052672
5156 022214 123245
5157 022216 026720
5158 022220 123324
5159 022222 052672
5160 022224 022204
5161
5162
5163 022226 016737 177702 001242
5164 022234 016737 177676 00124C
5165 022242 104221
5166 022244 000425
5167
5168
5169 022246 012737 022204 001240
5170 022254 012737 022214 001242
5171 022262 104222
5172 022264 000415
5173
5174
5175 022266 012737 015023 001243
5176 022274 010037 001242
5177 022300 104223
5178 022302 000406
5179
5180
5181 022304 012737 000210 001240
5182 022312 010537 001242
5183 022316 104224
5184
5185 022320
5186 022320 104412
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197 022322 000004
5198
5199 022324
5200 022324 104413
5201 022326 004767 000634
5202 022332 000000
5203 022334 016341
5204 022336 055772

BR YYBDONE
: THESE ARE DATA TABLES AND A DATA BUFFER.
YYBTP1: 023245
26720
123324
52672
YYBTP2: 123245
26720
123324
52672
YYBTP3: YYBTP1
: REPORT PC INCORRECT AFTER EXECUTION.
YYB25: MOV YYB2+2, 2*STMP4
MOV YYB2+4, 2*STMP3
IS: ERROR 221 ; PC NOT INCREMENTED BY 2.
BR YYBDONE
: REPORT RESULT INCORRECT:
YYB10: MOV *YYBTP1, 2*STMP3
MOV *YYBTP2, 2*STMP4
IS: ERROR 222 ; BAD DATA
BR YYBDONE
: REPORT R0 INCORRECT:
YYB15: MOV *YYBTP3-5201, 2*STMP3
MOV R0, 2*STMP4
IS: ERROR 223 ; SPEC DESTX R0X
BR YYBDONE
: REPORT FPS INCORRECT:
YYB20: MOV *210, 2*STMP3
MOV R5, 2*STMP4
IS: ERROR 224
YYBDONE:
RSETUP ; GO INITIALIZE THE FPS AND STACK; AND
; SEE IF THE USER HAS EXPRESSED
; THE DESIRE TO CHANGE THE SOFTWARE
; VIRTUAL CONSOLE SWITCH REGISTER (HAS
; THE USER TYPED CONTROL G?).
: *****
: *TEST 44 NEG0, ABS0 AND TSTD TEST
: *
: *THIS IS A TEST OF THE NEG0 ABS0 AND TSTD INSTRUCTIONS.
: *
: *****
TST44: SCOPE
: TEST NEG0 WITH POS NONZERO OPERAND
MWB1:
LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
JSR PC, NATSUB
IS: 0 ; FLAG=NEG0.
2S: 16341 ; OPERAND.
55772

E08

MAINDEC-11-FP034-A
OFFPCA.P11 31-OCT-76 17:16

PDP 11 34 FPP DIAGNOSTIC
T44

MACY11 27(1006) 31-OCT-76 17:35 PAGE 95
NEGD. ABSD AND TSTD TEST

5205 022340 021133
5206 022342 055447
5207 022344 116341
5208 022346 055772
5209 022350 021133
5210 022352 055447
5211 022354 016341
5212 022356 055772
5213 022360 021133
5214 022362 055447
5215 022364 000207
5216 022366 000210
5217 022370 000200
5218 022372 177777
5219 022374 104200
5220 022376 000401
5221 022400 104201
5222 022402
5223
5224 022402
5225 022402 104413
5226 022404 004767 000556
5227 022410 000000
5228 022412 152525
5229 022414 053545
5230 022416 055565
5231 022420 057505
5232 022422 052525
5233 022424 053545
5234 022426 055565
5235 022430 057505
5236 022432 152525
5237 022434 053545
5238 022436 055565
5239 022440 057505
5240 022442 000217
5241 022444 000200
5242 022446 000210
5243 022450 177777
5244 022452 104200
5245 022454 000401
5246 022456 104202
5247 022460
5248
5249 022460
5250 022460 104413
5251 022462 004767 000500
5252 022466 000001
5253 022470 060705
5254 022472 124735
5255 022474 060124
5256 022476 073560
5257 022500 060705
5258 022502 124735
5259 022504 060124
5260 022506 073560

21133
55447
35: 116341 ;RESULT.
55772
21133
55447
45: 16341 ;ERROR RES.
55772
21133
55447
55: 207 ;FPS BEFORE EXECUTION.
210 ;FPS AFTER EXECUTION.
200 ;ERROR FPS.
-1 ;FEC
65: ERROR 200 ;E10<---E10*200X ST 336
BR 75
ERROR 201 ;BUT ENBT ST 336X WENT TO 053 INTO 453
75:
;TEST NEGD WITH NEG OPERAND.
WMB2:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,NATSUB
15: 0 ;FLAG=NEGD.
25: 152525 ;OPERAND.
53545
55565
35: 57505 ;RESULT.
52525
53545
55565
45: 57505 ;ERROR RES.
152525
53545
55565
57505
55: 217 ;FPS BEFORE EXECUTION.
200 ;FPS AFTER EXECUTION.
210 ;ERROR FPS.
-1 ;FEC
65: ERROR 200 ;E10<---E10*200X S336
BR 75
ERROR 202 ;BUT ENBT X ST336 TO 453 INTO 053
75:
;TEST ABSD WITH POSITIVE OPERAND
WMB3:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,NATSUB
15: 1 ;FLAG=ABSD.
25: 60705 ;OPERAND.
124735
60124
73560
35: 60705 ;RESULT.
124735
60124
73560

F08

```

5261 022510 160705 4S: 160705 ;ERROR RES.
5262 022512 124735 124735
5263 022514 060124 60124
5264 022516 073560 73560
5265 022520 000217 5S: 217 ;FPS BEFORE EXECUTION.
5266 022522 000200 200 ;FPS AFTER EXECUTION.
5267 022524 000210 210 ;ERROR FPS.
5268 022526 177777 -1 ;EITHER BUT OP18
5269 022530 104203 6S: ERROR 203 ;BUT ST 055 TO 336 INTO 335
5270 022532 000401 BR 7S
5271 022534 104203 ERROR 203 ;OR BUT ENBT ST 335 TO 452 INTO 052
5272 022536
5273
5274 022536 ;TEST ABSD WITH NEG. OPERAND
5275 022536 104413 WWB4: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5276 022540 004767 000422 JSR PC,NATSUB
5277 022544 000001 1S: 1 ;FLAG=ABSD.
5278 022546 154345 2S: 154345 ;OPERAND.
5279 022550 076567 76567
5280 022552 032123 32123
5281 022554 043234 43234
5282 022556 054345 3S: 54345 ;RESULT.
5283 022560 076567 76567
5284 022562 032123 32123
5285 022564 043234 43234
5286 022566 154345 4S: 154345 ;ERROR RES.
5287 022570 076567 76567
5288 022572 032123 32123
5289 022574 043234 43234
5290 022576 000217 5S: 217 ;FPS BEFORE EXECUTION.
5291 022600 000200 200 ;FPS AFTER EXECUTION.
5292 022602 177777 -1 ;ERROR FPS.
5293 022604 177777 -1
5294 022606 104204 6S: ERROR 204 ;E10*E10*200X ST 452
5295 022610 000401 BR 7S
5296 022612 104171 ERROR 171
5297 022614
5298
5299 ;TEST WITH POSITIVE OP
5300 022614 104413 WWB5: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5301 022616 004767 000344 JSR PC,NATSUB
5302 022622 000002 1S: 2 ;FLAG=TSTD.
5303 022624 012321 2S: 12321 ;OPERAND.
5304 022626 045654 45654
5305 022630 070107 70107
5306 022632 034543 34543
5307 022634 012321 3S: 12321 ;RESULT.
5308 022636 045654 45654
5309 022640 070107 70107
5310 022642 034543 34543
5311 022644 112321 4S: 112321 ;ERROR RES.
5312 022646 045654 45654
5313 022650 070107 70107
5314 022652 034543 34543
5315 022654 000217 5S: 217 ;FPS BEFORE EXECUTION.
5316 022656 000200 200 ;FPS AFTER EXECUTION.

```

```

5317 022660 000210          210          ;ERROR FPS.
5318 022662 177777          -1
5319 022664 104205      6S:  ERROR    205          ;BUT (OP1B) X ST044 TO 336 INTO 334
5320 022666 000401          BR          7S
5321 022670 104206          ERROR    206          ;BUT ENBT ST 334 TO 453 INTO 053
5322 022672
5323
5324 022672          ;TEST TSTD WITH NEG OP
5325 022672 104413      WWB6:          ;SET UP THE LOOP ON ERROR ADDRESS.
5326 022674 004767      000266      LPERR
5327 022700 000002          JSR      PC,NATSUB
5328 022702 123765      1S:  2          ;FLAG=TSTD.
5329 022704 023407      2S:  123765        ;OPERAND.
5330 022706 034510          23407
5331 022710 045621          34510
5332 022712 177765      3S:  45621
5333 022714 023407          123765        ;RESULT.
5334 022716 034510          23407
5335 022720 045621          34510
5336 022722 023765      4S:  45621        ;ERROR RES.
5337 022724 023407          23765
5338 022726 034510          23407
5339 022730 045621          34510
5340 022732 000207      5S:  45621
5341 022734 000210          207          ;FPS BEFORE EXECUTION.
5342 022736 000200          210          ;FPS AFTER EXECUTION.
5343 022740 177777          200          ;ERROR FPS.
5344 022742 104207      6S:  -1
5345 022744 000401          ERROR    207          ;BUT OPB1 ST 055 TO 335 INTO 334
5346 022746 104210          BR          7S
5347 022750          ERROR    210          ;BUT ENBT ST 334 TO 053 INTO 453
5348
5349 022750          ;TEST TSTD 0 OP
5350 022750 104413      WWB7:          ;SET UP THE LOOP ON ERROR ADDRESS.
5351 022752 004767      000210      LPERR
5352 022756 000002          JSR      PC,NATSUB
5353 022760 000175      1S:  2          ;FLAG=TSTD.
5354 022762 176737      2S:  175          ;OPERAND.
5355 022764 071727          176737
5356 022766 037574          71727
5357 022770 000175      3S:  37574
5358 022772 176737          175          ;RESULT.
5359 022774 071727          176737
5360 022776 037574          71727
5361 023000 000000      4S:  37574
5362 023002 000000          0          ;ERROR RES.
5363 023004 000000          0
5364 023006 000000          0
5365 023010 000200      5S:  0
5366 023012 000204          200          ;FPS BEFORE EXECUTION.
5367 023014 000214          204          ;FPS AFTER EXECUTION.
5368 023016 177777          214          ;ERROR FPS.
5369 023020 104211      6S:  -1
5370 023022 000401          ERROR    211          ;BUT OP1B ST 255 TO 311 OR 312 INTO 310
5371 023024 104212          BR          7S
5372 023026          ERROR    212          ;BUT ENBT ST 310 TO 402 INTO 002

```

H08

MAINDEC-11-FPP34-A
OFFPCA.P11

31-OCT-76 17:16

PDP 11 34 FPP DIAGNOSTIC
T44

MACY11 27(1006)
NEGD. ABSD AND TSTD TEST

31-OCT-76 17:35 PAGE 98

```

5373
5374 023026
5375 023026 104413
5376 023030 004767 000132
5377 023034 000002
5378 023036 100123
5379 023040 021012
5380 023042 034565
5381 023044 043210
5382 023046 100123
5383 023050 021012
5384 023052 034565
5385 023054 043210
5386 023056 000000
5387 023060 000000
5388 023062 000000
5389 023064 000000
5390 023066 040203
5391 023070 040214
5392 023072 140214
5393 023074 177777
5394 023076 104211
5395 023100 000401
5396 023102 104213
5397 023104
5398
5399 023104
5400 023104 104413
5401 023106 004767 000054
5402 023112 000002
5403 023114 100137
5404 023116 024613
5405 023120 057024
5406 023122 060137
5407 023124 100137
5408 023126 024613
5409 023130 057024
5410 023132 060137
5411 023134 000000
5412 023136 000000
5413 023140 000000
5414 023142 000000
5415 023144 044200
5416 023146 144214
5417 023150 044214
5418 023152 000014
5419 023154 104211
5420 023156 000401
5421 023160 104214
5422 023162
5423 023162 000167 000414
5424
5425
5426
5427
5428

```

:TEST TSTD -0 OP FIUV=0

WWB8:

```

LPERR
JSR PC,NATSUB
15: 2
25: 100123
21012
34565
43210
35: 100123
21012
34565
43210
45: 0
0
0
0
55: 40203
040214
140214
-1
65: ERROR 211
BR 75
ERROR 213
75:

```

;SET UP THE LOOP ON ERROR ADDRESS.

;FLAG=TSTD.
;OPERAND.

;RESULT.

;ERROR RES.

;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ERROR FPS.

;+

;BUT FIUV ST 257 TO 355 INTO 255

:TEST TSTD -0 OP FIUV=1

WWB9:

```

LPERR
JSR PC,NATSUB
15: 2
25: 100137
24613
57024
60137
35: 100137
24613
57024
60137
45: 0
0
0
0
55: 44200
144214
044214
14
65: ERROR 211
BR 75
ERROR 214
75:
JMP WWBDONE

```

;SET UP THE LOOP ON ERROR ADDRESS.

;FLAG=TSTD.
;OPERAND.

;RESULT.

;ERROR RES.

;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ERROR FPS.

;+

;BUT FIUV ST 257 TO 255 INTO 355

```

;THIS SUBROUTINE, NATSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
;THE EITHER A TSTD, AN ABSD OR A NEGD INSTRUCTION AND CHECK THE RESULTS. A CALL
;TO IT IS MADE THUS:
;

```

5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484

023166 012601
023170 010102
023172 062702 000002
023176 012703 023570
023202 012704 000004
023206 012223
023210 077402
023212 016100 000032
023216 170100
023220 012700 023570
023224 011102
023226 006302
023230 006302
023232 012703 023246
023236 060203
023240 010337 001236
023244 000113
023246 170710
023250 000403
023252 170610
023254 000401
023256 170510
023260 170204

```

JSR      PC,2#NATSUB
FLAG:    .WORD    X           ;INSTRUCTION TYPE FLAG.
ACARG:   .WORD    X,X,X,X    ;OPERAND
RES:     .WORD    X,X,X,X    ;EXPECTED RESULT
ERRRES:  .WORD    X,X,X,X    ;ERROR RESULT
FPSB:    .WORD    X           ;FPS BEFORE EXECUTION
FPSA:    .WORD    X           ;FPS AFTER EXECUTION
FEC:     .WORD    X           ;EXPECTED FEC
ERRFPS:  .WORD    X           ;ERROR FPS.
ERR1:    ERROR    X           ;DATA ERROR.
          BR      CONT
ERR2:    ERROR    X           ;FPS ERROR.
CONT:
          ;RETURN ADDRESS

;THE OPERAND IS SET UP IN NATBF1. THEN
;THE EITHER THE TSTD, NEG0 OR ABS0 INSTRUCTION IS EXECUTED.
;NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
;IS TO BE EXECUTED: 0 = NEG0, 1 = ABS0, 2 = TSTD.
;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
;COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL
;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
;TO THE ERROR CALL AT ERR2, OTHERWISE NATSUB ITSELF
;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
;INSTRUCTION IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
;ANTICIPATED FAILING DATA PATTERN, ERRRES. IF THE FAILURE IN
;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRRES THEN NATSUB
;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

NATSUB:  MOV      (SP)+,R1           ;GET A POINTER TO THE ARGUMENTS.
          MOV      R1,R2           ;COPY THE OPERAND.
          ADD      #2,R2
          MOV      #NATBF1,R3
          MOV      #4,R4
1$:      MOV      (R2)+,(R3)+
          SOB      R4,1$
          MOV      32(R1),R0       ;LOAD THE FPS.
          LDFPS   R0
          MOV      #NATBF1,R0       ;SET UP THE OPERAND ADDRESS.
          MOV      (R1),R2         ;GET THE FLAG TO DETERMINE WHICH
          ASL      R2              ;INSTRUCTION TO EXECUTE.
          ASL      R2              ;0 = NEG0, 1 = ABS0, 2 = TSTD
          MOV      #NATINS,R3
          ADD      R2,R3
          MOV      R3,2#STMP2
          JMP      (R3)           ;GO EXECUTE THE INSTRUCTION.
NATINS:  NEG0   (R0)
          BR      2$
          ABS0  (R0)
          BR      2$
          TSTD  (R0)
2$:      STFPS   R4           ;GET THE FPS.

```


JOB

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 100
 DFFPCA.P11 31-OCT-76 17:16 T44 NEG. ABSD AND TSTD TEST

5485	023262	170305		STST	R5	;GET THE FEC.
5486	023264	010102		MOV	R1,R2	
5487	023266	062702	000002	ADD	#2,R2	
5488	023272	010237	001240	MOV	R2,#STMP3	
5489	023276	062702	000010	ADD	#10,R2	
5490	023302	010237	001244	MOV	R2,#STMP5	
5491	023306	012737	023570	001242	MOV	#NATBF1,#STMP4
5492	023314	010437	001250	MOV	R4,#STMP7	
5493	023320	016137	000034	001252	MOV	34(R1),#STMP10
5494	023326	010100		MOV	R1,R0	;WAS THE RESULT CORRECT?
5495	023330	062700	000012	ADD	#12,R0	
5496	023334	012702	023570	MOV	#NATBF1,R2	
5497	023340	012703	000004	MOV	#4,R3	
5498	023344	022022		3\$: CMP	(R0)+,(R2)+	
5499	023346	001014		BNE	10\$;BRANCH IF INCORRECT.
5500	023350	077303		SCB	R3,3\$	
5501	023352	026104	000034	CMP	34(R1),R4	;WAS THE FPS CORRECT?
5502	023356	001032		BNE	15\$;BRANCH IF INCORRECT.
5503	023360	005761	000034	TST	34(R1)	;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
5504	023364	100003		BPL	4\$	
5505	023366	026105	000040	CMP	40(R1),R5	;WAS THE FEC CORRECT.
5506	023372	001037		BNE	20\$;BRANCH IF INCORRECT.
5507	023374	000161	000050	4\$: JMP	50(R1)	;RETURN.
5508						
5509						;THE RESULT WAS INCORRECT BUT WAS THIS FAILURE ANTICIPATED?
5510						;SEE IF THE RESULT WAS ANTICIPATED.
5511	023400			10\$:		
5512	023400	011105		MOV	(R1),R5	
5513	023402	006305		ASL	R5	
5514	023404	006305		ASL	R5	
5515	023406	062705	023520	ADD	#NATER1,R5	
5516	023412	010100		MOV	R1,R0	
5517	023414	062700	000022	ADD	#22,R0	
5518	023420	012702	023570	MOV	#NATBF1,R2	
5519	023424	012703	000004	MOV	#4,R3	
5520	023430	022022		11\$: CMP	(R0)+,(R2)+	
5521	023432	001003		BNE	12\$;BRANCH IF NOT ANTICIPATED.
5522	023434	077303		SOB	R3,11\$	
5523						
5524						;THE ERROR WAS ANTICIPATED SO RETURN.
5525	023436	000161	000042	JMP	42(R1)	
5526						
5527						;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
5528	023442	000115		12\$: JMP	(R5)	;GO TO THE PROPER ERROR CALL.
5529						
5530						;THE FPS WAS INCORRECT.
5531	023444	026105	000036	15\$: CMP	36(R1),R5	;WAS THIS ERROR ANTICIPATED?
5532	023450	001002		BNE	16\$;BRANCH IF NOT ANTICIPATED.
5533						
5534						;THE FPS ERROR WAS ANTICIPATED SO RETURN.
5535	023452	000161	000046	JMP	46(R1)	
5536						
5537						;THE FPS FAILURE WAS NOT ANTICIPATED SO REPORT IT HERE.
5538	023456	011102		16\$: MOV	(R1),R2	
5539	023460	006302		ASL	R2	
5540	023462	006302		ASL	R2	

K08

```

5541 023464 062702 023536      ADD      #NATER2,R2
5542 023470 000112              JMP      (R2)                ;GO TO THE PROPER ERROR CALL.
5543
5544      ;REPORT THAT THE FEC WAS INCORRECT.
5545 023472 016137 000040 001256 205:  MOV     40(R1),2#STMP12
5546 023500 010537 001254      MOV     R5,2#STMP11
5547 023504 011102      MOV     (R1),R2
5548 023506 006302      ASL     R2
5549 023510 006302      ASL     R2
5550 023512 062702 023552      ADD     #NATER3,R2
5551 023516 000112              JMP      (R2)                ;GO TO THE PROPER ERROR CALL.
5552

```

```

5553      ;THESE ARE THE ERROR CALLS FOR EACH INDIVIDUAL INSTRUCTION AND CONDITION.
5554 023520 104165  NATER1: ERROR 165                ;NEGD BAD DATA
5555 023522 000403      BR      NATRET
5556 023524 104166      ERROR 166                ;ABSD BAD DATA
5557 023526 000401      BR      NATRET
5558 023530 104167      ERROR 167                ;TSTD BAD DATA
5559 023532 000161 000050  NATRET: JMP     50(R1)
5560

```

```

5561      ;FPS INCORRECT:
5562 023536 104170  NATER2: ERROR 170                ;NEGD FPSX
5563 023540 000774      BR      NATRET
5564 023542 104171      ERROR 171                ;ABSD FPSX
5565 023544 000772      BR      NATRET
5566 023546 104172      ERROR 172                ;TSTD FPSX
5567 023550 000770      BR      NATRET
5568

```

```

5569      ;FEC INCORRECT:
5570 023552 104173  NATER3: ERROR 173                ;NEGD FECX
5571 023554 000766      BR      NATRET
5572 023556 104174      ERROR 174                ;ABSD FECX
5573 023560 000764      BR      NATRET
5574 023562 104175      ERROR 175                ;TSTD FECX
5575 023564 000762      BR      NATRET
5576

```

```

5577 023566 177777      NATBF1: .WORD -1
5578 023570 177777 177777 177777  NATBF1: .WORD -1,-1,-1,-1,-1
5579 023576 177777 177777
5580

```

```

5581 023602      WWBDONE:
5582 023602 104412      RSETUP                ;GO INITIALIZE THE FPS AND STACK; AND
5583                                     ;SEE IF THE USER HAS EXPRESSED
5584                                     ;THE DESIRE TO CHANGE THE SOFTWARE
5585                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5586                                     ;THE USER TYPED CONTROL G?).
5587
5588
5589
5590

```

```

5591      ;*****
5592      ;*TEST 45      SOURCE MODES, MODE 1 (FL=0), TEST
5593      ;*
5594      ;* THIS IS A TEST OF SOURCE MODE 1
5595      ;* USING THE LDFPS INSTR
5596      ;*
5597      ;*****

```

```

5597 023604 000004          TST45: SCOPE
5598
5599
5600 023606          AAC1:
5601 023606 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5602
5603 023610 012700 023666          MOV      #AACTP1,R0          ;SET UP TEST DATA IN BUFFER.
5604 023614 012710 147517          MOV      #147517,(R0)
5605 023620 012737 147517 001240          MOV      #147517,@#STMP3 ;SAVE DATA IN CASE OF ERROR.
5606 023626 012737 023642 001236          MOV      #AAC2,@#STMP2
5607 023634 012737 023726 000004          MOV      #AAC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
5608 023642 170110          AAC2: LDFPS      (R0)          ;TEST INSTRUCTION.
5609
5610 023644 170205          STFPS      R5          ;GET FPS
5611
5612 023646 020027 023666          CMP      R0,#AACTP1          ;IS R0 CORRECT?
5613 023652 001007          BNE      AAC10              ;BR IF NOT.
5614 023654 022705 147517          CMP      #147517,R5          ;IS FPS CORRECT?
5615 023660 001013          BNE      AAC11              ;BR IF NOT.
5616 023662 000437          BR       AACDONE
5617
5618          ;TEST BUFFER AND DATA:
5619 023664 177777          -1
5620 023666 147517          AAC10: 147517
5621 023670 177777          -1
5622
5623          ;REPORT R0 INCORRECT.
5624 023672 012737 023666 001240 AAC10: MOV      #AACTP1,@#STMP3
5625 023700 010037 001242          MOV      R0,@#STMP4
5626 023704 104225          1$:      ERROR      225          ;R0 BAD BUT FSRC FAILED
5627 023706 000425          BR       AACDONE
5628
5629          ;REPORT FPS INCORRECT.
5630 023710 012737 147517 001240 AAC11: MOV      #147517,@#STMP3 ;REPORT FPS INCORRECT.
5631 023716 010537 001242          MOV      R5,@#STMP4
5632 023722 104226          1$:      ERROR      226
5633 023724 000416          BR       AACDONE
5634
5635          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
5636          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
5637          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
5638 023726          AAC20:
5639 023726 011602          MOV      (SP),R2
5640 023730 020227 023644          CMP      R2,#AAC2+2
5641 023734 001405          BEQ      1$
5642 023736 020227 023646          CMP      R2,#AAC2+4
5643 023742 001402          BEQ      1$
5644 023744 000137 042620          JMP      @#CPSPUR
5645 023750 022626          1$:      CMP      (SP)+,(SP)+
5646 023752 010237 001236          MOV      R2,@#STMP2
5647 023756 104227          2$:      ERROR      227          ;ODD ADRES
5648 023760 000400          BR       AACDONE          ;BUT FDSTX IN ST 771
5649
5650 023762          AACDONE:
5651 023762 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
5652          ;SEE IF THE USER HAS EXPRESSED

```

M08

;THE DESIRE TO CHANGE THE SOFTWARE
 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
 ;THE USER TYPED CONTROL G?).

```

5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665 023764 000004
5666
5667 023766
5668 023766 104413
5669
5670 023770 012700 024046
5671 023774 012710 145212
5672 024090 012737 145212 001240
5673 024006 012737 024022 001236
5674 024014 012737 024106 000004
5675
5676 024022 170120
5677
5678 024024 170205
5679
5680 024026 020027 024050
5681 024032 001007
5682 024034 022705 145212
5683 024040 001013
5684 024042 000436
5685
5686
5687
5688 024044 177777
5689 024046 177777
5690 024050 177777
5691
5692
5693
5694 024052 012737 024050 001240
5695 024060 010037 001242
5696 024064 104230
5697 024066 000424
5698
5699
5700 024070 012737 145212 001240
5701 024076 010537 001242
5702 024102 104231
5703 024104 000415
5704
5705
5706
5707
5708 024106

;*****
;TEST 46 SOURCE MODES, MODE 2 (FL=0), TEST
;
; THIS IS A TEST OF SOURCE MODE 2
; USING THE LDFPS INSTR
;*****
↑ST46: SCOPE
BBC1:
        LPERR                ;SET UP THE LOOP ON ERROR ADDRESS.
        MOV #BBC1P1,R0       ;SET UP TEST DATA IN BUFFER.
        MOV #145212,(R0)
        MOV #145212,2#STMP3 ;SAVE DATA IN CASE OF ERROR.
        MOV #BBC2,2#STMP2
        MOV #BBC20,2#ERRVECT ;SET UP FOR TRAPS TO 4.
BBC2:  LDFPS (R0)+           ;TEST INSTRUCTION.
        STFPS R5             ;GET FPS
        CMP R0,#BBC1P1+2    ;IS R0 CORRECT?
        BNE BBC10           ;BR IF NOT.
        CMP #145212,R5     ;IS THE FPS CORRECT?
        BNE BBC11           ;BR IF NOT.
        BR BBCDONE

;TEST BUFFER AND DATA:
        -1
BBC1P1: .WORD -1
        -1

;REPORT R0 INCORRECT.
BBC10:  MOV #BBC1P1+2,2#STMP3
        MOV R0,2#STMP4
1$:     ERROR 230           ;R0 BAD BUT FSRC FAILED
        BR BBCDONE

;REPORT FPS INCORRECT.
BBC11:  MOV #145212,2#STMP3 ;REPORT FPS INCORRECT.
        MOV R5,2#STMP4
1$:     ERROR 231
        BR BBCDONE

;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
BBC20:

```

N08

```

5709 024106 011602          MOV      (SP), R2
5710 024110 020227 024024    CMP      R2, #BBC2+2
5711 024114 001405          BEQ      1$
5712 024116 020227 024026    CMP      R2, #BBC2+4
5713 024122 001402          BEQ      1$
5714 024124 000137 042620    JMP      @#CPSPUR
5715 024130 022626          1$:     CMP      (SP)+, (SP)+
5716 024132 010237 001236    MOV      R2, @#$TMP2
5717 024136 104232          2$:     ERROR   232          ; ODD ADRES
5718                                     ; BUT FDSTX IN ST 771
5719
5720 024140          BBCDONE:
5721 024140 104412          RSETUP          ; GO INITIALIZE THE FPS AND STACK; AND
5722                                     ; SEE IF THE USER HAS EXPRESSED
5723                                     ; THE DESIRE TO CHANGE THE SOFTWARE
5724                                     ; VIRTUAL CONSOLE SWITCH REGISTER (HAS
5725                                     ; THE USER TYPED CONTROL G?).
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735 024142 000004          ;*****
5736                                     ;*TEST 47          SOURCE MODES, MODE 4 (FL=0), TEST
5737                                     ;*
5738                                     ;* THIS IS A TEST OF SOURCE MODE 4
5739                                     ;* USING THE LDFPS INSTR
5740                                     ;*
5741                                     ;*****
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753 024224 177777 177777 177777          -1,-1,-1,-1
5754 024232 177777
5755 024234 177777          DDC1:   -1
5756 024236 177777 177777 177777          -1,-1,-1,-1
5757 024244 177777
5758
5759 024246 012737 024234 001240    DDC10:  MOV      #DDCTP1, @#$TMP3
5760 024254 010037 001242          MOV      R0, @#$TMP4
5761 024260 104233          1$:     ERROR   233          ; R0 BAD BUT FSRC FAILED
5762 024262 000424          BR       DDCDONE
5763 024264 012737 105252 001240    DDC11:  MOV      #105252, @#$TMP3 ; REPORT FPS INCORRECT.
5764 024272 010537 001242          MOV      R5, @#$TMP4

```

024204
024206
042620
001236
104235
104412

18: ERROR 234
BR DCCDONE
DCC20: MOV (SP), R2
CMP R2, #DCC2+2
BEQ 18
CMP R2, #DCC2+4
BEQ 18
JMP J#CPSPUR
18: CMP (SP)+, (SP)+
MOV R2, J#STMP2
28: ERROR 235 ;DDD ADRES
DCCDONE: RSE*UP

:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER HAS
:THE USER TYPED CONTROL G?).

*TEST 50 SOURCE MODES, MODE 3 (FL=0), TEST
*
* THIS IS A TEST OF SOURCE MODE 3
* USING THE LDFPS INSTR

000004
104413
012700 024444
012710 024434
012767 103456 000054
012737 103456 001240
012737 024402 001236
012737 024512 000004
170130
170205
020027 024446
051021
022705 103456
001025
000450

TEST50: SCOPE
EEC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #EECTP2, R0
MOV #EECTP1, (R0)
MOV #103456, EECTP1
MOV #103456, J#STMP3
MOV #EEC2, J#STMP2
MOV #EEC20, J#ERRVECT ;SET UP FOR TRAPS TO 4.
EEC2: LDFPS J(R0)+ ;TEST INSTRUCTION.
STFPS R5 ;GET THE FPS.
CMP R0, #EECTP2+2 ;IS R0 CORRECT?
BNE EEC10 ;BR IF NOT.
CMP #103456, R5 ;IS THE FPS CORRECT?
BNE EEC11 ;BR IF NOT.
BR EECDONE

177777 177777 177777
177777
177777
177777 177777 177777
024434 177777 177777
177777 000000

;TEST BUFFER AND DATA:
-1,-1,-1,-1
EECTP1: -1
-1,-1,-1
EECTP2: EECTP1,-1,-1,-1,

024456 012737 024446 001240
024464 010037 001242
104236
000424

:REPORT R0 INCORRECT.
EEC10: MOV #EECTP2+2, J#STMP3
MOV R0, J#STMP4
18: ERROR 236 ;R0 BAD BUT FSRC FAILED
BR EECDONE

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

024474 012737 103456 001240
024475 012737 001242
024476 104240
024477 000415
024478
024479
024480
024481
024482
024483
024484
024485
024486
024487
024488
024489
024490
024491
024492
024493
024494
024495
024496
024497
024498
024499
024500
024501
024502
024503
024504
024505
024506
024507
024508
024509
024510
024511
024512
024513
024514
024515
024516
024517
024518
024519
024520
024521
024522
024523
024524
024525
024526
024527
024528
024529
024530
024531
024532
024533
024534
024535
024536
024537
024538
024539
024540
024541
024542
024543
024544
024545
024546
024547
024548
024549
024550
024551
024552
024553
024554
024555
024556
024557
024558
024559
024560
024561
024562
024563
024564
024565
024566
024567
024568
024569
024570
024571
024572
024573
024574
024575
024576

```
:REPORT FPS INCORRECT.  
EEC11: MOV #103456, @STMP3 :REPORT FPS INCORRECT.  
MOV R5, @STMP4  
18: ERROR 237  
BR EECDONE  
:TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING  
:EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT  
:FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.  
EEC20: MOV (SP), R2  
CMP R2, @EEC2+2  
BEQ 18  
CMP R2, @EEC2+4  
BEQ 18  
JMP @CPSJUR  
18: CMP (SP)+, (SP)+  
MOV R2, @STMP2  
28: ERROR 240 ;DDD ADRES  
EECDONE: RSE*UP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER .HAS  
;THE USER TYPED CONTROL G?).  
:*****  
:TEST 51 SOURCE MODES, MODE 5 (FL=0), TEST  
:*****  
: THIS IS A TEST OF SOURCE MODE 5  
: USING THE LDFPS INSTR  
:*****  
TEST1: SCOPE  
FFC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #FFCTP2+2, R0 ;SET UP THE TEST DATA BUFFER.  
MOV #FFCTP1, -2(R0)  
MOV #45412, @FFCTP1  
MOV #45412, @STMP3 ;SAVE DATA IN CASE OF ERROR.  
MOV #FFC1, @STMP2  
MOV #FFC20, @ERRVECT ;SET UP FOR TRAPS TO 4.  
FFC2: LDFPS @-(R0) ;TEST INSTRUCTION.  
STFPS R5 ;GET THE FPS.  
CMP R0, #FFCTP2 ;IS R0 CORRECT?  
BNE FFC10 ;BR IF NOT.  
CMP #45412, R5 ;IS THE FPS CORRECT?  
BNE FFC11 ;BR IF NOT.  
BR FFCDONE  
;TEST BUFFER AND DATA:  
FFCTP1: -1  
FFCTP2: FFCTP1, -1, -1, -1
```

5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932

024660 012737 024650 001240
024666 010037 001242
024672 104241
024674 000424

024676 012737 045412 001240
024704 010537 001242
024710 104242
024712 000415

024714 011602
024716 020227 024616
024722 001405
024724 020227 024620
024730 001402
024732 000137 042620
024736 022626
024740 010237 001236
024744 104243
024746 104412

024750 000004
024752
024752 104413
024754 012700 017643
024760 012737 046543 025044
024766 012737 046543 001240
024774 012737 025012 001236
025002 005001
025004 012737 025132 000004
025012 170160 005201
025016 170204
025020 005701
025022 001033
025024 020027 017643
025030 001012
025032 022704 046543
025036 001016
025040 000451

```
:REPORT RD INCORRECT.  
FFC10: MOV #FFC10, R2  
MOV R0, R2  
IS: ERROR 241 ;RD BAD BUT FSRC FAILED  
BR FFCDONE  
  
:REPORT FPS INCORRECT.  
FFC11: MOV #45412, R2  
MOV R5, R2  
IS: ERROR 242 ;REPORT FPS INCORRECT.  
BR FFCDONE  
  
:TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING  
:EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT  
:FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.  
FFC20: MOV (SP), R2  
CMP R2, #+2  
BEQ IS  
CMP R2, #FFC2+4  
BEQ IS  
JMP #CPSRJR  
IS: CMP (SP)+, (SP)+  
MOV R2, #FFC2+4  
IS: ERROR 243 ;ODD ADRES  
FFCDONE: RSETUP  
  
:GO INITIALIZE THE FPS AND STACK; AND  
:SEE IF THE USER HAS EXPRESSED  
:THE DESIRE TO CHANGE THE SOFTWARE  
:VIRTUAL CONSOLE SWITCH REGISTER (HAS  
:THE USER TYPED CONTROL G?).  
:*****  
:TEST 52 SOURCE MODES, MODE 6 (FL=0), TEST  
:*****  
: THIS IS A TEST OF SOURCE MODE 6  
: USING THE LDFPS INSTR  
:*****  
TEST52: SCOPE  
GGC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #GGCTP1-5201, R0 ;SET UP THE TEST DATA BUFFER.  
MOV #46543, R4  
MOV #46543, R2  
MOV #GGC2, R2 ;SAVE DATA IN CASE OF ERROR.  
CLR R1  
MOV #GGC20, R2  
GGC2: LDFPS 5201(R0) ;SET UP FOR TRAPS TO 4.  
STFPS R4 ;TEST INSTRUCTION.  
TST R1 ;GET THE FPS.  
BNE GGC25 ;WAS PC CORRECT AFTER EXECUTION?  
CMP R0, #GGCTP1-5201 ;IS RD CORRECT?  
BNE GGC10 ;BR IF NOT.  
CMP #46543, R4 ;IS THE FPS CORRECT?  
BNE GGC11 ;BR IF NOT.  
BR GGCDONE
```


E09

5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988

025042 177777
025044 177777 177777 177777
025052 177777
025054 177777

025056 012737 017643 001240
025064 010037 001242
025070 104244
025072 000434

025074 012737 046543 001240
025102 010437 001242
025106 104245
025110 000425

025112 012737 025016 001240
025120 012737 025014 001242
025126 104246
025130 000415

025132 011602
025134 020227 025014
025140 001405
025142 020227 025016
025146 001402
025150 000137 042620
025154 022626
025156 010237 001236
025162 104247
025164
025164 104412

:TEST BUFFER AND DATA:
-1
GGCTP1: -1,-1,-1,-1
-1
:REPORT RD INCORRECT.
GGC10: MOV #GGCTP1-5201,2#STMP3
MOV R0,2#STMP4
IS: ERROR 244 ;RD BAD BUT FSRC FAILED
BR GGCDONE
:REPORT FPS INCORRECT.
GGC11: MOV #46543,2#STMP3 ;REPORT FPS INCORRECT.
MOV R4,2#STMP4
IS: ERROR 245
BR GGCDONE
:REPORT PC INCORRECT AFTER INSTRUCTION.
GGC25: MOV #GGC2+4,2#STMP3
MOV #GGC2+2,2#STMP4
IS: ERROR 246 ;PC X
BR GGCDONE
:TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
:EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
:FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
GGC20: MOV (SP) R2
CMP R2,#GGC2+2
BEQ IS
CMP R2,#GGC2+4
BEQ IS
JMP 2#CPSPUR
IS: CMP (SP)+(SP)+
MOV R2,2#STMP2
IS: ERROR 247 ;ODD ADRES
GGCDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
:TEST S3 SOURCE MODES, MODE 7 (FL=0), TEST
:*****
: THIS IS A TEST OF SOURCE MODE 7
: USING THE LDFPS INSTR
:*****
:ST53: SCOPE
HHC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.
MOV #HHCTP1,5201(R0)
MOV #4547,2#HHCTP1
MOV #4547,2#STMP3 ;SAVE DATA IN CASE OF ERROR.

F09

MAINOEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 109
 OFFPCA.P11 31-OCT-76 17:16 T53 SOURCE MODES, MODE 7 (FL=0), TEST

```

5989 025220 012737 025236 001236      MOV      #HHC2,2#STMP2
5990 025226 005001                CLR      R1
5991 025230 012737 025364 000C04      MOV      #HHC20,2#ERRVECT ;SET UP FOR TRAPS TO 4.
5992 025236 170170 005201      HHC2:   LDFPS 25201(R0)          ;TEST INSTRUCTION.
5993 025242 170204                STFPS   R4              ;GET THE FPS.
5994 025244 005701                TST     R1              ;WAS PC CORRECT AFTER EXECUTION?
5995 025246 001036                BNE     HHC25           ;BR IF NOT.
5996 025250 020027 020077      CMP     R0,#HHC2P2-5201 ;IS R0 CORRECT?
5997 025254 001015                BNE     HHC10           ;BR IF NOT.
5998 025256 022704 004547      CMP     #4547,R4       ;IS THE FPS CORRECT?
5999 025262 001021                BNE     HHC11           ;BR IF NOT.
6000 025264 000454                BR      HHCDONE

6001
6002
6003                ;TEST BUFFER AND DATA:
6004 025266 177777                -1
6005 025270 177777 177777 177777      HHC2P1: .WORD -1,-1,-1,-1
6006 025276 177777
6007 025300 177777 177777 177777      HHC2P2: .WORD -1,-1,-1,-1
6008 025306 177777
6009
6010                ;REPORT R0 INCORRECT.
6011 025310 012737 020077 001240      HHC10:  MOV      #HHC2P2-5201,2#STMP3
6012 025316 010037 001242                MOV      R0,2#STMP4
6013 025322 104250                IS:     ERROR   250          ;R0 BAD BUT FSRC FAILED
6014 025324 000434                BR      HHCDONE
6015
6016                ;REPORT FPS INCORRECT.
6017 025326 012737 004547 001240      HHC11:  MOV      #4547,2#STMP3  ;REPORT FPS INCORRECT.
6018 025334 010437 001242                MOV      R4,2#STMP4
6019 025340 104251                IS:     ERROR   251
6020 025342 000425                BR      HHCDONE
6021
6022                ;REPORT PC INCORRECT AFTER INSTRUCTION.
6023 025344 012737 025242 001240      HHC25:  MOV      #HHC2+4,2#STMP3
6024 025352 012737 025240 001242                MOV      #HHC2+2,2#STMP4
6025 025360 104252                IS:     ERROR   252          ;PC X
6026 025362 000415                BR      HHCDONE
6027
6028                ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
6029                ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SC REPORT
6030                ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
6031 025364 011602                HHC20:  MOV      (SP),R2
6032 025366 020227 025240                CMP     R2,#HHC2+2
6033 025372 001405                BEQ     IS
6034 025374 020227 025242                CMP     R2,#HHC2+4
6035 025400 001402                BEQ     IS
6036 025402 000137 042620                JMP     2#CPSPUR
6037 025406 022626                IS:     CMP     (SP)+,(SP)+
6038 025410 010237 001236                MOV     R2,2#STMP2
6039 025414 104253                2S:    ERROR   253          ;DOD ADDRESS
6040 025416 104412                HHCDONE: RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
6041
6042                ;SEE IF THE USER HAS EXPRESSED
6043                ;THE DESIRE TO CHANGE THE SOFTWARE
6044                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                ;THE USER TYPED CONTROL G).

```

6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100

025420 000004
025422
025422 104413
025424 012737 025450 001236
025432 012737 025522 000004
025440 012700 000300
025444 170100
025446 005001
025450 177027
025452 005201
025454 005201
025456 005201
025460 005201
025462 020127 000003
025466 001421
025470 012704 025454
025474 162701 000003
025500 006301
025502 160104
025504 010437 001242
025510 012737 025454 001240
025516 104254
025520 000404
025522 011637 001236
025526 022626
025530 104255
025532
025532 104412

```
::*****  
: *TEST 54 SOURCE MODES, MODE 2 GR7 (FL=1), TEST  
: *  
: * THIS IS A TEST OF THE LDCLD WITH  
: * IMMEDIATE ADDRESSING MODE  
: *  
:*****  
†ST54: SCOPE  
  
IIC1:  
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #IIC2,2#STMP2 ;SAVE DATA IN CASE OF ERROR.  
MOV #IIC20,2#ERRVECT ;SET UP FOR TRAPS TO 4.  
MOV #300,R0  
LDFPS R0  
CLR R1  
  
IIC2: LDCLD (R7)+,ACD ;TEST INSTRUCTION.  
5201  
5201  
5201  
5201  
  
CMP R1,#3 ;WAS PC CORRECT AFTER EXECUTION?  
BEQ IICDONE ;BR IF YES.  
  
:REPORT PC INCORRECT AFTER INSTRUCTION.  
IIC3: MOV #IIC2+4,R4  
SUB #3,R1  
ASL R1  
SUB R1,R4  
MOV R4,2#STMP4  
MOV #IIC2+4,2#STMP3  
IS: ERROR 254 ;BAD CONSTANT  
BR IICDONE  
:TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING  
:EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT  
:FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.  
IIC20: MOV (SP),2#STMP2  
CMP (SP)+,(SP)+  
IS: ERROR 255 ;BAD CONSTANT ODD ADD  
  
IICDONE:  
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
;THE USER TYPED CONTROL G?).  
  
:*****  
: *TEST 55 SOURCE MODES, MODE 2 (FL=1), TEST
```

```

6101
6102
6103
6104
6105
6106 025534 000004
6107
6108 025536
6109 025536 104413
6110 025540 016737 000014 001236
6111 025546 012700 000300
6112 025552 170100
6113 025554 012700 025650
6114 025560 177020
6115
6116 025562 170204
6117 025564 012701 025660
6118 025570 012702 000200
6119 025574 170102
6120 025576 174011
6121 025600 020027 025654
6122 025604 001407
6123
6124 025606 010037 001242
6125 025612 012737 025654 001240
6126 025620 104256
6127 025622 000422
6128
6129 025624 022704 000300
6130 025630 001417
6131
6132
6133 025632 010437 001242
6134 025636 012737 000300 001240
6135 025644 104257
6136 025646 000410
6137
6138
6139
6140 025650 001234 067076 054321
6141 025656 012345
6142 025660 177777 177777 177777
6143 025666 177777
6144
6145 025670
6146 025670 104412
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156

```

```

:
: * THIS IS A TEST OF THE LDCLD INSTR
: * WITH MODE 2.
:
: *****
↑ST55: SCOPE
TCC1:
LPERR                                ;SET UP THE LOOP ON ERROR ADDRESS.
MOV TCC2,2*STMP2                      ;SAVE DATA IN CASE OF ERROR.
MOV #300,R0
LDFPS R0
MOV #TCCBFO,R0                        ;SET UP THE TEST DATA BUFFER.
TCC2: LDCLD (R0)+,ACD                  ;TEST INSTRUCTION.
STFPS R4                              ;GET THE FPS.
MOV #TCCBF1,R1                        ;GET THE RESULT.
MOV #200,R2
LDFPS R2
STD ACD,(R1)
CMP R0,#TCCBFO+4                      ;IS R0 CORRECT?
BEQ TCC3
;REPORT R0 INCORRECT.
MOV R0,2*STMP4
MOV #TCCBFO+4,2*STMP3
IS: ERROR 256                          ;BAD CONST
BR TCCDONE
TCC3: CMP #300,R4                      ;IS THE FPS CORRECT?
BEQ TCCDONE
;REPORT FPS INCORRECT.
MOV R4,2*STMP4
MOV #300,2*STMP3
IS: ERROR 257                          ;FPS X
BR TCCDONE
;TEST BUFFER AND DATA:
TCCBFO: .WORD 01234,67076,54321,012345
TCCBF1: -1,-1,-1,-1
TCCDONE:
RSETUP                                ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
: *****
: *TEST 56 LDCIF AND LDCLF TEST
:
:

```

```

6157
6158
6159
6160
6161 025672 000004
6162
6163
6164
6165
6166 025674
6167 025674 104413
6168 025676 004737 027026
6169
6170 025702 000000 000000
6171 025706 000000 000000
6172 025712 177777 177777
6173 025716 000000
6174 025720 000004
6175 025722 177777
6176 025724 104260
6177 025726 000401
6178 025730 104261
6179 025732
6180
6181
6182 025732
6183 025732 104413
6184 025734 004737 027026
6185
6186 025740 000000 177777
6187 025744 000000 000000
6188 025750 004177 177400
6189 025754 000000
6190 025756 000004
6191 025760 177777
6192 025762 104262
6193 025764 000401
6194 025766 104261
6195 025770
6196
6197
6198 025770
6199 025770 104413
6200 025772 004737 027026
6201
6202 025776 000000 000000
6203 026002 000000 000000
6204 026006 177777 177777
6205 026012 000100
6206 026014 000104
6207 026016 000004
6208 026020 104260
6209 026022 000401
6210 026024 104263
6211 026026
6212

; * THIS IS A TEST OF THE LDCIF AND
; * THE LDCLF INSTRUCTIONS.
; *
; *****
↑T56: SCOPE

;ZERO OPERAND FL=0
KKC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC, @#LDCFSUB ;GO EXECUTE INSTRUCTION.
1$: .WORD 0,0 ;FSRC OPERAND.
2$: .WORD 0,0 ;EXPECTED RESULT.
3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
4$: 0 ;FPS BEFORE EXECUTION.
4 4 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 260 ;REPORT RESULT INCORRECT.
BR 6$
ERROR 261 ;REPORT FPS INCORRECT.
6$:
;ZERO OPERAND FL=0
KKC2:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC, @#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
1$: .WORD 0,-1 ;FSRC OPERAND.
2$: .WORD 0,0 ;EXPECTED RESULT.
3$: 4177,177400 ;ANTICIPATED ERRONEOUS RESULT.
4$: 0 ;FPS BEFORE EXECUTION.
4 4 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 262 ;(BUT FL) ST
BR 6$ ;277 TO 300
ERROR 261 ;INTO 301
6$:
;ZERO OPERAND FL=1
KKC3:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC, @#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
1$: .WORD 0,0 ;FSRC OPERAND.
2$: .WORD 0,0 ;EXPECTED RESULT.
3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
4$: 100 ;FPS BEFORE EXECUTION.
4 104 ;FPS AFTER EXECUTION.
4 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 260 ;REPORT RESULT INCORRECT.
BR 6$
ERROR 263 ;FL WAS CLR'ED
6$:
;OPERAND POSITIVE FL=0

```

```

6213 026026
6214 026026 104413
6215 026030 004737 027026
6216 026034 040000 000000
6217 026040 043600 000000
6218 026044 047600 000000
6219 026050 000017
6220 026052 000000
6221 026054 177777
6222 026056 104264
6223 026060 000401
6224 026062 104261
6225 026064
6226
6227 026064
6228 026064 104413
6229 026066 004737 027026
6230 026072 000001 000000
6231 026076 040200 000000
6232 026102 044200 000000
6233 026106 000017
6234 026110 000000
6235 026112 177777
6236 026114 104264
6237 026116 000401
6238 026120 104261
6239 026122
6240
6241
6242
6243 026122
6244 026122 104413
6245 026124 004737 027026
6246 026130 000252 000000
6247 026134 042052 000000
6248 026140 046052 000000
6249 026144 000000
6250 026146 000000
6251 026150 177777
6252 026152 104264
6253 026154 000401
6254 026156 104261
6255 026160
6256
6257
6258 026160
6259 026160 104413
6260 026162 004737 027026
6261 026166 140000 000000
6262 026172 143600 000000
6263 026176 043600 000000
6264 026202 000007
6265 026204 000010
6266 026206 177777
6267 026210 104265
6268 026212 000401

```

```

KKC4:
LPERR
JSR PC,2#LDCFSJB ;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
1$: .WORD 40000,0 ;FSRC OPERAND.
2$: .WORD 43600,0 ;EXPECTED RESULT.
3$: .WORD 47600,0 ;ANTICIPATED ERRONEOUS RESULT.
4$: 17 ;FPS BEFORE EXECUTION.
0 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 264 ;ST 107 BAD
BR 6$ ;CONSTANT 231 INSD
ERROR 261 ;215
6$:
;OPERAND=1, FL=0
KKC5:
LPERR
JSR PC,2#LDCFSUB ;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
1$: .WORD 1,0 ;FSRC OPERAND.
2$: .WORD 40200,0 ;EXPECTED RESULT.
3$: .WORD 44200,0 ;ANTICIPATED ERRONEOUS RESULT.
4$: 17 ;FPS BEFORE EXECUTION.
0 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 264 ;REPORT RESULT INCORRECT.
BR 6$
ERROR 261 ;REPORT FPS INCORRECT.
6$:
;OPERAND= PATTERN FL=0
KKC6:
LPERR
JSR PC,2#LDCFSJB ;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
1$: .WORD 252,0 ;FSRC OPERAND.
2$: .WORD 42052,0 ;EXPECTED RESULT.
3$: .WORD 46052,0 ;ANTICIPATED ERRONEOUS RESULT.
4$: 0 ;FPS BEFORE EXECUTION.
0 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 264 ;REPORT RESULT INCORRECT.
BR 6$
ERROR 261 ;REPORT FPS INCORRECT.
6$:
;OPERAND=-40000 FL=0
KKC7:
LPERR
JSR PC,2#LDCFSUB ;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
1$: .WORD -40000,0 ;FSRC OPERAND.
2$: .WORD 143600,0 ;EXPECTED RESULT.
3$: .WORD 43600,0 ;ANTICIPATED ERRONEOUS RESULT.
4$: 7 ;FPS BEFORE EXECUTION.
10 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
5$: ERROR 265 ;(SET SIGN) ST 146
BR 6$

```

```

6269 026214 104261          ERROR 261          ;REPORT FPS INCORRECT.
6270 026216          6S:
6271
6272          ;OPERAND=-1      FL=0
6273 026216          KKCB:
6274 026216 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6275 026220 004737 027026          JSR          PC,2#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6276 026224 177777 000000          1S: .WORD -1,0          ;FSRC OPERAND.
6277 026230 140200 000000          2S: .WORD 140200,0          ;EXPECTED RESULT.
6278 026234 144000 000400          3S: .WORD 144000,400          ;ANTICIPATED ERRONEOUS RESULT.
6279 026240 000000          4S: 0          ;FPS BEFORE EXECUTION.
6280 026242 000010          10          ;FPS AFTER EXECUTION.
6281 026244 177777          -1          ;ANTICIPATED ERRONEOUS FPS.
6282 026246 104266          5S: ERROR 266          ;ST 372 TO 152 INTO
6283 026250 000401          BR 6S          ;112 (BUF XNBT)
6284 026252 104261          ERROR 261          ;REPORT FPS INCORRECT.
6285 026254          6S:
6286
6287          ;OPERAND=PATTERN      FL=0
6288 026254          KKCB:
6289 026254 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6290 026256 004737 027026          JSR          PC,2#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6291 026262 125252 000000          1S: .WORD 125252,0          ;FSRC OPERAND.
6292 026266 143652 126000          2S: .WORD 143652,126000          ;EXPECTED RESULT.
6293 026272 043652 126000          3S: .WORD 43652,126000          ;ANTICIPATED ERRONEOUS RESULT.
6294 026276 000007          4S: 7          ;FPS BEFORE EXECUTION.
6295 026300 000010          10          ;FPS AFTER EXECUTION.
6296 026302 177777          -1          ;ANTICIPATED ERRONEOUS FPS.
6297 026304 104265          5S: ERROR 265          ;REPORT RESULT INCORRECT.
6298 026306 000401          BR 6S
6299 026310 104261          ERROR 261          ;REPORT FPS INCORRECT.
6300 026312          6S:
6301
6302          ;OPERAND      POS      FL=1
6303 026312          KKCB:
6304 026312 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6305 026314 004737 027026          JSR          PC,2#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6306 026320 040000 000000          1S: .WORD 40000,0          ;FSRC OPERAND.
6307 026324 047600 000000          2S: .WORD 47600,0          ;EXPECTED RESULT.
6308 026330 043600 000000          3S: .WORD 43600,0          ;ANTICIPATED ERRONEOUS RESULT.
6309 026334 000117          4S: 117          ;FPS BEFORE EXECUTION.
6310 026336 000100          100          ;FPS AFTER EXECUTION.
6311 026340 177777          -1          ;ANTICIPATED ERRONEOUS FPS.
6312 026342 104267          5S: ERROR 267          ;ST 107 CONSTANT
6313 026344 000401          BR 6S          ;BAD 237 INST 217
6314 026346 104261          ERROR 261          ;REPORT FPS INCORRECT.
6315 026350          6S:
6316
6317          ;OPERAND=1      FL=1
6318 026350          KKCB:
6319 026350 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6320 026352 004737 027026          JSR          PC,2#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6321 026356 000000 000001          1S: .WORD 0,1          ;FSRC OPERAND.
6322 026362 040200 000000          2S: .WORD 40200,0          ;EXPECTED RESULT.
6323 026366 034200 000000          3S: .WORD 34200,0          ;ANTICIPATED ERRONEOUS RESULT.
6324 026372 000100          4S: 100          ;FPS BEFORE EXECUTION.

```

```

6325 026374 000100          100          ;FPS AFTER EXECUTION.
6326 026376 177777          -1          ;ANTICIPATED ERRONEOUS FPS.
6327 026400 104267          5$: ERROR 267          ;REPORT RESULT INCORRECT.
6328 026402 000401          BR 6$
6329 026404 104261          ERROR 261          ;REPORT FPS INCORRECT.
6330 026406
6331
6332          ;OPERAND=          PATTERN FL=1
6333 026406          KKC12:
6334 026406 104413          LPERR
6335 026410 004737 027026          JSR PC,2#LDCFSUB          ;SET UP THE LOOP ON ERROR ADDRESS.
6336 026414 000000 000252          1$: .WORD 0,252          ;GO EXECUTE THE INSTRUCTION.
6337 026420 042052 000000          2$: .WORD 42052,0          ;FSRC OPERAND.
6338 026424 036052 000000          3$: .WORD 36052,0          ;EXPECTED RESULT.
6339 026430 000111          4$: 111          ;ANTICIPATED ERRONEOUS RESULT.
6340 026432 000100          100          ;FPS BEFORE EXECUTION.
6341 026434 177777          -1          ;FPS AFTER EXECUTION.
6342 026436 104267          5$: ERROR 267          ;ANTICIPATED ERRONEOUS FPS.
6343 026440 000401          BR 6$          ;REPORT RESULT INCORRECT.
6344 026442 104261          ERROR 261
6345 026444          6$:
6346
6347          ;OPERAND=-40000,0          FL=1
6348 026444          KKC13:
6349 026444 104413          LPERR
6350 026446 004737 027026          JSR PC,2#LDCFSUB          ;SET UP THE LOOP ON ERROR ADDRESS.
6351 026452 140000 000000          1$: .WORD -40000,0          ;GO EXECUTE THE INSTRUCTION.
6352 026456 147600 000000          2$: .WORD 147600,0          ;FSRC OPERAND.
6353 026462 047600 000000          3$: .WORD 47600,0          ;EXPECTED RESULT.
6354 026466 000107          4$: 107          ;ANTICIPATED ERRONEOUS RESULT.
6355 026470 000110          110          ;FPS BEFORE EXECUTION.
6356 026472 177777          -1          ;FPS AFTER EXECUTION.
6357 026474 104265          5$: ERROR 265          ;ANTICIPATED ERRONEOUS FPS.
6358 026476 000401          BR 6$          ;SET SIGN
6359 026500 104261          ERROR 261          ;REPORT FPS INCORRECT.
6360 026502          6$:
6361
6362          ;OPERAND=-1,-1          FL=1
6363 026502          KKC14:
6364 026502 104413          LPERR
6365 026504 004737 027026          JSR PC,2#LDCFSUB          ;SET UP THE LOOP ON ERROR ADDRESS.
6366 026510 177777 177777          1$: .WORD -1,-1          ;GO EXECUTE THE INSTRUCTION.
6367 026514 140200 000000          2$: .WORD 140200,0          ;FSRC OPERAND.
6368 026520 150000 000000          3$: .WORD 150000,0          ;EXPECTED RESULT.
6369 026524 000100          4$: 100          ;ANTICIPATED ERRONEOUS RESULT.
6370 026526 000110          110          ;FPS BEFORE EXECUTION.
6371 026530 177777          -1          ;FPS AFTER EXECUTION.
6372 026532 104266          5$: ERROR 266          ;ANTICIPATED ERRONEOUS FPS.
6373 026534 000401          BR 6$          ;(BUT XNBT)
6374 026536 104261          ERROR 261          ;REPORT FPS INCORRECT.
6375 026540          6$:
6376
6377          ;OPERAND=-PATTERN          FL=1,          ROUND MODE
6378 026540          KKC15:
6379 026540 104413          LPERR
6380 026542 004737 027026          JSR PC,2#LDCFSUB          ;SET UP THE LOOP ON ERROR ADDRESS.

```


6437 026730
 6438 026730 104413
 6439 026732 004737 027026
 6440 026736 100000 000000
 6441 026742 144000 000000
 6442 026746 143600 000000
 6443 026752 000007
 6444 026754 000010
 6445 026756 177777
 6446 026760 104272
 6447 026762 000401
 6448 026764 104261
 6449 026766
 6450
 6451
 6452 026766
 6453 026766 104413
 6454 026770 004737 027026
 6455 026774 100000 000000
 6456 027000 150000 000000
 6457 027004 147600 000000
 6458 027010 000107
 6459 027012 000110
 6460 027014 177777
 6461 027016 104272
 6462 027020 000401
 6463 027022 104261
 6464 027024 000506
 6465
 6466
 6467
 6468
 6469
 6470
 6471
 6472
 6473
 6474
 6475
 6476
 6477
 6478
 6479
 6480
 6481
 6482
 6483
 6484
 6485
 6486
 6487
 6488
 6489
 6490
 6491
 6492

KKC19:
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
 JSR PC, @LDCFSUB ;GO EXECUTE THE INSTRUCTION.
 1S: .WORD 100000,0 ;FSRC OPERAND.
 2S: .WORD 144000,0 ;EXPECTED RESULT.
 3S: .WORD 143600,0 ;ANTICIPATED ERRONEOUS RESULT.
 4S: 7 ;FPS BEFORE EXECUTION.
 10 ;FPS AFTER EXECUTION.
 -1 ;ANTICIPATED ERRONEOUS FPS.
 5S: ERROR 272 ;ST 630 RH*R14+1
 BR 6S
 ERROR 261 ;REPORT FPS INCORRECT.
 6S:

;OPERAND=100000,0 F_=1
 KKC20:
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
 JSR PC, @LDCFSUB ;GO EXECUTE THE INSTRUCTION.
 1S: .WORD 100000,0 ;FSRC OPERAND.
 2S: .WORD 150000,0 ;EXPECTED RESULT.
 3S: .WORD 147600,0 ;ANTICIPATED ERRONEOUS RESULT.
 4S: 107 ;FPS BEFORE EXECUTION.
 110 ;FPS AFTER EXECUTION.
 -1 ;ANTICIPATED ERRONEOUS FPS.
 5S: ERROR 272 ;REPORT RESULT INCORRECT.
 BR 6S
 ERROR 261 ;REPORT FPS INCORRECT.
 6S: BR KKCDONE

; THIS SUBROUTINE, LDCFSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
 ; THE LDCIF OR LDCLF INSTRUCTION AND CHECK THE RESULTS. A CALL
 ; TO IT IS MADE THUS:

```

JSR PC, @LDCFSUB
ACARG: .WORD X,X ;AC OPERAND
RES: .WORD X,X ;EXPECTED RESULT
ERRES: .WORD X,X ;ERRG. RESULT
FPSB: .WORD X ;FPS BEFORE EXECUTION
FPSA: .WORD X ;FPS AFTER EXECUTION
ERFPS: .WORD X ;ERROR FPS
ERR1: ERROR X ;DATA ERROR
BR CONT
ERR2: ERROR X ;FPS ERROR
CONT: ;RETURN ADDRESS
  
```

; THE OPERANDS ARE SET UP (USING ACD AS THE ACCUMULATOR). THEN
 ; THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED.
 ; THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 ; COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL
 ; TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCFSUB WILL
 ; COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCFSUB WILL RETURN
 ; TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF
 ; REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
 ; LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
 ; ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
 ; THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB

6549 027223 001302
6550 027223 001302
6551
6552
6553 027226
6554 027226 104261
6555 027230 000051
6556
6557
6558 027232 000000 000000 000000
6559 027242 000000
6560 027242 104412
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575 027244 000004
6576
6577 027246
6578 027246 104413
6579 027250 004737 030044
6580 027254 000000 000000
6581 027260 000000 000000 000000
6582 027266 000000
6583 027270 177777 177777 177777
6584 027276 177777
6585 027300 000213
6586 027302 000204
6587 027304 177777
6588 027306 104273
6589 027310 003401
6590 027312 104274
6591 027314
6592
6593 027314
6594 027314 104413
6595 027316 004737 030044
6596 027322 000000 177777
6597 027326 000000 000000 000000
6598 027334 000000
6599 027336 004177 177400 000000
6600 027344 000000
6601 027346 000200
6602 027350 000204
6603 027352 177777
6604 027354 104275

BNE 168
JMP 26(R1)
:FPS ERROR NOT ANTICIPATED SO REPORT IT HERE.
168:
178: ERROR 26: ;BAD FPS
BR 38
:DATA BUFFER:
LDCT: .WORD 0,0,0,0
KKCDONE:
PSETUP
:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER .445
:THE USER TYPED CONTROL G?).
:*****
:TEST 57 LDCIF AND LDCIF TEST
:*****
: THIS IS A TEST OF LDCIF AND LDCIF
:*****
:TEST 57: SCOPE
:OPERAND=0 FL=0, FD=1
LLC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,2#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
18: .WORD 0,0 ;FSRC OPERAND.
28: .WORD 0,0,0,0 ;EXPECTED RESULT.
38: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
48: 213 ;FPS BEFORE EXECUTION.
204 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
58: ERROR 273 ;REPORT RESULT INCORRECT.
BR 65
ERROR 274 ;REPORT FPS INCORRECT.
68:
:OPERAND=0 FL=0, FD=1
LLC2:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,2#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
18: .WORD 0,-1 ;FSRC OPERAND.
28: .WORD 0,0,0,0 ;EXPECTED RESULT.
38: .WORD 4177,177400,0,0 ;ANTICIPATED ERRONEOUS RESULT.
48: 200 ;FPS BEFORE EXECUTION.
204 ;FPS AFTER EXECUTION.
-1 ;ANTICIPATED ERRONEOUS FPS.
58: ERROR 275 ;(BUT FL)S+277

```

6605 027356 000401 BR 6S :TO 300 INTO 301
6606 027360 104274 ERROR 274 :REPORT FPS INCORRECT.
6607 027362
6608
6609 ;OPERAND=0 FL=1 FD=1
6610 LLC3:
6611 027362 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
6612 027364 004737 030044 JSR PC,2#LDCDSUB :GO EXECUTE THE INSTRUCTION.
6613 027370 000000 000000 1S: .WORD 0,0 :FSRC OPERAND.
6614 027374 000000 000000 2S: .WORD 0,0,0,0 :EXPECTED RESULT.
6615 027402 000000
6616 027404 177777 177777 3S: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6617 027412 177777
6618 027414 000211 4S: 211 :FPS BEFORE EXECUTION.
6619 027416 000204 204 :FPS AFTER EXECUTION.
6620 027420 177777 -1 :ANTICIPATED ERRONEOUS FPS.
6621 027422 104273 5S: ERROR 273 :REPORT RESULT INCORRECT.
6622 027424 000401 BR 6S
6623 027426 104274 ERROR 274 :REPORT FPS INCORRECT.
6624 027430
6625
6626 ;OPERAND=40000 FL=0 FD=1
6627 LLC4:
6628 027430 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
6629 027432 004737 030044 JSR PC,2#LDCDSUB :GO EXECUTE THE INSTRUCTION.
6630 027436 040000 000000 1S: .WORD 40000,0 :FSRC OPERAND.
6631 027442 043600 000000 2S: .WORD 43600,0,0,0 :EXPECTED RESULT.
6632 027450 000000
6633 027452 047600 000000 3S: .WORD 47600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6634 027460 000000
6635 027462 000217 4S: 217 :FPS BEFORE EXECUTION.
6636 027464 000200 200 :FPS AFTER EXECUTION.
6637 027466 177777 -1 :ANTICIPATED ERRONEOUS FPS.
6638 027470 104276 5S: ERROR 276 :ST 107 BAD CONST
6639 027472 000401 BR 6S
6640 027474 104274 ERROR 274 :REPORT FPS INCORRECT.
6641 027476
6642
6643 ;OPERAND=-40000 FL=0 FD=1
6644 LLC5:
6645 027476 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
6646 027500 004737 030044 JSR PC,2#LDCDSUB :GO EXECUTE THE INSTRUCTION.
6647 027504 140000 000000 1S: .WORD -40000,0 :FSRC OPERAND.
6648 027510 143600 000000 2S: .WORD 143600,0,0,0 :EXPECTED RESULT.
6649 027516 000000
6650 027520 043600 000000 3S: .WORD 43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6651 027526 000000
6652 027530 000200 4S: 200 :FPS BEFORE EXECUTION.
6653 027532 000210 210 :FPS AFTER EXECUTION.
6654 027534 177777 -1 :ANTICIPATED ERRONEOUS FPS.
6655 027536 104277 5S: ERROR 277 : (SET SIGN) ST 176
6656 027540 000401 BR 6S
6657 027542 104274 ERROR 274 :REPORT FPS INCORRECT.
6658 027544
6659
6660 ;OPERAND=40000,0 FL=1 FD=1

```

```

6661 027544          LLC6:
6662 027544 104413          LPERR
6663 027546 004737 030044          JSR      PC,2#LDCDSJB  ;SET UP THE LOOP ON ERROR ADDRESS.
6664 027552 040000 000000          1S:     .WORD 40000,0  ;GO EXECUTE THE INSTRUCTION.
6665 027556 047500 000000 000000          2S:     .WORD 47500,0,0,0 ;FSRC OPERAND.
6666 027564 000000          2S:     .WORD 47500,0,0,0 ;EXPECTED RESULT.
6667 027566 043600 000000 000000          3S:     .WORD 43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6668 027574 000000          3S:     .WORD 43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6669 027576 000317          317
6670 027600 000300          300
6671 027602 177777          -1
6672 027604 104300          5S:     ERROR 300 ;FPS BEFORE EXECUTION.
6673 027606 000401          BR      65 ;FPS AFTER EXECUTION.
6674 027610 104274          ERROR 274 ;ANTICIPATED ERRONEOUS FPS.
6675 027612          6S:     ;ST 107 BAD CONS
6676          ;REPORT FPS INCORRECT.
6677          ;OPERAND=0,1 FL=1 FD=1
6678 027612          LLC7:
6679 027612 104413          LPERR
6680 027614 004737 030044          JSR      PC,2#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
6681 027620 000000 000001          1S:     .WORD 0,1 ;GO EXECUTE THE INSTRUCTION.
6682 027624 045200 000000 000000          2S:     .WORD 40200,0,0,0 ;FSRC OPERAND.
6683 027632 000000          2S:     .WORD 40200,0,0,0 ;EXPECTED RESULT.
6684 027634 034200 000000 000000          3S:     .WORD 34200,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6685 027642 000000          3S:     .WORD 34200,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6686 027644 000300          4S:     300 ;FPS BEFORE EXECUTION.
6687 027646 000300          4S:     300 ;FPS AFTER EXECUTION.
6688 027650 177777          -1
6689 027652 104300          5S:     ERROR 300 ;ANTICIPATED ERRONEOUS FPS.
6690 027654 000401          BR      65 ;REPORT FPS INCORRECT.
6691 027656 104274          ERROR 274 ;REPORT FPS INCORRECT.
6692 027660          6S:     ;REPORT FPS INCORRECT.
6693          ;OPERAND=77777,177777 FL=1 FD=1
6694          LLC8:
6695 027660          LPERR
6696 027660 104413          JSR      PC,2#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
6697 027662 004737 030044          JSR      PC,2#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
6698 027666 077777 177777          1S:     .WORD 77777,177777 ;FSRC OPERAND.
6699 027672 047777 177777 177000          2S:     .WORD 47777,177777,177000,0 ;EXPECTED RESULT.
6700 027700 000000          2S:     .WORD 47777,177777,177000,0 ;EXPECTED RESULT.
6701 027702 177777 177777 177777          3S:     .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6702 027710 177777          3S:     .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6703 027712 000317          4S:     317 ;FPS BEFORE EXECUTION.
6704 027714 000300          4S:     300 ;FPS AFTER EXECUTION.
6705 027716 177777          -1
6706 027720 104273          5S:     ERROR 273 ;ANTICIPATED ERRONEOUS FPS.
6707 027722 000401          BR      65 ;REPORT RESULT INCORRECT.
6708 027724 104274          ERROR 274 ;REPORT FPS INCORRECT.
6709 027726          6S:     ;REPORT FPS INCORRECT.
6710          ;OPERAND=-PATTERN FL=1 FD=1
6711          LLC9:
6712          LPERR
6713 027726 104413          JSR      PC,LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
6714 027726 004767 000110          JSR      PC,LDCDSUB ;GO EXECUTE THE INSTRUCTION.
6715 027730 177777 177526          1S:     .WORD -1,-252 ;FSRC OPERAND.
6716 027734 177777          1S:     .WORD -1,-252 ;FSRC OPERAND.

```

F10

```

6717 027740 142052 000000 000000 25: .WORD 142052.0,0,0 :EXPECTED RESULT.
6718 027746 000000
6719 027750 136052 000000 000000 35: .WORD 136052.0,0,0 :ANTICIPATED ERRONEOUS RESULT.
6720 027756 000000
6721 027760 000307 45: 307 :FPS BEFORE EXECUTION.
6722 027762 000310 :310 :FPS AFTER EXECUTION.
6723 027764 177777 -1 :ANTICIPATED ERRONEOUS FPS.
6724 027766 104300 55: ERROR 300 :REPORT RESULT INCORRECT.
6725 027770 000401 BR 65
6726 027772 104274 ERROR 274 :REPORT FPS INCORRECT.
6727 027774 65:
6728
6729 ;OPERAND=PATTERN FL=1 FD=1 FT=1
6730 027774 LLCID:
6731 027774 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
6732 027776 004767 000042 JSR PC,LDCCSUB :GO EXECUTE THE INSTRUCTION.
6733 030002 012345 067012 15: .WORD 12345,67012 :FSAC OPERAND.
6734 030006 047247 025560 050000 25: .WORD 47247,025560,050000.0 :EXPECTED RESULT.
6735 030014 000000
6736 030016 177777 177777 35: .WORD -1,-1,-1,-1 :ANTICIPATED ERRONEOUS RESULT.
6737 030024 177777
6738 030026 000352 45: 352 :FPS BEFORE EXECUTION.
6739 030030 000340 :340 :FPS AFTER EXECUTION.
6740 030032 177777 -1 :ANTICIPATED ERRONEOUS FPS.
6741 030034 104273 55: ERROR 273 :REPORT RESULT INCORRECT.
6742 030036 000401 BR 65
6743 030040 104274 ERROR 274 :REPORT FPS INCORRECT.
6744 030042 000502 BR LLCIDONE
  
```

;THIS SUBROUTINE, LDCCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
 ;THE LDCID OR LDCLD INSTRUCTION AND CHECK THE RESULTS. A CALL
 ;TO IT IS MADE THUS:

```

JSR PC,LDCCSUB
ACARG: .WORD X,X :AC OPERAND
RES: .WORD X,X,X,X :EXPECTED RESULT
ERRS: .WORD X,X,X,X :ERROR RESULT
FPSB: .WORD X :FPS BEFORE EXECUTION
FPSA: .WORD X :FPS AFTER EXECUTION
ERFPS: .WORD X :ERROR FPS.
ERR1: ERROR X :DATA ERROR.
BR CONT
ERR2: ERROR X :FPS ERROR.
CONT: :RETURN ADDRESS
  
```

;THE OPERANDS ARE SET UP (USING ACC AS THE ACCUMULATOR). THEN
 ;THE LDCID OR LDCLD INSTRUCTION IS EXECUTED.
 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCCSUB RETURNS CONTROL
 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCCSUB
 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCCSUB WILL RETURN
 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCCSUB ITSELF
 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
 ;LDCID OR LDCLD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
 ;ANTICIPATED FAILING DATA PATTERN, ERRS. IF THE FAILURE IN
 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRS THEN LDCCSUB

6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772

G10

6773
 6774
 6775
 6776
 6777
 6778
 6779
 6780
 6781
 6782
 6783
 6784
 6785
 6786
 6787
 6788
 6789
 6790
 6791
 6792
 6793
 6794
 6795
 6796
 6797
 6798
 6799
 6800
 6801
 6802
 6803
 6804
 6805
 6806
 6807
 6808
 6809
 6810
 6811
 6812
 6813
 6814
 6815
 6816
 6817
 6818
 6819
 6820
 6821
 6822
 6823
 6824
 6825
 6826
 6827
 6828

030044 012601
 030046 01E100 000024
 030052 170100
 030054 012737 030064 001236
 030062 010100
 030064 177010
 030066 170204
 030070 012700 027232
 030074 012702 000200
 030100 170102
 030102 174010
 030104 012702 027232
 030110 010237 001242
 030114 010137 001240
 030120 010103
 030122 062703 000004
 030126 010337 001244
 030132 010437 001250
 030136 016137 000026 001252
 030144 010100
 030146 062700 000004
 030152 012703 000002
 030156 022022
 030160 001006
 030162 077303
 030164 026104 000026
 030170 001020
 030172 000161 000040
 030176 012702 027232
 030202 010100
 030204 062700 000014
 030210 012703 000002
 030214 022022
 030216 001003
 030220 077303
 030222 000161 000032
 030226
 030226 104273
 030230 000760
 030232 026104 000030
 030236 001002
 030240 000161 000036

```

:WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
:RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCOSUB WILL
:REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

LDCOSUB:  MOV      (SP)+,R1      ;GET A POINTER TO THE ARGUMENTS.
          MOV      24(R1),R0    ;SET THE FPS.
          LDFPS   R0
          MOV      #15,2#STMP2
          MOV      R1,R0
1$:       LDCID   (R0),ACD      ;TEST INSTRUCTION, LDCID OR LDCLD.

          STFPS   R4           ;GET FPS.
          MOV      #LDCT,R0    ;GET THE RESULT.
          MOV      #200,R2
          LDFPS   R2
          STD     ACD,(R0)

:SEE IF THE RESULT IS CORRECT.
          MOV      #LDCT,R2
          MOV      R2,2#STMP4
          MOV      R1,2#STMP3
          MOV      R1,R3
          ADD     #4,R3
          MOV      R3,2#STMP5
          MOV      R4,2#STMP7
          MOV      26(R1),2#STMP10
          MOV      R1,R0
          ADD     #4,R0
          MOV      #2,R3
2$:       CMP     (R0)+,(R2)+
          BNE     10$
          SOB     R3,2$
          ;BR IF INCORRECT.

          CMP     26(R1),R4      ;IS THE FPS CORRECT?
          BNE     15$           ;BR IF INCORRECT.
3$:       JMP     40(R1)        ;RETURN.

:THE RESULT WAS INCORRECT SO SEE IF THE ERROR WAS ANTICIPATED.
10$:      MOV      #LDCT,R2
          MOV      R1,R0
          ADD     #14,R0
          MOV      #2,R3
11$:      CMP     (R0)+,(R2)+
          BNE     13$
          SOB     R3,11$
          JMP     32(R1)

13$:      ;ERROR NOT ANTICIPATED SO REPORT RESULT INCORRECT HERE.
14$:      ERROR   273          ;BAD RES
          BR      3$

:THE FPS WAS INCORRECT. SEE IF FAILURE WAS ANTICIPATED.
15$:      CMP     30(R1),R4
          BNE     16$
          JMP     36(R1)
          ;FPS ERROR WAS NOT ANTICIPATED SO REPORT FAILURE HERE.
  
```


H10

6829 030244
6830
6831 030244 104274
6832 030246 000751
6833
6834 030250
6835 030250 104412
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851 030252 000004
6852
6853
6854 030254
6855 030254 104413
6856 030256 004767 001334
6857 030262 012345 067012 034567
6858 030270 012345
6859 030272 000010
6860 030274 042145 067012 034567
6861 030302 012345
6862 030304 002145 067012 034567
6863 030312 012345
6864 030314 047217
6865 030316 047200
6866 030320 147200
6867 030322 177777
6868 030324 104304
6869 030326 000400
6870 030330 104305
6871
6872
6873 030332
6874 030332 104413
6875 030334 004737 031616
6876 030340 123456 070123 045670
6877 030346 123456
6878 030350 000177
6879 030352 177656 070123 045670
6880 030360 123456
6881 030362 137656 070123 045670
6882 030370 123456
6883 030372 047207
6884 030374 047210

16S:
17S: ERROR 274 ;BAD FPS
BR 38
LLCDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
:TEST 60 LDEXP TEST
:
:* THIS IS A TEST OF THE LDEXP INST
:* A SUBROUTINE IS USED TO SET UP
:* OPERANDS, EXECUTE THE LDEXP INST AND
:* CHECK THE RESULTS.
:
:*****
†ST60: SCOPE
:
:NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
MMC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,LDXSUB ;GO EXECUTE THE INSTRUCTION.
1S: .WORD 12345,67012,34567,012345 ;ACD OPERAND.
2S: .WORD 10 ;EXPONENT OPERAND.
3S: .WORD 42145,67012,34567,012345 ;EXPECTED RESULT.
4S: .WORD 2145,67012,34567,012345 ;ANTICIPATED ERRONEOUS RESULT.
5S: 47217 ;FPS BEFORE EXECUTION.
47200 ;FPS AFTER EXECUTION.
147200 ;ANTICIPATED ERRONEOUS FPS.
-1 ;EXPECTED FEC.
6S: ERROR 304 ;E12+E12+200 BAD
BR 7S ;ST 624
7S: ERROR 305 ;REPORT FPS INCORRECT.
;ST 625 INTO 304
:NON-ZERO RES NEG.
MMC2:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,2#LDXSUB ;EXPON=377
1S: .WORD 123456,70123,45670,123456 ;ACD OPERAND.
2S: .WORD 177 ;EXPONENT OPERAND.
3S: .WORD 177656,70123,45670,123456 ;EXPECTED RESULT.
4S: .WORD 137656,70123,45670,123456 ;ANTICIPATED ERRONEOUS RESULT.
5S: 47207 ;FPS BEFORE EXECUTION.
47210 ;FPS AFTER EXECUTION.

```

6885 030376 147210 147210 ;ANTICIPATED ERRONEOUS FPS.
6886 030400 177777 -1 ;EXPECTED FEC.
6887 030402 104304 6S: ERROR 304 ;REPORT RESULT INCORRECT.
6888 030404 000401 BR 7S ;
6889 030406 104305 7S: ERROR 305 ;REPORT FPS INCORRECT.
6890 030410
6891
6892 ;NON-ZERO RES, EXP=256=(56)REAL
6893 030410 MMC3:
6894 030410 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6895 030412 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
6896 030416 073261 057645 043323 1S: .WORD 73261,057645,43323,101760 ;ACO OPERAND.
6897 030424 101760
6898 030426 000056 2S: .WORD 56 ;EXPONENT OPERAND.
6899 030430 053461 057645 043323 3S: .WORD 53461,057645,43323,101760 ;EXPECTED RESULT.
6900 030436 101760
6901 030440 177777 177777 4S: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6902 030446 177777
6903 030450 047200 5S: 47200 ;FPS BEFORE EXECUTION.
6904 030452 047200 47200 ;FPS AFTER EXECUTION.
6905 030454 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
6906 030456 177777 -1 ;EXPECTED FEC.
6907 030460 104301 6S: ERROR 301 ;REPORT RESULT INCORRECT.
6908 030462 000401 BR 7S ;
6909 030464 104305 7S: ERROR 305 ;REPORT FPS INCORRECT.
6910 030466
6911
6912 ;EXP=27 (EXCESS 200)=-151 (OCT)
6913 030466 MMC4:
6914 030466 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6915 030470 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
6916 030474 012223 024252 062720 1S: .WORD 12223,24252,62720,21222 ;ACO OPERAND.
6917 030502 021222
6918 030504 177627 2S: .WORD -151 ;EXPONENT OPERAND.
6919 030506 005623 024252 062720 3S: .WORD 5623,24252,62720,21222 ;EXPECTED RESULT.
6920 030514 021222
6921 030516 177777 177777 4S: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6922 030524 177777
6923 030526 047200 5S: 47200 ;FPS BEFORE EXECUTION.
6924 030530 047200 47200 ;FPS AFTER EXECUTION.
6925 030532 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
6926 030534 177777 -1 ;EXPECTED FEC.
6927 030536 104301 6S: ERROR 301 ;REPORT RESULT INCORRECT.
6928 030540 000401 BR 7S ;
6929 030542 104306 7S: ERROR 306 ;(BUT EZBT) ST 544 TO 504 INTO 704 0 (BUT EXBT) ST 704 I
6930 030544
6931
6932 ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
6933 ; FIV=1
6934 030544 MMC5:
6935 030544 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6936 030546 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
6937 030552 030131 032334 035363 1S: .WORD 30131,32334,35363,73031 ;ACO OPERAND.
6938 030560 073031
6939 030562 177600 2S: .WORD -200 ;EXPONENT OPERAND.
6940 030564 000131 032334 035363 3S: .WORD 00131,32334,35363,73031 ;EXPECTED RESULT.

```

J10

MAINDEC-11-FPP34-A
DFFPCA.P11

31-OCT-76

PDP 11 34 FPP DIAGNOSTIC
T60

LDEXF TEST

MACY11 27(1006) 31-OCT-76 17:35 PAGE 126

```

6941 030572 073031
6942 030574 000000 000000 000000 4S: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6943 030602 000000
6944 030604 042200 5S: 42200 ;FPS BEFORE EXECUTION.
6945 030606 142204 142204 ;FPS AFTER EXECUTION.
6946 030610 042202 42202 ;ANTICIPATED ERRONEOUS FPS.
6947 030612 000012 12 ;EXPECTED FEC.
6948 030614 104307 6S: ERROR 307 ;(BUT EXBT) ST 704 TO 64 INST 264
6949 030616 000401 BR 7S
6950 030620 104310 ERROR 310 ;(BUT FIU) ST 264 X
6951 030622
6952
6953 ;EXP=0 (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
6954 030622 MMC6:
6955 030622 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6956 030624 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
6957 030630 140414 024344 045464 1S: .WORD 140414,24344,45464,74045 ;ACD OPERAND.
6958 030636 074045
6959 030640 177600 2S: .WORD -200 ;EXPONENT OPERAND.
6960 030642 100014 024344 045464 3S: .WORD 100014,24344,45464,74045 ;-0 ;EXPECTED RESULT.
6961 030650 074045
6962 030652 000000 000000 000000 4S: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
6963 030660 000000
6964 030662 042200 5S: 42200 ;FPS BEFORE EXECUTION.
6965 030664 142214 142214 ;FPS AFTER EXECUTION.
6966 030666 042214 42214 ;ANTICIPATED ERRONEOUS FPS.
6967 030670 000012 12 ;EXPECTED FEC.
6968 030672 000307 6S: ERROR 307 ;REPORT RESULT INCORRECT.
6969 030674 000401 BR 7S
6970 030676 104310 ERROR 310 ;REPORT FPS INCORRECT.
6971 030700
6972
6973 ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
6974
6975 030700 MMC7:
6976 030700 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6977 030702 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
6978 030706 051525 035455 005675 1S: .WORD 51525,35455,5675,05152 ;ACD OPERAND.
6979 030714 005152
6980 030716 177600 2S: .WORD -200 ;EXPONENT OPERAND.
6981 030720 000000 000000 000000 3S: .WORD 0,0,0,0 ;EXPECTED RESULT.
6982 030726 000000
6983 030730 000125 035455 005675 4S: .WORD 00125,35455,5675,05152 ;ANTICIPATED ERRONEOUS RESULT.
6984 030736 005152
6985 030740 045200 45200 ;FPS BEFORE EXECUTION.
6986 030742 045204 45204 ;FPS AFTER EXECUTION.
6987 030744 145204 145204 ;ANTICIPATED ERRONEOUS FPS.
6988 030746 177777 -1 ;EXPECTED FEC.
6989 030750 104311 6S: ERROR 311 ;(BUT FIU) ST 264 X ;REPORT RESULT INCORRECT
6990 030752 000401 BR 7S
6991 030754 104302 ERROR 302 ;REPORT FPS INCORRECT.
6992 030756
6993
6994 ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
6995 030756 MMC8:
6996 030756 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.

```


L10

```

7053 ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
7054 031170 MMC11:
7055 031170 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7056 031172 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7057 031176 012131 014151 016171 1S: .WORD 12131,14151,16171,10111 ;ACD OPERAND.
7058 031204 010111
7059 031206 001006 2S: .WORD 1006 ;EXPONENT OPERAND.
7060 031210 041531 014151 016171 3S: .WORD 41531,14151,16171,10111 ;EXPECTED RESULT.
7061 031216 010111
7062 031220 000000 000000 000000 4S: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7063 031226 000000
7064 031230 041200 5S: 41200 ;FPS BEFORE EXECUTION.
7065 031232 141202 141202 ;FPS AFTER EXECUTION.
7066 031234 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
7067 031236 000010 10 ;EXPECTED FEC.
7068 031240 104314 6S: ERROR 314 ;(BUT FIV) ST 104
7069 031242 000401 BR 7S
7070 031244 104302 ERROR 302 ;REPORT FPS INCORRECT.
7071 031246
7072
7073 ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
7074 031246 MMC12:
7075 031246 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7076 031250 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7077 031254 027262 025242 023222 1S: .WORD 27262,25242,23222,21202 ;ACD OPERAND.
7078 031262 021202
7079 031264 016115 2S: .WORD 16115 ;EXPONENT OPERAND.
7080 031266 000000 000000 000000 3S: .WORD 0,0,0,0 ;EXPECTED RESULT.
7081 031274 000000
7082 031276 063262 025242 023222 4S: .WORD 63262,25242,23222,21202 ;ANTICIPATED ERRONEOUS RESULT.
7083 031304 021202
7084 031306 046200 5S: 46200 ;FPS BEFORE EXECUTION.
7085 031310 046206 46206 ;FPS AFTER EXECUTION.
7086 031312 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
7087 031314 177777 -1 ;EXPECTED FEC.
7088 031316 104315 6S: ERROR 315 ;(BUT FIV) ST 104
7089 031320 000401 BR 7S
7090 031322 104302 ERROR 302 ;REPORT FPS INCORRECT.
7091 031324
7092
7093 ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1
7094
7095 031324 MMC13:
7096 031324 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7097 031326 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7098 031332 030313 032333 034353 1S: .WORD 30313,32333,34353,36373 ;ACD OPERAND.
7099 031340 036373
7100 031342 010611 2S: .WORD 10611 ;EXPONENT OPERAND.
7101 031344 002313 032333 034353 3S: .WORD 2313,32333,34353,36373 ;EXPECTED RESULT.
7102 031352 036373
7103 031354 000000 000000 000000 4S: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7104 031362 000000
7105 031364 041200 5S: 41200 ;FPS BEFORE EXECUTION.
7106 031366 141202 141202 ;FPS AFTER EXECUTION.
7107 031370 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
7108 031372 000010 10 ;EXPECTED FEC.

```

M10

```

7109 031374 104316 6S: ERROR 316 ;(BUT FIV) ST 144
7110 031376 000401 BR 7S
7111 031400 104302 ERROR 302 ;REPORT FPS INCORRECT.
7112 031402 7S:
7113
7114 ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
7115
7116 031402 MMC14:
7117 031402 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7118 031404 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7119 031410 040414 042434 044454 1S: .WORD 40414,42434,44454,46474 ;ACO OPERAND.
7120 031416 046474
7121 031420 016723 2S: .WORD 16723 ;EXPONENT OPERAND.
7122 031422 000000 000000 000000 3S: .WORD 0,0,0,0 ;EXPECTED RESULT.
7123 031430 000000
7124 031432 024614 042434 044454 4S: .WORD 24614,42434,44454,46474 ;ANTICIPATED ERRONEOUS RESULT.
7125 031440 046474
7126 031442 046200 5S: 46200 ;FPS BEFORE EXECUTION.
7127 031444 046206 46206 ;FPS AFTER EXECUTION.
7128 031446 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
7129 031450 177777 -1 ;EXPECTED FEC.
7130 031452 104317 6S: ERROR 317 ;(BUT FIV) ST 144
7131 031454 000401 BR 7S
7132 031456 104302 ERROR 302 ;REPORT FPS INCORRECT.
7133 031460 7S:
7134
7135 ;EXP= 254 (OCT)= 454 (EXCESS 200) FIV=1
7136
7137 031460 MMC15:
7138 031460 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7139 031462 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7140 031466 050515 052535 054555 1S: .WORD 50515,52535,54555,56575 ;ACO OPERAND.
7141 031474 056575
7142 031476 000254 2S: .WORD 254 ;EXPONENT OPERAND.
7143 031500 013115 052535 054555 3S: .WORD 13115,52535,54555,56575 ;EXPECTED RESULT.
7144 031506 056575
7145 031510 000000 000000 000000 4S: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7146 031516 000000
7147 031520 041200 5S: 41200 ;FPS BEFORE EXECUTION.
7148 031522 141202 141202 ;FPS AFTER EXECUTION.
7149 031524 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
7150 031526 000010 10 ;EXPECTED FEC.
7151 031530 104320 6S: ERROR 320 ;(BUT FIV) ST344
7152 031532 000401 BR 7S
7153 031534 104302 ERROR 302 ;REPORT FPS INCORRECT.
7154 031536 7S:
7155
7156 ;EXP= 313 (OCT)= 513(EXCESS 200) FIV=0
7157
7158 031536 MMC16:
7159 031536 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7160 031540 004737 031616 JSR PC,2#LDXSUB ;GO EXECUTE THE INSTRUCTION.
7161 031544 060616 062636 064656 1S: .WORD 60616,62636,64656,66676 ;ACO OPERAND.
7162 031552 066676
7163 031554 000313 2S: .WORD 313 ;EXPONENT OPERAND.
7164 031556 000000 000000 000000 3S: .WORD 0,0,0,0 ;EXPECTED RESULT.

```

N10

MAINDEC-11-FPP34-A POP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 130
 DFFPCA.P11 31-OCT-76 17:16 T60 LDEXF TEST

```

7165 031564 000000
7166 031566 022616 062636 064656 45: .WORD 22616,62636,64656,66676 ;ANTICIPATED ERRONEOUS RESULT.
7167 031574 066676
7168 031576 046200 55: 46200 ;FPS BEFORE EXECUTION.
7169 031600 046206 46206 ;FPS AFTER EXECUTION.
7170 031602 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
7171 031604 177777 -1 ;EXPECTED FEC.
7172 031606 104321 65: ERROR 321 ;(BUT FIV) ST 344
7173 031610 000401 BR 75
7174 031612 10430L ERROR 302 ;REPORT FPS INCORRECT.
7175 031614
7176 031614 000540 75: BR MMCDONE
  
```

;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
 ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
 ;TO IT IS MADE THUS:

```

JSR PC, @LDXSUB
ACARG: .WORD X,X,X,X ;AC OPERAND
EXP: .WORD X ;EXPONENT
RES: .WORD X,X,X,X ;EXPECTED RESULT
ERRES: .WORD X,X,X,X ;ERROR RESULT
FPSB: .WORD X ;FPS BEFORE EXECUTION
FPSA: .WORD X ;FPS AFTER EXECUTION
ERFPS: .WORD X ;ERROR FPS.
FEC: .WORD X ;EXPECTED FEC
ERR1: ERROR X ;DATA ERROR.
      BR CONT
ERR2: ERROR X ;FPS ERROR.
CONT: ;RETURN ADDRESS
  
```

;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
 ;THE LDEXP INSTRUCTION IS EXECUTED.
 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDXSUB RETURNS CONTROL
 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDXSUB
 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDXSUB WILL RETURN
 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDXSUB ITSELF
 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
 ;LDEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDXSUB
 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDXSUB WILL
 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

```

7211 031616 012601 LDXSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
7212 031620 012700 000200 MOV #200,R0 ;LOAD THE ACO OPEXAND.
7213 031624 170100 LDFPS R0
7214 031626 010100 MOV R1,R0
7215 031630 172410 LDD (R0),ACO
7216 031632 012737 031654 001236 MOV #15,@STMP2
7217 031640 016100 000032 MOV 32(R1),R0 ;SET UP THE FPS.
7218 031644 170100 LDFPS R0
7219 031646 010100 MOV R1,R0
7220 031650 062700 000010 ADD #10,R0
  
```

```

031654 176410 18: LDEXP (R0),R0 ;TEST INSTRUCTION.
031656 170204 STFPS R4 ;GET THE FPS.
031660 170395 STST R5 ;GET THE FEC.
031662 012700 000200 MOV #200,R0 ;GET THE RESULT.
031666 170100 LDFPS R0
031670 012700 032106 MOV #LDXT,R0
031674 174019 STD ACC,(R0)
031676 010437 001250 MOV R4,#STMP7
031678 016137 000034 001252 MOV 34(R1),#STMP10
031710 010537 001254 MOV R5,#STMP11
031714 016137 000040 001256 MOV 40(R1),#STMP12
031722 010102 MOV R1,R2
031724 010237 001240 MOV R2,#STMP3
031730 062702 000010 ADD #10,R2
031734 011237 001242 MOV (R2),#STMP4
031740 062702 000002 ADD #2,R2
031744 010237 001244 MOV R2,#STMP5
031750 012737 032106 001246 MOV #LDXT,#STMP6
031756 012702 032106 MOV #LDXT,R2 ;SEE IF THE RESULT WAS CORRECT.
031762 010103 MOV R1,R3
031764 062703 000012 ADD #12,R3
031770 012700 000004 MOV #4,R0
031774 022223 28: CMP (R2)+,(R3)+
031776 001014 BNE 10$ ;BRANCH IF NOT CORRECT.
032000 077003 SCB R0,28
032002 020461 000034 CMP R4,34(R1) ;SEE IF THE FPS WAS CORRECT.
032006 001026 BNE 15$ ;BRANCH IF NOT CORRECT.
032010 005761 000034 ST 34(R1)
032014 100003 BPL 35$
032016 020561 000040 CMP R5,40(R1) ;SEE IF THE FEC WAS CORRECT.
032022 001027 BNE 20$ ;BRANCH IF NOT CORRECT.
032024 000161 000050 35: JMP 50(R1) ;RETURN.
;THE RESULT WAS INCORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
032030 012702 032106 10$: MOV #LDXT,R2
032034 010103 MOV R1,R3
032036 062703 000022 ADD #22,R3
032042 012700 000004 MOV #4,R0
032046 022223 11$: CMP (R2)+,(R3)+
032050 001003 BNE 12$
032052 077003 SCB R0,11$
032054 000161 000042 JMP 42(R1)
;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
032060 12$:
032060 104301 13$: ERROR 301 ;BAD RES
032062 000760 BR 38
;SEE IF THE FPS ERROR WAS ANTICIPATED.
032064 026104 000036 15$: CMP 36(R1),R4
032070 001002 BNE 16$
032072 000161 000046 JMP 46(R1)
032076 16$:

```


7278 032076 104302
7279 032100 000751
7280
7281 032102
7282
7283 032102 104303
7284 032104 000747
7285
7286 032106 030000 000000 000000
7287 032114 000000
7288
7289
7290
7291 032116
7292 032116 104412
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307 032120 000004
7308
7309
7310 032122
7311 032122 104413
7312 032124 012700 032222
7313 032130 012701 000006
7314 032134 012720 177777
7315 032140 077103
7316 032142 012700 102345
7317 032146 012737 032170 001236
7318 032154 012737 032322 000004
7319 032162 170100
7320 032164 012700 032226
7321
7322 032170 170210
7323 032172 020027 032226
7324 032176 001017
7325 032200 023727 032226 102345
7326 032206 001023
7327 032210 023727 032230 177777
7328 032216 001030
7329 032220 000453
7330
7331
7332 032222 177777 177777

:THE FPS WAS NOT ANTICIPATED SO REPORT IT HERE.
178: ERROR 302 :BAD FPS
BR 38 :BU? EZB?YB
:S? 063

208:
:REPORT FEC INCORRECT.
218: ERROR 303 :BAD FEC
BR 38

:DATA BUFFER:
LEXT: .WORD 0.0.0.0

MDCONE:
RSETUP
:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER (HAS
:THE USER TYPED CONTROL G?).

:TEST 61 DESTINATION MODES, MODE 1 (FL=0), TEST
* THIS IS A TEST OF DESTINATION MODE 1 USING
* THE STFPS INSTRUCTION

†ST61: SCOPE

NMC1:
LPERR :SET UP THE LOOP ON ERROR ADDRESS.
MOV #NMC1B0,R0 :SET UP THE DATA BUFFER.
MOV #6,R1
18: MOV #-1,(R0)+
SOB R1,18
MOV #102345,R0
MOV #NMC2,@STMP2
MOV #NMC25,@ERRVECT :SET JP FOR TRAPS TO 4.
LDFPS R0 :SET JP FPS.
MOV #NMC1B1,R0
NMC2: STFPS (R0) :TEST INSTRUCTION.
CMP R0,#NMC1B1 :IS R0 CORRECT?
BNE NMC10 :BRANCH IF NOT CORRECT.
CMP @NMC1B1,#102345 :IS RESULT CORRECT?
BNE NMC15 :BRANCH IF NOT CORRECT.
CMP @NMC1B1+2,#-1 :IS THE RESULT CORRECT?
BNE NMC20 :BRANCH IF NOT CORRECT.
BR NMCDONE

:TEST DATA BUFFER:
NMC1B0: .WORD -1,-1

D11

```

7333 032226 177777 177777 177777 NNCTB1: .WORD -1,-1,-1,-1
7334 032234 177777
7335
7336
7337 032236 010037 001242 :REPORT RD INCORRECT.
7338 032242 012737 032226 001240 NNC10: MOV RD,2@STMP4
7339 032250 : MOV @NNCTB1,2@STMP3
7340 032250 104377 IS:
7341 032252 000001 ERROR 377
7342 : .WORD 1
7343 032254 000435 BR NNCDONE ;RD BAD (BUT
7344 : FOST,X
7345
7346 032256 012737 102345 001240 :REPORT RESULT INCORRECT.
7347 032264 013737 032226 001242 NNC15: MOV @102345,2@STMP3 : ST 634
7348 032272 : MOV @NNCTB1,2@STMP4
7349 032272 104377 IS:
7350 032274 000002 ERROR 377
7351 : .WORD 2 ;BAD DATA
7352 032276 000424 BR NNCDONE
7353
7354
7355
7356 032300 012737 177777 001240 :REPORT RESULT INCORRECT.
7357 032306 013737 032230 001242 NNC20: MOV @-1,2@STMP3
7358 032314 : MOV @NNCTB1+2,2@STMP4
7359 032314 104377 IS:
7360 032316 000003 ERROR 377
7361 : .WORD 3 ;(BUT GR7,FL)
7362 032320 000413 BR NNCDONE ;ST 357 TO 416
7363 : INTO 417
7364
7365 :IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7366 :DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7367 :TO THE SPURIOUS TRAP TO 4 HANDLER.
7368 032322 011604 NNC25: MOV (SP),R4
7369 032324 020427 032172 CMP R4,@NNC2+2
7370 032330 001402 BEQ IS
7371 032332 000137 042620 JMP @CPSPUR
7372
7373 032336 011637 001236 IS: MOV (SP),2@STMP2
7374 032342 022626 CMP (SP)+,(SP)+
7375 032344
7376 032344 104377 2S:
7377 032346 000004 ERROR 377
7378 : .WORD 4 ;(BUT FOST)+ ST634
7379
7380 032350 NNCDONE:
7381 032350 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
7382 :SEE IF THE USER HAS EXPRESSED
7383 :THE DESIRE TO CHANGE THE SOFTWARE
7384 :VIRTUAL CONSOLE SWITCH REGISTER (HAS
7385 :THE USER TYPED CONTROL G?).
7386
7387
7388

```

E11

7389
7390
7391
7392
7393
7394
7395 032352 000004
7396
7397
7398 032354
7399 032354 104413
7400 032356 012700 032454
7401 032362 012701 000006
7402 032366 012720 177777
7403 032372 077103
7404 032374 012700 105412
7405 032400 012737 032422 001236
7406 032406 012737 032554 000004
7407 032414 170100
7408 032416 012700 032460
7409
7410 032422 170220
7411 032424 020027 032462
7412 032430 001017
7413 032432 023727 032460 105412
7414 032440 001023
7415 032442 023727 032462 177777
7416 032450 001030
7417 032452 000453
7418
7419
7420 032454 177777 177777
7421 032460 177777 177777 177777
7422 032466 177777
7423
7424
7425 032470 010037 001242
7426 032474 012737 032462 001240
7427 032502
7428 032502 104377
7429 032504 000005
7430
7431 032506 000435
7432
7433
7434 032510 012737 105412 001240
7435 032516 013737 032460 001242
7436 032524
7437 032524 104377
7438 032526 000006
7439
7440 032530 000424
7441
7442
7443
7444 032532 012737 177777 001240

: *TEST 62 DESTINATION MODES, MODE 2 (FL=0), TEST
: *
: * THIS IS A TEST OF DESTINATION MODE 2 USING
: * THE STFPS INSTRUCTION
: *
: *****
: *ST62: SCOPE

0001: LPERA ; SET UP THE LOOP ON ERROR ADDRESS.
MOV #00CTB0, R0 ; SET UP THE DATA BUFFER.
MOV #6, R1
15: MOV #-1, (R0)+
SOB R1, 15
MOV #105412, R0
MOV #00C2, 2#STMP2
MOV #00C25, 2#ERRVECT ; SET UP FOR TRAPS TO VECTOR 4.
LDFPS R0 ; SET UP FPS.
MOV #00CTB1, R0

0002: STFPS (R0)+ ; TEST INSTRUCTION.
CMP R0, #00CTB1+2 ; IS R0 CORRECT?
BNE 00C10 ; BRANCH IF NOT CORRECT.
CMP 2#00CTB1, #105412 ; IS THE RESULT CORRECT?
BNE 00C15 ; BRANCH IF NOT CORRECT.
CMP 2#00CTB1+2, #-1 ; IS THE RESULT CORRECT?
BNE 00C20 ; BRANCH IF NOT CORRECT.
BR 00C0NE

: TEST DATA BUFFER:
00CTB0: .WORD -1, -1
00CTB1: .WORD -1, -1, -1, -1

: REPORT R0 INCORRECT.
00C10: MOV R0, 2#STMP4
MOV #00CTB1+2, 2#STMP3
15: ERROR 377
.WORD 5
BR 00CDONE ; R0 BAD (BUT
; FDST)X

: REPORT RESULT INCORRECT.
00C15: MOV #105412, 2#STMP3 ; ST 634
MOV 2#00CTB1, 2#STMP4
15: ERROR 377
.WORD 6
BR 00CDONE ; BAD DATA

: REPORT RESULT INCORRECT.
00C20: MOV #-1, 2#STMP3

F11

7445 032540 013737 032462 001242
7446 032546
7447 032546 104377
7448 032550 000007
7449
7450 032552 000413
7451
7452
7453
7454
7455
7456 032554 011604
7457 032556 020427 032424
7458 032562 001402
7459 032564 000137 042620
7460
7461 032570 011637 001236
7462 032574 022626
7463 032576
7464 032576 104377
7465 032600 000010
7466
7467
7468 032602
7469 032602 104412
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484 032604 000004
7485
7486 032606
7487 032606 104413
7488 032610 012700 032706
7489 032614 012701 000006
7490 032620 012720 177777
7491 032624 077103
7492 032626 012700 105555
7493 032632 012737 032654 001236
7494 032640 012737 033006 000004
7495 032646 170100
7496 032650 012700 032714
7497
7498 032654 170240
7499 032656 020027 032712
7500 032662 001017

MOV 2#00CTB1+2,2#STMP4
1S: ERROR 377
.WORD 7
BR COCDONE ;(BUT GR7,FL)
;ST 357 TO 416
;INTC 417
:IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
:DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED. IF NOT GO
:TO THE SPURIOUS TRAP TO 4 HANDLER.
OOC25: MOV (SP),R4
CMP R4,#00C2+2
BEQ 1S
JMP 2#CPSPUR
1S: MOV (SP),2#STMP2
CMP (SP)+,(SP)+
2S: ERROR 377
.WORD 10 ;(BUT FDST)+ ST634
OOCDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
:TEST 63 DESTINATION MODES, MODE 4 (FL=0), TEST
:*****
: THIS IS A TEST OF DESTINATION MODE 4 USING
: THE STFPS INSTRUCTION
:*****
↑ST63: SCOPE
PPC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #6,R1 ;SET UP THE DATA BUFFER.
1S: MOV #-1,(R0)+
SOB R1,1S
MOV #105555,R0
MOV #PPC2,2#STMP2
MOV #PPC25,2#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS R0 ;SET UP FPS.
MOV #PPCTB1+2,R0
PPC2: STFPS -(R0) ;TEST INSTRUCTION.
CMP R0,#PPCTB1 ;IS R0 CORRECT?
BNE PPC10 ;BRANCH IF NOT CORRECT.

G11

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 136
 OFFPCA.P11 31-OCT-76 17:16 T63 DESTINATION MODES, MODE 4 (FL=0), TEST

```

7501 032664 023727 032712 105555      CMP      2*PPCTB1,0105555      ;IS THE RESULT CORRECT?
7502 032672 001023      BNE      PPC15      ;BRANCH IF NOT CORRECT.
7503 032674 023727 032714 177777      CMP      2*PPCTB1+2,0-1      ;IS THE RESULT CORRECT?
7504 032702 001030      BNE      PPC20      ;BRANCH IF NOT CORRECT.
7505 032704 000453      BR      PPCDONE
7506
7507      ;TEST DATA BUFFER:
7508 032706 177777 177777      PPC7BC: .WORD      -1,-1
7509 032712 177777 177777 177777      PPC7B1: .WORD      -1,-1,-1,-1
7510 032720 177777
7511
7512      ;REPORT RO INCORRECT.
7513 032722 010037 001242      PPC1C: MOV      RO,2*STMP4
7514 032726 012737 032712 001240      MOV      #PPCTB1,2*STMP3
7515 032734      IS:
7516 032734 104377      ERROR      377
7517 032736 000011      .WORD      11
7518      ;RO BAD (BUT
7519 032740 000435      BR      PPCDONE      ;FDST)X
7520
7521      ;REPORT RESULT INCORRECT.
7522 032742 012737 105555 001240      PPC15: MOV      #105555,2*STMP3      ; ST 634
7523 032750 013737 032712 001242      MOV      2*PPCTB1,2*STMP4
7524 032756      IS:
7525 032756 104377      ERROR      377
7526 032760 000012      .WORD      12
7527      ;BAD DATA
7528 032762 000424      BR      PPCDONE
7529
7530
7531      ;REPORT RESULT INCORRECT.
7532 032764 012737 177777 001240      PPC20: MOV      #-1,2*STMP3
7533 032772 013737 032714 001242      MOV      2*PPCTB1+2,2*STMP4
7534 033000      IS:
7535 033000 104377      ERROR      377
7536 033002 000013      .WORD      13
7537      ;(BUT GR7,FL)
7538 033004 000413      BR      PPCDONE      ;ST 357 TO 416
7539      ;INTO 417
7540
7541      ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7542      ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7543      ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7544 033006 011604      PPC25: MOV      (SP),R4
7545 033010 020427 032656      CMP      R4,#PPC2+2
7546 033014 001402      BEQ      15
7547 033016 000137 042620      JMP      2*CPSPUR
7548
7549 033022 011637 001236      IS:      MOV      (SP),2*STMP2
7550 033026 022626      CMP      (SP)+,(SP)+
7551 033030      2S:
7552 033030 104377      ERROR      377
7553 033032 000014      .WORD      14
7554      ;(BUT FDST)+ ST634
7555
7556 033034      PPCDONE:
  
```

H11

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 137
DFFPCA.P11 31-OCT-76 17:16 T63 DESTINATION MODES, MODE 4 (FL=0), TEST

```
7557 033034 104412 RSETJP ;GO INITIALIZE THE FPS AND STACK; AND
7558 ;SEE IF THE USER HAS EXPRESSED
7559 ;THE DESIRE TO CHANGE THE SOFTWARE
7560 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7561 ;THE USER TYPED CONTROL G?).
7562
7563
7564
7565 *****
7566 :*TEST 54 DESTINATION MODES, MODE 3 (FL=0), TEST
7567 :*
7568 :* THIS IS A TEST OF DESTINATION MODE 3 USING
7569 :* THE STFPS INSTRUCTION
7570 :*
7571 *****
7572 033036 000004 ST64: SCOPE
7573
7574 033040 QOC1:
7575 033040 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
7576 033042 012700 033144 MOV #QOCTB0,RO ;SET UP THE DATA BUFFER.
7577 033046 012701 000010 MOV #10,R1
7578 033052 012720 177777 IS: MOV #-1,(RO)+
7579 033056 077103 SOB R1,IS
7580 033060 012700 106653 MOV #106653,RO
7581 033064 012737 033112 001236 MOV #QOC2,2*STMP2
7582 033072 012737 033250 000004 MOV #QOC25,2*ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
7583 033100 170100 LDFPS RO ;SET UP FPS.
7584 033102 012700 033160 MOV #QOCTB2,RO
7585 033106 012710 033150 MOV #QOCTB1,(RO)
7586
7587 033112 170230 QOC2: STFPS 2(RO)+ ;TEST INSTRUCTION.
7588 033114 020027 033162 CMP RO,#QOCTB2+2 ;IS RO CORRECT?
7589 033120 001021 BNE QOC10 ;BRANCH IF NOT CORRECT.
7590 033122 023727 033150 106653 CMP 2*QOCTB1,#106653 ;IS THE RESULT CORRECT?
7591 033130 001025 BNE QOC15 ;BRANCH IF NOT CORRECT.
7592 033132 023727 033160 033150 CMP 2*QOCTB2,#QOCTB1 ;IS THE RESULT CORRECT?
7593 033140 001032 BNE QOC20 ;BRANCH IF NOT CORRECT.
7594 033142 000455 BR QOCDONE
7595
7596 :TEST DATA BUFFER:
7597 033144 177777 177777 QOCTB0: .WORD -1,-1
7598 033150 177777 177777 177777 QOCTB1: .WORD -1,-1,-1,-1
7599 033156 177777
7600 033160 177777 177777 QOCTB2: .WORD -1,-1
7601
7602 :REPORT RO INCORRECT.
7603 033164 010037 001242 QOC10: MOV RO,2*STMP4
7604 033170 012737 033162 001240 MOV #QOCTB2+2,2*STMP3
7605 033176
7606 033176 104377 IS: ERROR 377
7607 033200 000015 .WORD 15
7608 ;RO BAD (BUT
7609 033202 000435 BR QOCDONE ; FDST)X
7610
7611 :REPORT RESULT INCORRECT.
7612 033204 012737 106653 001240 QOC15: MOV #106653,2*STMP3 ; ST 634
```

```

7613 033212 013737 033150 001242      MOV      000CTB1,00STMP4
7614 033220      IS:
7615 033220 104377      ERROR    377
7616 033222 000016      .WORD   16
7617      :BAD DATA
7618 033224 000424      BR       QOCDONE
7619
7620
7621      ;REPORT RESULT INCORRECT.
7622 033226 012737 033160 001240 QOC20:  MOV      000CTB2,00STMP3      ; (BLT FDS*)
7623 033234 013737 033152 001242      MOV      000CTB1+2,00STMP4
7624 033242      IS:
7625 033242 104377      ERROR    377
7626 033244 000017      .WORD   17
7627 033246 000413      BR       QOCDONE
7628
7629

```

```

7630      ; IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7631      ; DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7632      ; TO THE SPURIOUS TRAP TO 4 HANDLER.

```

```

7633 033250 011604      QOC25:  MOV      (SP),R4
7634 033252 020427 033114      CMP      R4,000C2+2
7635 033256 001402      BEQ     15
7636 033260 000137 042620      JMP     00CPSPUR
7637
7638 033264 011637 001236      IS:     MOV      (SP),00STMP2
7639 033270 022626      CMP     (SP)+,(SP)+
7640 033272      2S:
7641 033272 104377      ERROR    377
7642 033274 000020      .WORD   20
7643      ; (BUT FDS)+ ST634
7644

```

```

7645 033276      QOCDONE:
7646 033276 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
7647      ;SEE IF THE USER HAS EXPRESSED
7648      ;THE DESIRE TO CHANGE THE SOFTWARE
7649      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7650      ;THE USER TYPED CONTROL G?).
7651
7652
7653

```

```

7654      ;*****
7655      ;TEST 65      DESTINATION MODES, MODE 5 (FL=0), TEST
7656      ;
7657      ; THIS IS A TEST OF DESTINATION MODE 5 USING
7658      ; THE STFPS INSTRUCTION
7659      ;
7660      ;*****

```

```

7661 033300 000004      ST65:  SCOPE
7662
7663
7664 033302      RRC1:
7665 033302 104413      LPERR
7666 033304 012700 033410      MOV      #RRC1B0,R0      ;SET UP THE LOOP ON ERROR ADDRESS.
7667 033310 012701 000006      MOV      #6,R1          ;SET UP THE DATA BUFFER.
7668 033314 012720 177777      IS:     MOV      #-1,(R0)+

```

J11

```

7669 033320 077103 SOB R1,15
7670 033322 012700 004301 MOV #004301,R0
7671 033326 012737 033356 001236 MOV #RRC2,2#STMP2
7672 033334 012737 033514 003004 MOV #RRC25,2#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
7673 033342 170100 LDFPS R0 ;SET UP FPS.
7674 033344 012700 033426 MOV #RRC2B2+2,R0
7675 033350 012760 033414 177776 MOV #RRC2B1,-2(R0)
7676
7677 033356 170250 RRC2: STFPS 2-(R0) ;TEST INSTRUCTION.
7678 033360 020027 033424 CMP R0,#RRC2B2 ;IS R0 CORRECT?
7679 033364 001021 BNE RRC10 ;BRANCH IF NOT CORRECT.
7680 033366 023727 033414 004301 CMP 2#RRC2B1,#004301 ;IS THE RESULT CORRECT?
7681 033374 001025 BNE RRC15 ;BRANCH IF NOT CORRECT.
7682 033376 023727 033424 033414 CMP 2#RRC2B2,#RRC2B1 ;IS THE RESULT CORRECT?
7683 033404 001032 BNE RRC20 ;BRANCH IF NOT CORRECT.
7684 033406 000455 BR RRCDONE

```

```

7685
7686 :TEST DATA BUFFER:
7687 033410 177777 177777 RRC2B0: .WORD -1,-1
7688 033414 177777 177777 177777 RRC2B1: .WORD -1,-1,-1,-1
7689 033422 177777
7690 033424 177777 177777 RRC2B2: .WORD -1,-1
7691

```

```

7692 :REPORT R0 INCORRECT.
7693 033430 010037 001242 RRC10: MOV R0,2#STMP4
7694 033434 012737 033424 001240 MOV #RRC2B2,2#STMP3
7695 033442
7696 033442 104377
7697 033444 000021
7698
7699 033446 000435 BR RRCDONE ;R0 BAD (BUT
; FDST)X
7700

```

```

7701 :REPORT RESULT INCORRECT.
7702 033450 012737 004301 001240 RRC15: MOV #004301,2#STMP3 ; ST 634
7703 033456 013737 033414 001242 MOV 2#RRC2B1,2#STMP4
7704 033464
7705 033464 104377
7706 033466 000022
7707
7708 033470 000424 BR RRCDONE ;BAD DATA
7709
7710

```

```

7711 :REPORT RESULT INCORRECT.
7712 033472 012737 033424 001240 RRC20: MOV #RRC2B2,2#STMP3 ;BUT FDST)
7713 033500 013737 033416 001242 MOV 2#RRC2B1+2,2#STMP4
7714 033506
7715 033506 104377
7716 033510 000023
7717
7718 033512 000413 BR RRCDONE ;(BUT GR7,FL)
;ST 357 TO 416
;INTO 417
7719
7720

```

```

7721 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7722 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7723 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7724 033514 011604 RRC25: MOV (SP),R4

```


K11

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 140
DFFPCA.P11 31-OCT-76 17:16 T65 DESTINATION MODES, MODE 5 (FL=0), TEST

```
7725 033516 020427 033360      CMP      R4, #RRC2+2
7726 033522 001402      BEQ      15
7727 033524 000137 042620      JMP      @#CPSPUR
7728
7729 033530 011637 001236      15:     MOV      (SP), @#STMP2
7730 033534 022626      CMP      (SP)+, (SP)+
7731 033536
7732 033536 104377      25:     ERROR   377
7733 033540 000024      .WORD   24
7734
7735
7736 033542
7737 033542 104412      RRCDONE:
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751 033544 000004      RSETUP
7752
7753
7754 033546
7755 033546 104413
7756 033550 012700 033660      ;GO INITIALIZE THE FPS AND STACK; AND
7757 033554 012701 000006      ;SEE IF THE USER HAS EXPRESSED
7758 033560 012720 177777      ;THE DESIRE TO CHANGE THE SOFTWARE
7759 033564 077103
7760 033566 012700 102514      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7761 033572 012737 033616 001236      ;THE USER TYPED CONTROL G?).
7762 033600 012737 033760 000004
7763 033606 170100
7764 033610 005001
7765 033612 012700 026463
7766
7767 033616 170260 005201
7768 033622 020127 000000
7769 033626 001070
7770 033630 020027 026463
7771 033634 001017
7772 033636 023727 033664 102514
7773 033644 001023
7774 033646 023727 033666 177777
7775 033654 001030
7776 033656 000456
7777
7778
7779 033660 177777 177777
7780 033664 177777 177777 177777
```

:TEST 66 DESTINATION MODES, MODE 6 (FL=0), TEST
: *
: * THIS IS A TEST OF DESTINATION MODE 6 USING
: * THE STFPS INSTRUCTION
: *
: *****
:ST66: SCOPE

SSC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #SSCTB0, R0 ;SET UP THE DATA BUFFER.
MOV #6, R1
15: MOV #-1 (R0)+
SOB R1, 15
MOV #102514, R0
MOV #SSC2, @#STMP2
MOV #SSC25, @#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS R0 ;SET UP FPS.
CLR R1
MOV #SSCTB1-5201, R0

SSC2: STFPS 5201(R0) ;TEST INSTRUCTION.
CMP R1, #0 ;WAS PC CORRECT AFTER EXECUTION?
BNE SSC30 ;BRANCH IF NOT CORRECT.
CMP R0, #SSCTB1-5201 ;IS R0 CORRECT?
BNE SSC10 ;BRANCH IF NOT CORRECT.
CMP @#SSCTB1, #102514 ;IS THE RESULT CORRECT?
BNE SSC15 ;BRANCH IF NOT CORRECT.
CMP @#SSCTB1+2, #-1 ;IS THE RESULT CORRECT?
BNE SSC20 ;BRANCH IF NOT CORRECT.
BR SSCDONE

:TEST DATA BUFFER:
SSCTB0: .WORD -1, -1
SSCTB1: .WORD -1, -1, -1, -1

7781 033672 177777

7782
7783

7784 033674 010037 001242

7785 033700 012737 026463 001240

7786 033706

7787 033706 104377

7788 033710 000025

7789

7790 033712 000440

7791

7792

7793 033714 012737 102534 001240

7794 033722 013737 033664 001242

7795 033730

7796 033730 104377

7797 033732 000026

7798

7799 033734 000427

7800

7801

7802

7803 033736 012737 177777 001240

7804 033744 013737 033666 001242

7805 033752

7806 033752 104377

7807 033754 000027

7808

7809 033756 000416

7810

7811

7812

7813

7814

7815 033760 011604

7816 033762 020427 033620

7817 033766 001402

7818 033770 000137 042620

7819

7820 033774 011637 001236

7821 034000 022626

7822 034002

7823 034002 104377

7824 034004 000030

7825

7826 034006 000402

7827

7828

7829 034010

7830 034010

7831 034010 104377

7832 034012 000031

7833

7834

7835

7836

:REPORT RD INCORRECT.

SSC10: MOV R0, @STMP4

MOV #SSCTB1-5201, @STMP3

IS: ERROR 377

.WORD 25

;RO BAD

BR SSCDONE

:REPORT RESULT INCORRECT.

SSC15: MOV #102534, @STMP3

MOV @SSCTB1, @STMP4

IS: ERROR 377

.WORD 26

;BAD DATA

BR SSCDONE

:REPORT RESULT INCORRECT.

SSC20: MOV #-1, @STMP3

MOV @SSCTB1+2, @STMP4

IS: ERROR 377

.WORD 27

;(BUT GR7, FL)

;ST 357 TO 416

;INTO 417

BR SSCDONE

:IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
:DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED. IF NOT GO
:TO THE SPURIOUS TRAP TO 4 HANDLER.

SSC25: MOV (SP), R4

CMP R4, #SSC2+2

BEQ IS

JMP @CPSPUR

IS: MOV (SP), @STMP2

CMP (SP)+, (SP)+

2S: ERROR 377

.WORD 30

;(BUT FDST)+ ST634

BR SSCDONE

:REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.

SSC30:

IS: ERROR 377

.WORD 31

;PC NOT

;INCREMENTED

;BY 2

M11

7837 034014
7838 034014 104412
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852 034016 000004
7853
7854 034020
7855 034020 104413
7856 034022 012700 034140
7857 034026 012701 000010
7858 034032 012720 177777
7859 034036 077103
7860 034040 012700 103747
7861 034044 012737 034076 001236
7862 034052 012737 034244 000004
7863 034060 170100
7864 034062 005001
7865 034064 012700 026753
7866 034070 012760 034144 005201
7867
7868 034076 170270 005201
7869 034102 022701 000000
7870 034106 001072
7871 034110 020027 026753
7872 034114 001021
7873 034116 023727 034144 103747
7874 034124 001025
7875 034126 023727 034146 177777
7876 034134 001032
7877 034136 000460
7878
7879
7880 034140 177777 177777
7881 034144 177777 177777 177777
7882 034152 177777
7883 034154 177777 177777
7884
7885
7886 034160 010037 001242
7887 034164 012737 026753 001240
7888 034172
7889 034172 104377
7890 034174 000032
7891
7892 034176 000440

SSCDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).

*TEST 67 DESTINATION MODES, MODE 7 (FL=0), TEST
*
* THIS IS A TEST OF DESTINATION MODE 7 USING
* THE STFPS INSTRUCTION
*

↑ST67: SCOPE

TTC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #TTCB0,RO ;SET UP THE DATA BUFFER.
MOV #10,R1
IS: MOV #-1,(RO)+
SOB R1,IS
MOV #103747,RO
MOV #TTC2,@#STMP2
MOV #TTC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS RO ;SET UP FPS.
CLR R1
MOV #TTCB2-5201,RO
MOV #TTCB1,5201(RO)
TTC2: STFPS @5201(RO) ;TEST INSTRUCTION.
CMP #0,R1 ;WAS PC CORRECT AFTER EXECUTION?
BNE TTC30 ;BRANCH IF NOT CORRECT.
CMP R0,#TTCB2-5201 ;IS R0 CORRECT?
BNE TTC10 ;BRANCH IF NOT CORRECT.
CMP @#TTCB1,#103747 ;IS THE RESULT CORRECT?
BNE TTC15 ;BRANCH IF NOT CORRECT.
CMP @#TTCB1+2,#-1 ;IS THE RESULT CORRECT?
BNE TTC20 ;BRANCH IF NOT CORRECT.
BR TTCDONE

:TEST DATA BUFFER:
↑TTCB0: .WORD -1,-1
TTCB1: .WORD -1,-1,-1,-1
TTCB2: .WORD -1,-1

:REPORT R0 INCORRECT.
↑TTC10: MOV R0,@#STMP4
MOV #TTCB2-5201,@#STMP3
IS: ERROR 377
.WORD 32 ;R0 BAD
BR TTCDONE

N11

```

7893
7894
7895 :REPORT RESULT INCORRECT.
7896 034200 012737 103747 001240 TTC15: MOV #103747,2*STMP3
7897 034206 013737 034144 001242 MOV 3*TTCTB1,2*STMP4
7898 034214
7899 034214 104377
7900 034216 000033
7901
7902 034220 000427 BR TTCDONE ;BAD DATA
7903
7904
7905 :REPORT RESULT INCORRECT.
7906 034222 012737 177777 001240 TTC20: MOV #-1,2*STMP3
7907 034230 013737 034146 001242 MOV 2*TTCTB1+2,2*STMP4
7908 034236
7909 034236 104377
7910 034240 000034
7911
7912 034242 000416 BR TTCDONE ;(BUT GR7,FL)
7913 ;ST 357 TO 416
7914 ;INTO 417
7915
7916 :IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7917 :DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED. IF NOT GO
7918 :TO THE SPURIOUS TRAP TO 4 HANDLER.
7919 034244 011604 TTC25: MOV (SP),R4
7920 034246 020427 034100 CMP R4,#TTC2+2
7921 034252 001402 BEQ 15
7922 034254 000137 042620 JMP 2*CPSPUR
7923 034260 011637 001236
7924 034264 022626
7925 034266 104377
7926 034270 000035
7927
7928 034272 000402 BR TTCDONE ;(BUT FSDT)+ ST634
7929
7930 :REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
7931 034274 TTC30:
7932 034274
7933 034274 104377
7934 034276 000036
7935
7936 ;PC NOT
7937 034300 TTCDONE: RSETUP ;INCREMENTED
7938 034300 104412 ;GO INITIALIZE THE FPS AND STACK; AND
7939 ;SEE IF THE USER HAS EXPRESSED
7940 ;THE DESIRE TO CHANGE THE SOFTWARE
7941 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7942 ;THE USER TYPED CONTROL G?).
7943
7944 ;*****
7945 ;*TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST
7946 ;*
7947 ;* THIS IS A TEST OF DESTINATION MODE
7948 ;* 2 USING STCOL WITH REGISTER 0

```

```

000004
000004 104413
000004 012700 000300
000004 034364
000004 034334 001236
000004 034376
000004 175420
000004 020027 034402
000004 001420
000004 010037 001242
000004 012737 034402 001240
000004 104377
000004 000037
000004 034362 000410
000004 034364 000000 000000 000000
000004 034372 000000
000004 034374 177777
000004 034376 177777 177777 177777
000004 034404
000004 034404 104412
000004 034406 000004
000004 034410
000004 034410 104413
000004 034412 012700 000300
000004 034416 170100
000004 034420 012700 034470
000004 034424 172410
000004 034426 012737 034440 001236
000004 034434 012700 034506

```

```

*****
*ST70: SCOPE
*UUC1:
  L PERR                               :SET UP THE LOOP ON ERROR ADDRESS.
  MOV #300,RO                          :SET UP FPS.
  LDFPS RO
  MOV #UUCTP1,RO                        :SET UP THE ACC OPERAND.
  LDD (RO),ACC
  MOV #UUC2,#STMP2
  MOV #UUCBFD,RO
*UUC2: STCDL ACC,(RO)+                 :TEST INSTRUCTION.
  CMP RO,#UUCBFD+4                     :IS RO CORRECT?
  BEQ UUCDONE                          :BRANCH IF CORRECT.
:REPORT RO INCORRECT.
*UUC3: MOV RO,#STMP4
  MOV #UUCBFD+4,#STMP3
IS:
  ERROR 377
  .WORD 37
                                           ;RO NOT INCR BY 4
  BR UUCDONE
:TEST DATA BUFFER:
*UUCTP1: .WORD 0,0,0,0
*UUCBFD: .WORD -1,-1,-1
*UUCDONE:
  RSETUP
:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER (HAS
:THE USER TYPED CONTROL G?).

```

```

*****
*TEST 71 DESTINATION MODES, MODE 4 (FL=1), TEST
*
* THIS IS A TEST OF DESTINATION MODE
* 4 USING STCDL WITH REGISTER 0
*****
*ST71: SCOPE
*VVC1:
  L PERR                               :SET UP THE LOOP ON ERROR ADDRESS.
  MOV #300,RO                          :SET UP FPS.
  LDFPS RO
  MOV #VVC1,RO                          :SET UP THE ACC OPERAND.
  LDD (RO),ACC
  MOV #VVC2,#STMP2
  MOV #VVCBFD+4,RO

```

```

034440 175440 VVC2: STCDL ACD, -(R0) ;TEST INSTRUCTION.
034440 020027 034502 CMP R0, @VVCBFD ;IS RC CORRECT?
034446 001423 BEQ VVC DONE
;REPORT RC INCORRECT.
034450 010037 001242 VVC3: MOV R0, @STMP4
034454 012737 034502 001240 MOV @VVCBFD, @STMP3
18: ERROR 377
.WORD 40 ;RC NOT DECR BY 4
034466 000410 BR VVC DONE
;TEST DATA BUFFER:
034473 000000 000000 000000 VVC TP1: .WORD 0,0,0,0
034477 000000 177777 177777 VVCBFD: .WORD -1, -1, -1
034480 177777 177777 177777 VVC DONE: RSETJP
034483 104412 ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER HAS
;THE USER TYPED CONTROL G'.

```

TEST 72 STCDI AND STCDL TEST

* THIS IS A TEST OF THE STCDI AND
* STCDL INSTRUCTIONS. NOTE THAT A
* SUBROUTINE, STCSUB, IS USED TO
* SET UP THE OPERANDS, EXECUTE THE STC
* INSTRUCTION AND CHECK THE RESULT.

034512 000004

ST72: SCOPE

;FIRST TEST STC WITH EXP=100 (EXCESS 200)

```

VVC1:
.PERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC, @STCSUB ;GO EXECUTE THE INSTRUCTION.
18: .WORD 20000,0,0,0 ;ACD OPERAND.
25: .WORD 0,0 ;EXPECTED RESULT.
35: .WORD -1,-1 ;ERROR RES.
45: 40300 ;FPS BEFORE EXECUTION.
40304 ;FPS AFTER EXECUTION.
140304 ;ANTICIPATED ERRONEOUS FPS.
-1 ;REPORT RESULT INCORRECT.
55: ERROR 322 ;RESULT INCORP.
BR 65 ;EITHER (BUT FLAG)
ERROR 325 ;ST 662
;OR CLEAR FLAG
;ST 774

```

```

034515 104413
034516 004737 035662
034522 020000 000000 000000
034530 000000
034532 000000 000000
034538 177777 177777
034542 040300
034544 040304
034546 140304
034550 177777
034552 104322
034554 000401
034556 104325
034560

```

```

8080 :EXP=0 (OCT) FL=1 FIC=0
8081 MWC2: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8082 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8083 .WORD 40000,0,0,0 ;AC ;ACD OPERAND.
8084 15: 15: .WORD 0,0 ;EXPECTED RESULT.
8085 25: 25: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
8086 35: 35: 40313 ;FPS BEFORE EXECUTION.
8087 45: 45: 40304 ;FPS AFTER EXECUTION.
8088 140304 ;ANTICIPATED ERRONEOUS FPS.
8089 -1 ;EXPECTED FEC.
8090 55: 55: ERROR 322 ;REPORT RESULT INCORRECT.
8091 BR 65
8092 ERROR 326 ;REPORT FPS INCORRECT.
8093 65:
8094
8095 :EXP=37 (OCT) FL=1 FIC=1
8096 MWC4: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8097 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8098 .WORD 47667,75757,157737,167773 ;ACD OPERAND.
8099 15: 15: .WORD 55675,173757 ;EXPECTED RESULT.
8100 25: 25: .WORD 122102,004021 ;ANTICIPATED ERRONEOUS RESULT.
8101 35: 35: 40717 ;FPS BEFORE EXECUTION.
8102 45: 45: 40700 ;FPS AFTER EXECUTION.
8103 140705 ;ANTICIPATED ERRONEOUS FPS.
8104 -1 ;EXPECTED FEC.
8105 55: 55: ERROR 327 ;(BUT ENBT) ST 632
8106 BR 65
8107 ERROR 326 ;REPORT FPS INCORRECT.
8108 65:
8109
8110 :EXP=40 (OCT) FL=1 FIC=1
8111 MWC5: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8112 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8113 .WORD 50000,0,0,0 ;ACD OPERAND.
8114 15: 15: .WORD 0,0 ;EXPECTED RESULT.
8115 25: 25: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
8116 35: 35: 40700 ;FPS BEFORE EXECUTION.
8117 45: 45: 140705 ;FPS AFTER EXECUTION.
8118 140705 ;ANTICIPATED ERRONEOUS FPS.
8119 40705
8120 6 ;EXPECTED FEC.
8121 55: 55: ERROR 322 ;REPORT RESULT INCORRECT.
8122 BR 65
8123 ERROR 330 ;(BUT-FIC) ST 004 ;REPORT FPS INCORRECT.
8124 65: ;TO 305 INTO ;315
8125
8126 :EXP=40 (OCT) FL=1 FIC=0
8127 MWC6: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.

```

035000	035000	035662	000000	15:	JSR	PC,2#STCSUB	:GO EXECUTE THE INSTRUCTION.
035000	035000	000000	000000		.WORD	5000,0,0,0	:ACD OPERAND.
035000	035000	000000	000000	25:	.WORD	0,0	:EXPECTED RESULT.
035000	035000	000000	000000	35:	.WORD	-1,-1	:ANTICIPATED ERRONEOUS RESULT.
035000	035000	000000	000000	45:	40312		:FPS BEFORE EXECUTION.
035000	035000	000000	000000		40305		:FPS AFTER EXECUTION.
035000	035000	000000	000000		140305		:ANTICIPATED ERRONEOUS FPS.
035000	035000	000000	000000		-1		:EXPECTED FEC.
035000	035000	000000	000000	55:	ERROR	322	:REPORT RESULT INCORRECT.
035000	035000	000000	000000		BR	65	
035000	035000	000000	000000	65:	ERROR	331	: (BUT FIC) ST 004 TO :315 INTO 305
;EXP=30 (OCT) FL=1 FIC=1 WMC7:							
035000	035000	035662	000000	15:	LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
035002	004737	035662	000000		JSR	PC,2#STCSUB	:GO EXECUTE THE INSTRUCTION.
035006	046000	000001	000000		.WORD	46000,1,0,0	:ACD OPERAND.
035014	000000	000001	000000	25:	.WORD	200,1	:EXPECTED RESULT.
035016	000200	000001	000000	35:	.WORD	-1,-1	:ANTICIPATED ERRONEOUS RESULT.
035022	177777	177777	177777	45:	40700		:FPS BEFORE EXECUTION.
035026	040700				40700		:FPS AFTER EXECUTION.
035030	040700				-1		:ANTICIPATED ERRONEOUS FPS.
035032	177777				-1		:EXPECTED FEC.
035034	177777			55:	ERROR	322	:REPORT RESULT INCORRECT.
035036	104322				BR	65	
035040	000401			65:	ERROR	323	:REPORT FPS INCORRECT.
035042	104323						
035044							
;EXP=27 (OCT) FL=1 FIC=1 WMC8:							
035044	035044	035662	000000	15:	LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
035046	004737	035662	000000		JSR	PC,2#STCSUB	:GO EXECUTE THE INSTRUCTION.
035052	045600	000001	000000		.WORD	45600,1,0,0	:ACD OPERAND.
035060	000000	000000	000000	25:	.WORD	100,0	:EXPECTED RESULT.
035062	000100	000000	000000	35:	.WORD	-1,-1	:ANTICIPATED ERRONEOUS RESULT.
035066	177777	177777	177777	45:	40707		:FPS BEFORE EXECUTION.
035072	040707				40700		:FPS AFTER EXECUTION.
035074	040700				-1		:ANTICIPATED ERRONEOUS FPS.
035076	177777				-1		:EXPECTED FEC.
035100	177777			55:	ERROR	322	:REPORT RESULT INCORRECT.
035102	104322				BR	65	
035104	000401			65:	ERROR	323	:REPORT FPS INCORRECT.
035106	104323						
035110							
;EXP=17 (OCT) FL=0 FIC=1 WMC9:							
035110	035110	035662	000000	15:	LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
035112	004737	035662	000000		JSR	PC,2#STCSUB	:GO EXECUTE THE INSTRUCTION.
035116	043600	000000	000000		.WORD	43600,0,0,0	:ACD OPERAND.
035124	000000	000000	000000	25:	.WORD	40000,-1	:EXPECTED RESULT.
035126	040000	177777	177777	35:	.WORD	0,-1	:ANTICIPATED ERRONEOUS RESULT.
035132	000000	177777	177777				


```

8229 035324 000401 BR 65 :ST 275 TO 074
8230 035326 104323 ERROR 323 :INTO 274
8231 035330 65:
8232
8233 :EXP=37 (OCT), FL=1, FIC=1, AC NEG
8234 WWC13:
8235 035330 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
8236 035332 004737 035662 JSR PC,2#STCSUB :GO EXECUTE THE INSTRUCTION.
8237 035336 147600 000000 001000 15: .WORD 147600,0,1000,0 :ACD OPERAND.
8238 035344 000000
8239 035346 137777 177777 25: .WORD 137777,177777 :EXPECTED RESULT.
8240 035352 140000 177777 35: .WORD 140000,177777 :ANTICIPATED ERRONEOUS RESULT.
8241 035356 040707 45: 40707 :FPS BEFORE EXECUTION.
8242 035360 040710 40710 :FPS AFTER EXECUTION.
8243 035362 177777 -1 :ANTICIPATED ERRONEOUS FPS.
8244 035364 177777 -1 :EXPECTED FEC.
8245 035366 104340 55: ERROR 340 :(BUT COUT) ST 375
8246 035370 000401 BR 65 :TO 274 INTO 074
8247 035372 104323 ERROR 323 :REPORT FPS INCORRECT.
8248 035374 65:
8249
8250 :EXP=41 (OCT), AC NEG, FL=1, FIC=1
8251 035374 WWC14:
8252 035374 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
8253 035376 004737 035662 JSR PC,2#STCSUB :GO EXECUTE THE INSTRUCTION.
8254 035402 150200 000000 000000 15: .WORD 150200,0,0,0 :ACD OPERAND.
8255 035410 000000
8256 035412 000000 000000 25: .WORD 0,0 :EXPECTED RESULT.
8257 035416 177777 177777 35: .WORD -1,-1 :ANTICIPATED ERRONEOUS RESULT.
8258 035422 040700 45: 40700 :FPS BEFORE EXECUTION.
8259 035424 140705 140705 :FPS AFTER EXECUTION.
8260 035426 177777 -1 :ANTICIPATED ERRONEOUS FPS.
8261 035430 000006 6 :EXPECTED FEC.
8262 035432 104322 55: ERROR 322 :REPORT RESULT INCORRECT.
8263 035434 000401 BR 65
8264 035436 104341 ERROR 341 :(BUT EZBT) ST 377
8265 035440 65:
8266 :EXP=40 (OCT), AC NEG, FL=1, FIC=1
8267 035440 WWC15:
8268 035440 104413 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
8269 035442 004737 035662 JSR PC,2#STCSUB :GO EXECUTE THE INSTRUCTION.
8270 035446 150000 000001 000000 15: .WORD 150000,1,0,0 :ACD OPERAND.
8271 035454 000000
8272 035456 000000 000000 25: .WORD 0,0 :EXPECTED RESULT.
8273 035462 100000 177600 35: .WORD 100000,-200 :ANTICIPATED ERRONEOUS RESULT.
8274 035466 040700 45: 40700 :FPS BEFORE EXECUTION.
8275 035470 140705 140705 :FPS AFTER EXECUTION.
8276 035472 040700 40700 :ANTICIPATED ERRONEOUS FPS.
8277 035474 000006 6 :EXPECTED FEC.
8278 035476 104342 55: ERROR 342 :(BUT COUT) ST 360
8279 035500 000401 BR 65 :TO 654 INTO 454
8280 035502 104323 ERROR 323 :REPORT FPS INCORRECT.
8281 035504 65:
8282
8283 :EXP=40, AC NEGATIVE, FL=1, FIC=1
8284 035504 WWC16:

```

H12

```

8285 035504 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8286 035506 004737 035662 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8287 035512 150001 000000 000000 15: .WORD 150001,0,0,0 ;ACO OPERAND.
8288 035520 000000 25: .WORD 0,0 ;EXPECTED RESULT.
8289 035522 000000 000000 35: .WORD 77400,0 ;ANTICIPATED ERRONEOUS RESULT.
8290 035526 077400 000000 45: 40700 ;FPS BEFORE EXECUTION.
8291 035532 040700 45: 140705 ;FPS AFTER EXECUTION.
8292 035534 140705 ;ANTICIPATED ERRONEOUS FPS.
8293 035536 177777 -1 ;EXPECTED FEC.
8294 035540 000006 55: ERROR 343 ;REPORT RESULT INCORRECT.
8295 035542 104343 BR 65
8296 035544 000401 BR 65
8297 035546 104323 65: ERROR 323 ;REPORT FPS INCORRECT.
8298 035550
8299
8300

```

```

8301 ;EXP 40 (OCT), AC MOST NEG LONG INT, FL=1
8302 ;FIC=1
8303 WWC17:

```

```

8304 035550 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8305 035552 004737 035662 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8306 035556 150000 000000 000000 15: .WORD 150000,0,0,0 ;ACO OPERAND.
8307 035564 000000 25: .WORD 100000,0 ;EXPECTED RESULT.
8308 035566 100000 000000 35: .WORD 0,0 ;ANTICIPATED ERRONEOUS RESULT.
8309 035572 000000 000000 45: 40700 ;FPS BEFORE EXECUTION.
8310 035576 040700 45: 40710 ;FPS AFTER EXECUTION.
8311 035600 040710 ;ANTICIPATED ERRONEOUS FPS.
8312 035602 140705 -1 ;EXPECTED FEC.
8313 035604 177777 55: ERROR 344 ;(BUT NBIT) ST 654
8314 035606 104344 BR 65 ;OR (BUT COUT) ST 454
8315 035610 000401 BR 65
8316 035612 104323 65: ERROR 323 ;REPORT FPS INCORRECT.
8317 035614
8318
8319 ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
8320

```

```

8321 035614 WWC18:
8322 035614 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8323 035616 004737 035662 JSR PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
8324 035622 144000 000001 000000 15: .WORD 144000,1,0,0 ;ACO OPERAND.
8325 035630 000000 25: .WORD 100000,-1 ;EXPECTED RESULT.
8326 035632 100000 177777 35: .WORD 100000,177400 ;ANTICIPATED ERRONEOUS RESULT.
8327 035636 100000 177400 45: 40600 ;FPS BEFORE EXECUTION.
8328 035642 040600 45: 40610 ;FPS AFTER EXECUTION.
8329 035644 040610 ;ANTICIPATED ERRONEOUS FPS.
8330 035646 140605 -1 ;EXPECTED FEC.
8331 035650 177777 55: ERROR 345 ;(BUT FL) ST 633
8332 035652 104345 BR 65 ;TO 655 INTO 654
8333 035654 000401 BR 65
8334 035656 104323 65: ERROR 323 ;REPORT FPS INCORRECT.
8335
8336 035660 000534 65: BR WWC DONE
8337

```

```

8338 ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
8339 ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS. A CALL
8340 ;TO IT IS MADE THUS:

```

8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396

035662 012601
035664 012700 000200
035670 170100
035672 010100
035674 172410
035676 012702 036142
035702 012700 000034
035706 012722 177777
035712 077003
035714 016100 000020
035720 170100
035722 012737 035734 001236
035730 012700 036142
035734 175410
035736 170204
035740 170305
035742 010102
035744 010237 001240
035750 062702 000010
035754 010237 001244
035760 012737 036142 001242
035766 010437 001250
035772 016137 000022 001252
036000 010102
036002 062702 000010
036006 012700 036142

```

JSR      PC,@STCSUB
ACARG:   .WORD   X,X,X,X      ;AC OPERAND
RES:     .WORD   X,X          ;EXPECTED RESULT
ERRES:   .WORD   X,X          ;ERROR RESULT
FPSB:    .WORD   X            ;FPS BEFORE EXECUTION
FPSA:    .WORD   X            ;FPS AFTER EXECUTION
ERFPS:   .WORD   X            ;ERROR FPS.
FEC:     .WORD   X            ;EXPECTED FEC
ERR1:    ERROR   X            ;DATA ERROR.
          BR      CONT
ERR2:    ERROR   X            ;FPS ERROR.
CONT:    CONT

```

THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
THE STCDI OR STCDL INSTRUCTION IS EXECUTED.
THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCSUB
COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCSUB WILL RETURN
TO THE ERROR CALL AT ERR2, OTHERWISE STCSUB ITSELF
REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
STCDI OR STCDL IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCSUB
WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCSUB WILL
REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

```

STCSUB:  MOV      (SP)+,R1      ;GET A POINTER TO THE ARGUMENTS.
          MOV      #200,R0     ;SET UP THE ACO OPERAND.
          LDFPS   R0
          MOV      R1,R0
          LDD     (R0),ACO
          MOV      #STCIBF,R2  ;INITIALIZE THE OUT PUT BUFFER.
          MOV      #4,R0
1$:      MOV      #-1,(R2)+
          SOB     R0,1$
          MOV      20(R1),R0   ;SET THE FPS.
          LDFPS   R0
          MOV      #25,@STMP2
          MOV      #STCIBF,R0
2$:      STCDL   ACO,(R0)     ;TEST INSTRUCTION.

          STFPS   R4           ;GET THE FPS.
          STST   R5           ;GET THE FEC.
          MOV      R1,R2
          MOV      R2,@STMP3
          ADD     #10,R2
          MOV      R2,@STMP5
          MOV      #STCIBF,@STMP4
          MOV      R4,@STMP7
          MOV      22(R1),@STMP10
          MOV      R1,R2
          ADD     #10,R2
          MOV      #STCIBF,R0  ;SEE IF THE RESULT IS CORRECT.

```

```

8397 036012 012703 000002
8398 036016 022022
8399 036020 001014
9400 036022 077303
8401 036024 016102 000022
8402 036030 020204
8403 036032 001025
8404 036034 005702
8405 036036 100003
8406 036040 026105 000026
8407 036044 001027
8408
8409 036046 000161 000036
8410
8411
8412 036052 010102
8413 036054 062702 000014
8414 036060 012700 036142
8415 036064 012703 000002
8416 036070 022022
8417 036072 001003
8418 036074 077303
8419 036076 000161 000030
8420 036102
8421
8422 036102 104322
8423 036104 000760
8424
8425
8426 036106 020461 000024
8427 036112 001002
8428 036114 000161 000034
8429 036120
8430
8431 036120 104323
8432 036122 000751
8433
8434
8435 036124 016137 000026 001256
8436 036132 010537 001254
8437 036136 104324
8438 036140 000742
8439
8440
8441 036142 177777 177777 177777
8442 036150 177777
8443
8444 036152
8445 036152 104412
8446
8447
8448
8449
8450
8451
8452

```

```

MOV #2,R3
35: CMP (R0)+,(R2+
BNE 155
SOB R3,35
MOV 22(R1),R2
CMP R2,R4 ;SEE IF THE FPS IS CORRECT.
BNE 205 ;BRANCH IF INCORRECT.
TST R2
BPL 45
CMP 26(R1),R5 ;SEE IF THE FEC IS CORRECT.
BNE 255 ;BRANCH IF INCORRECT.
45: JMP 36(R1) ;RETURN.
:DATA ERROR:
:SEE IF THE FAILURE WAS ANTICIPATED.
155: MOV R1,R2
ADD #14,R2
MOV #STCIBF,R0
MOV #2,R3
165: CMP (R0)+,(R2)+
BNE 175
SOB R3,165
JMP 30(R1)
175:
:FAILURE WAS NOT ANTICIPATED SO REPORT INCORRECT RESULT HERE.
185: ERROR 322 ;DATA BAD
BR 45
:FPS INCORRECT, SO SEE IF FAILURE WAS ANTICIPATED.
205: CMP R4,24(R1)
BNE 215
JMP 34(R1)
215:
:NOT ANTICIPATED SO REPORT BAD FPS HERE.
225: ERROR 323 ;FPS BAD
BR 45
:REPORT INCORRECT FEC.
255: MOV 26(R1),#STMP12
MOV R5,#STMP11
265: ERROR 324
BR 45
:DATA BUFFER:
STCIBF: .WORD -1,-1,-1,-1
WWCDONE:
RSETUP
;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
;*****

```

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

036154 000004

036156
036156 104413
036156 004737 035662
036164 047777 177777 177777
036172 177777
036174 077777 177600
036200 077777 177777
036204 040100
036206 040100
036210 177777
036212 177777
036214 104346
036216 000401
036220 104323
036222
036222 104412

```
;*TEST 73      STCFL AND STCFI TEST
;*
;* THIS IS A TEST OF STCFL AND STCFI.  IT
;* MAKES USE OF THE SAME SUBROUTINE, STCSJB,
;* WHICH WAS USED TO TEST STCDL AND STCDI.
;*****
†ST73:  SCOPE

;EXPONENT=37.  FL=1
XXC1:
      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      JSR            PC,2#STCSUB ;GO EXECUTE THE INSTRUCTION.
1$:   .WORD          47777,-1,-1,-1 ;AC0 OPERAND.

      .WORD          77777,177600 ;EXPECTED RESULT.
2$:   .WORD          77777,177777 ;ANTICIPATED ERRONEOUS RESULT.
3$:   .WORD          40100        ;FPS BEFORE EXECUTION.
4$:   .WORD          40100        ;FPS AFTER EXECUTION.
      -1             ;ANTICIPATED ERRONEOUS FPS.
      -1             ;EXPECTED FEC.
5$:   ERROR          346          ;X11(1,0)+0 ST 773X
      BR             6$
      ERROR          323
6$:

XXCDONE:
      RSETUP         ;GO INITIALIZE THE FPS AND STACK; AND
                    ;SEE IF THE USER HAS EXPRESSED
                    ;THE DESIRE TO CHANGE THE SOFTWARE
                    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                    ;THE USER TYPED CONTROL G?).
```

```
;*****
;*TEST 74      STEXP TEST
;*
;* THIS IS A TEST OF THE STEXP
;* INSTRUCTION
;*****
†ST74:  SCOPE

; EXP = 100 (EXCESS 200)
YYC1:
      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      JSR            PC,2#STXSU3
1$:   .WORD          20000,0,0,0 ;AC

      -100          ;EXP RES
2$:   .WORD          52525        ;ERROR EXP.
3$:   .WORD          40000        ;FPSB
4$:   .WORD          40010        ;FPSA
      .WORD          40000        ;ERROR FPS
5$:   ERROR          347          ;BAD EXP
```

```

8509 036260 000401 BR 65
8510 036262 104352 ERROR 352 ;+(BUT ENBT) ST 376
8511 036264 65:
8512 ; EXP = 200 (EXCESS 200)
8513 YYC2:
8514 036264 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8515 036264 104413 JSR PC,2#STXSUB ;GO EXECUTE THE INSTRUCTION.
8516 036266 004737 036514 000000 15: .WORD 40000,0,0,0 ;ACD OPERAND.
8517 036272 040000
8518 036300 000000
8519 036302 000000 25: 0 ;EXPECTED EXPONENT RESULT.
8520 036304 052525 35: 52525 ;ANTICIPATED ERRONEOUS RESULT.
8521 036306 040000 45: 40000 ;FPS BEFORE EXECUTION.
8522 036310 040004 45: 40004 ;FPS AFTER EXECUTION.
8523 036312 040000 45: 40000 ;ANTICIPATED ERRONEOUS FPS.
8524 036314 104347 55: ERROR 347 ;REPORT RESULT INCORRECT.
8525 036316 000401 BR 65
8526 036320 104353 ERROR 353 ;(BUT EZBT) ST 071
8527 ;TO 072 INT 272
8528 036322 65:
8529
8530 ; EXP = 201 (EXCESS 200)
8531 YYC3:
8532 036322 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8533 036322 104413 JSR PC,2#STXSUB ;GO EXECUTE THE INSTRUCTION.
8534 036324 004737 036514 000000 15: .WORD 40200,0,0,0 ;ACD OPERAND.
8535 036330 040200
8536 036336 000000
8537 036340 000001 25: 1 ;EXPECTED EXPONENT RESULT.
8538 036342 052525 35: 52525 ;ANTICIPATED ERRONEOUS RESULT.
8539 036344 040000 45: 40000 ;FPS BEFORE EXECUTION.
8540 036346 040000 45: 40000 ;FPS AFTER EXECUTION.
8541 036350 040004 45: 40004 ;ANTICIPATED ERRONEOUS FPS.
8542 036352 104347 55: ERROR 347 ;REPORT RESULT INCORRECT.
8543 036354 000401 BR 65
8544 036356 104354 ERROR 354 ;(BUT EZBT) ST 071
8545 036360 ;TO 272 INTO 072
8546 65:
8547 ; EXP = 375 (EXCESS 200)
8548 YYC4:
8549 036360 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8550 036360 104413 JSR PC,2#STXSUB ;GO EXECUTE THE INSTRUCTION.
8551 036362 004737 036514 000000 15: .WORD 77200,0,0,0 ;ACD OPERAND.
8552 036366 077200
8553 036374 000000
8554 036376 000175 25: 175 ;EXPECTED EXPONENT RESULT.
8555 036400 052525 35: 52525 ;ANTICIPATED ERRONEOUS RESULT.
8556 036402 040000 45: 40000 ;FPS BEFORE EXECUTION.
8557 036404 040000 45: 40000 ;FPS AFTER EXECUTION.
8558 036406 040010 45: 40010 ;ANTICIPATED ERRONEOUS FPS.
8559 036410 104347 55: ERROR 347 ;REPORT RESULT INCORRECT.
8560 036412 000401 BR 65
8561 036414 104355 ERROR 355 ;(BUT ENBT) ST 376
8562 036416 ;TO 471 INTO 071
8563 65:
8564 ; EXP = 1 (EXCESS 200)

```

```

8565
8566 036416
8567 036416 104413
8568 036420 004737 036514
8569 036424 000200 000000 000000 1S: .WORD 200,0,0,0
8570 036432 000000
8571 036434 177601 2S: -177
8572 036436 052525 3S: 52525 ; ANTICIPATED ERRONEOUS RESULT.
8573 036440 040000 4S: 40000 ; FPS BEFORE EXECUTION.
8574 036442 040010 ; FPS AFTER EXECUTION.
8575 036444 040000 ; ANTICIPATED ERRONEOUS FPS.
8576 036446 104347 5S: ERROR 347 ; REPORT RESULT INCORRECT.
8577 036450 000401 BR 6S
8578 036452 104352 ERROR 352 ; REPORT FPS INCORRECT.
8579 036454

```

; EXP = 156 (EXCESS 200)

```

8581
8582
8583 036454
8584 036454 104413
8585 036456 004737 036514
8586 036462 033400 000000 000000 1S: .WORD 33400,0,0,0
8587 036470 000000
8588 036472 177756 2S: -22
8589 036474 052525 3S: 52525 ; ANTICIPATED ERRONEOUS RESULT.
8590 036476 047707 4S: 47707 ; FPS BEFORE EXECUTION.
8591 036500 047710 ; FPS AFTER EXECUTION.
8592 036502 177777 ; ANTICIPATED ERRONEOUS FPS.
8593 036504 104347 5S: ERROR 347 ; REPORT RESULT INCORRECT.
8594 036506 000401 BR 6S
8595 036510 104350 ERROR 350 ; REPORT FPS INCORRECT.
8596
8597 036512 000510 6S: BR YYCDONE
8598
8599

```

```

; THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
; THE STXP INSTRUCTION AND CHECK THE RESULTS. A CALL
; TO IT IS MADE THUS:

```

```

JSR PC, @STXSUB
ACARG: .WORD X,X,X,X ; AC OPERAND
RES: .WORD X ; EXPECTED RESULT
ERRS: .WORD X ; ERROR RESULT
FPSB: .WORD X ; FPS BEFORE EXECUTION
FPSA: .WORD X ; FPS AFTER EXECUTION
ERFPS: .WORD X ; ERROR FPS.
ERR1: ERROR X ; DATA ERROR.
BR CONT
ERR2: ERROR X ; FPS ERROR.
CONT: ; RETURN ADDRESS

```

```

; THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
; THE STXP INSTRUCTION IS EXECUTED.
; THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
; COMPARED WITH FPSA IF THIS TOO IS CORRECT STXSUB RETURNS CONTROL
; TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STXSUB
; COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STXSUB WILL RETURN

```

```

8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8620

```



```

8621
8622
8623
8624
8625
8626
8627
8628
8629
8630 036514 012601
8631 036516 010102
8632 036520 010237 001240
8633 036524 062702 000010
8634 036530 012237 001244
8635 036534 012737 036602 001236
8636 036542 012737 123456 036722
8637 036550 012737 076543 036724
8638 036556 012700 000200
8639 036562 170100
8640 036564 010100
8641 036566 172410
8642 036570 016100 000016
8643 036574 170100
8644 036576 012700 036722
8645 036602 175010
8646 036604 170204
8647 036606 010437 001250
8648 036612 016137 000016 001252
8649 036620 013737 036722 001242
8650 036626 026137 000010 036722
8651 036634 001411
8652 036636 026137 000012 036722
8653 036644 001002
8654 036646 000161 000022
8655
8656
8657 036652
8658 036652 104347
8659 036654 000161 000030
8660
8661 036660 020461 000016
8662 036664 001407
8663 036666 020461 000020
8664 036672 001002
8665 036674 000161 000026
8666
8667
8668 036700
8669 036700 104350
8670 036702 000764
8671
8672
8673 036704 022737 076543 036724
8674 036712 001760
8675 036714 104351
8676 036716 000756

```

```

: TO THE ERROR CALL AT ERR2, OTHERWISE STXSUB ITSELF
: REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
: STXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
: ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
: THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STXSUB
: WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
: RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STXSUB WILL
: REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

STXSJB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
MOV R1,R2
MOV R2,2*STMP3
ADD #0,R2
MOV (R2)+,2*STMP5
MOV #15,2*STMP2
MOV #123456,2*STXBF
MOV #76543,2*STXBF+2
MOV #200,R0
LDFPS R0
MOV R1,R0 ;SET UP THE ACO OPERAND.
LDD (R0),ACO
MOV 16(R1),R0 ;SET THE FPS.
LDFPS R0
MOV #STXBF,R0
15: STXP ACO,(R0) ;TEST INSTRUCTION.
STFPS R4 ;GET FPS.
MOV R4,2*STMP7
MOV 16(R1),2*STMP10
MOV 2*STXBF,2*STMP4
CMP 10(R1),2*STXBF ;WAS RESULT CORRECT?
BEQ 55 ;BRANCH IF CORRECT.
CMP 12(R1),2*STXBF ;OTHERWISE SEE IF THE FAILURE WAS ANTICIPATED.
BNE 25
JMP 22(R1)

: IF NOT ANTICIPATED REPORT ERROR HERE.
25:
35: ERROR 347 ;EXP BAD
45: JMP 30(R1)

55: CMP R4,16(R1) ;SEE IF THE FPS IS CORRECT.
BEQ 105 ;BRANCH IF CORRECT.
CMP R4,20(R1) ;SEE IF THE FAILURE WAS ANTICIPATED.
BNE 65
JMP 26(R1)

: FPS ERROR WAS NOT ANTICIPATED SO REPORT ERROR HERE.
65:
75: ERROR 350 ;FPS BAD
BR 45

: SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
105: CMP #76543,2*STXBF+2
BEQ 45
115: ERROR 351 ;FDPL+0 ST 347X
BR 45

```

```

036736 000004
036740 104413
036742 012700 040000
036746 170100
036750 170003
036752 012700 037126
036756 012710 177777
036762 012760 177777 000002
036770 012737 036776 001236
036776 170310
037000 170204
037002 012700 037126
037006 011037 001240
037012 016037 000002 001242
037020 012737 000002 001244
037026 012737 036750 001246
037034 010437 001250
037040 012737 140000 001252
037046 022710 000002
037052 001010
037054 022760 036750 000002
037062 001006
037064 022704 140000
037070 001013
037072 000422

```

STEXP: .WORD -1,-1,-1,-1,-1

YYCDONE: RSETJP

```

:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER (HAS
:THE USER TYPED CONTROL G?).

```

```

*****
*EST 75 STST TEST
*****

```

```

* THIS IS A TEST OF THE STST
* INSTRUCTION. FIRST AN ILLEGAL FPP OP CODE
* INSTRUCTION IS USED TO ENTER AN
* ERROR CONDITION IN THE FEC AND
* FEA. THE STST IS EXECUTED AND
* THE FEC AND FEA ARE CHECKED

```

ST75: SCOPE

```

ZC01: LPEER          :SET UP THE LOOP ON ERROR ADDRESS.
      MOV #40000,R0  :SET FPS. F:D=1.
      LD FPS R0
ZC02: .WORD 170003  :ILLEGAL FPP
                        :OP CODE
      MOV #ZC0BF,R0  :SET UP THE OUTPUT BUFFER.
      MOV #-1,(R0)
      MOV #-12,(R0)
      MOV #ZC03,2*STMP2
ZC03: STST (R0)      :GET FEC AND
                        :FEA
                        :GET FPS.
      STFPS R4
      MOV #ZC0BF,R0
      MOV (R0),2*STMP3
      MOV 2(R0),2*STMP4
      MOV #2,2*STMP5
      MOV #ZC02,2*STMP6
      MOV R4,2*STMP7
      MOV #140000,2*STMP10
ZC05: CMP #2,(R0)  :SEE IF FEC IS CORRECT.
      BNE ZC05     :BRANCH IF INCORRECT.
      CMP #ZC02,2(R0) :SEE IF FEA ADDRESS IS CORRECT.
      BNE ZC10     :BRANCH IF INCORRECT.
      CMP #140000,R4 :SEE IF FPS IS CORRECT.
      BNE ZC15     :BRANCH IF INCORRECT.
      BR ZCDONE

```

:REPORT FEC INCORRECT

D13

```

8789 037202 104401 037210 TYPE 658 :: TYPE ASCIZ STRING
8790 037206 000407 BR 648 :: GET OVER THE ASCIZ
8791 :: 658: .ASCIZ (12)(15) END PASS 0
8792 037226 648: MOV $PASS,-(SP) :: SAVE $PASS FOR TYPEOUT
8793 037226 016746 142072 :: TYPE PASS NUMBER IN OCTAL
8794 :: GO TYPE--OCTAL ASCII
8795 037232 104403 TYPOS :: TYPE 6 DIGITS
8796 037234 006 .BYTE 6 :: SUPPRESS LEADING ZEROS
8797 037235 000 .BYTE 0 :: TYPE ASCIZ STRING
8798 037236 104401 037244 TYPE 678 :: GET OVER THE ASCIZ
8799 037242 000421 BR 668 :: TOTAL ERRORS SINCE LAST REPORT /
8800 :: 678: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
8801 668: MOV $ERTTL,-(SP) :: SAVE $ERTTL FOR TYPEOUT
8802 037306 016746 141600 :: TOTAL NUMBER OF ERRORS IN OCTAL
8803 037312 104403 TYPOS :: GO TYPE--OCTAL ASCII
8804 037314 006 .BYTE 6 :: TYPE 6 DIGITS
8805 037315 000 .BYTE 0 :: SUPPRESS LEADING ZEROS
8806 037316 104401 001313 TYPE $SCLF :: TYPE CARRIAGE RETURN, LINE FEED
8807 037322 005067 141564 CLR $ERTTL :: CLEAR ERROR TOTAL
8808 037326 013700 000042 $GET42: MOV $R42,R0 :: GET MONITOR ADDRESS
8809 037332 001414 BEQ $DOAGN :: BRANCH IF NO MONITOR
8810 037334 005046 CLR -(SP) :: INSURE THE "T" BIT IS CLEAR
8811 037336 012746 037344 MOV $SCLR.T,-(SP) :: SETUP FOR AN RTI OR RTT
8812 037342 000426 BR $RTN :: GO DO AN RTI OR RTT TO LOAD THE PSW
8813 :: WITH A CLEARED "T" BIT
8814 $SCLR.T: MOV $R42,R0 :: INSURE R0 CONTAINS THE MONITORS
8815 037344 013700 000042 BEQ $DOAGN :: RETURN ADDRESS
8816 037350 001405 RESET :: CLEAR THE WORLD
8817 037352 000005 $ENDAD: JSR PC,(R0) :: GO TO MONITOR
8818 037354 004710 NOP :: SAVE ROOM
8819 037356 000240 NOP :: FOR
8820 037360 000240 NOP :: ACT11
8821 037362 000240 $DOAGN: TRAP :: PUSH OLD PSW AND PC ON STACK
8822 037364 104400 BIC #20,(SP) :: CLEAR THE "T" BIT
8823 037366 042716 000020 BIT #BIT12,$SWR :: RUN WITH TRACE TRAP?
8824 037372 032777 010000 141540 BNE IS :: BR IF NO
8825 037400 001005 COM $TBIT :: IS IT TIME FOR TRACE TRAP
8826 037402 005167 000020 BMI IS :: BR IF NO
8827 037406 100402 BIS #20,(SP) :: SET TRACE TRAP
8828 037410 052716 000020 IS: MOV $SLOOP,-(SP) :: JUMP TO START OF TEST
8829 037414 012746 037422 $PTRN: RTI :: RETURN--THIS IS CHANGED TO
8830 037420 000002 :: AN "RTT" IF "RTT" IS A LEGAL
8831 037422 INSTRUCTION .
8832 037422 000137 $LOOP: JMP @PC+ :: RETURN
8833 037422 006576 $RTNAD: .WORD LOOP
8834 037424 000000 $TBIT: .WORD 0 :: "T" BIT STATE INDICATOR
8835 037426 000000 $ENULL: .BYTE -1,-1,0 :: NULL CHARACTER STRING
8836 037430 377 .EVEN
8837 037434 .SBTTL SCOPE HANDLER ROUTINE
8838
8839
8840
8841
8842
8843
8844

```

::*****

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

0327424
0327425
0327426
0327427
0327428
0327429
0327430
0327431
0327432
0327433
0327434
0327435
0327436
0327437
0327438
0327439
0327440
0327441
0327442
0327443
0327444
0327445
0327446
0327447
0327448
0327449
0327450
0327451
0327452
0327453
0327454
0327455
0327456
0327457
0327458
0327459
0327460
0327461
0327462
0327463
0327464
0327465
0327466
0327467
0327468
0327469
0327470
0327471
0327472
0327473
0327474
0327475
0327476
0327477
0327478
0327479
0327480
0327481
0327482
0327483
0327484
0327485
0327486
0327487
0327488
0327489
0327490
0327491
0327492
0327493
0327494
0327495
0327496
0327497
0327498
0327499
0327500
0327501
0327502
0327503
0327504
0327505
0327506
0327507
0327508
0327509
0327510
0327511
0327512
0327513
0327514
0327515
0327516
0327517
0327518
0327519
0327520
0327521
0327522
0327523
0327524
0327525
0327526
0327527
0327528
0327529
0327530
0327531
0327532
0327533
0327534
0327535
0327536
0327537
0327538
0327539
0327540
0327541
0327542
0327543
0327544
0327545
0327546
0327547
0327548
0327549
0327550
0327551
0327552
0327553
0327554
0327555
0327556
0327557
0327558
0327559
0327560
0327561
0327562
0327563
0327564
0327565
0327566
0327567
0327568
0327569
0327570
0327571
0327572
0327573
0327574
0327575
0327576
0327577
0327578
0327579
0327580
0327581
0327582
0327583
0327584
0327585
0327586
0327587
0327588
0327589
0327590
0327591
0327592
0327593
0327594
0327595
0327596
0327597
0327598
0327599
0327600
0327601
0327602
0327603
0327604
0327605
0327606
0327607
0327608
0327609
0327610
0327611
0327612
0327613
0327614
0327615
0327616
0327617
0327618
0327619
0327620
0327621
0327622
0327623
0327624
0327625
0327626
0327627
0327628
0327629
0327630
0327631
0327632
0327633
0327634
0327635
0327636
0327637
0327638
0327639
0327640
0327641
0327642
0327643
0327644
0327645
0327646
0327647
0327648
0327649
0327650
0327651
0327652
0327653
0327654
0327655
0327656
0327657
0327658
0327659
0327660
0327661
0327662
0327663
0327664

```

::THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
::AND LOAD THE TEST NUMBER(SYSTNM) INTO THE DISPLAY REG.(DISPLA) 7:C
::AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY(15:03)
::THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
::SW14=1      LOOP ON TEST
::SW11=1      INHIBIT ITERATIONS
::SW09=1      LOOP ON ERROR
::SW08=1      LOOP ON TEST IN SWR(7:0)
::CALL
::* SCOPE          ;;SCOPE=IC1

$SCOPE:
1S:  CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
    BIT  #BIT14,2SWR  ;;LOOP ON PRESENT TEST?
    BNE  $OVER       ;;YES IF SW14=1
    ;;*****START OF CODE FOR THE XOR TESTER*****
    $XTSTR: BR      6S
    MOV  2$ERRVEC, -(SP)  ;;IF RUNNING ON THE "XOR" TESTER CHANGE
    MOV  #55,2$ERRVEC    ;;THIS INSTRUCTION TO A "NOP" (NOP=243)
    TST  2$177060        ;;SAVE THE CONTENTS OF THE ERROR VECTOR
    MOV  (SP)+, 2$ERRVEC  ;;SET FOR TIMEOUT
    BR   $SVLAD          ;;TIME OUT ON XOR?
    CMP  (SP)+, (SP)+    ;;RESTORE THE ERROR VECTOR
    MOV  (SP)+, 2$ERRVEC  ;;GO TO THE NEXT TEST
    BR   7S              ;;CLEAR THE STACK AFTER A TIME OUT
    BR   7S              ;;RESTORE THE ERROR VECTOR
    BR   7S              ;;LOOP ON THE PRESENT TEST
6S:  ;;*****END OF CODE FOR THE XOR TESTER*****
    BIT  #BIT08,2SWR    ;;LOOP ON SPEC. TEST?
    BEQ  2S             ;;BR IF NO
    CMPB 2SWR, $STNM    ;;ON THE RIGHT TEST? SWR(7:0)
    BEQ  $OVER          ;;BR IF YES
    TSTB $SERFLG        ;;HAS AN ERROR OCCURRED?
    BEQ  3S             ;;BR IF NO
    CMPB $SERMAX, $SERFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
    BHI  3S             ;;BR IF NO
    BIT  #BIT09,2SWR    ;;LOOP ON ERROR?
    BEQ  4S             ;;BR IF NO
    MOV  $LPERR, $LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
    BR   $OVER
4S:  CLRB $SERFLG      ;;ZERO THE ERROR FLAG
    CLR  $TIMES         ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
    BR   1S            ;;ESCAPE TO THE NEXT TEST
3S:  BIT  #BIT11,2SWR  ;;INHIBIT ITERATIONS?
    BNE  1S            ;;BR IF YES
    TST  $PASS         ;;IF FIRST PASS OF PROGRAM
    BEQ  1S            ;;INHIBIT ITERATIONS
    INC  $ICNT         ;;INCREMENT ITERATION COUNT
    CMP  $TIMES, $ICNT  ;;CHECK THE NUMBER OF ITERATIONS MADE
    BGE  $OVER         ;;BR IF MORE ITERATION REQUIRED
    MOV  #1, $ICNT     ;;REINITIALIZE THE ITERATION COUNTER
    MOV  $MXCNT, $TIMES  ;;SET NUMBER OF ITERATIONS TO DO
    $SVLAD: INCB $STNM  ;;COUNT TEST NUMBERS
    MOV  $STNM, $STNM   ;;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

```

```

8901 037670 112767 000001 141217      MOV      B1,SEMAX          ;; ONLY ALLOW ONE(1) ERROR ON 'NEXT TEST'
8902 037676 016777 141200 141236  SOVER:  MOV      $*STNM,DISPLAY  ;; DISPLAY TEST NUMBER
8903 037704 016716 141176      MOV      SLPADR,(SP)      ;; FUDGE RETURN ADDRESS
8904 037710 000002      RTI                          ;; FIXES FS
8905 037712 000001      SMXCNT: 1                  ;; MAX. NUMBER OF ITERATIONS

```

.SBTTL ERROR HANDLER ROUTINE

```

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
*AND GO TO ERTYPE ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1      HALT ON ERROR
*SW13=1      INHIBIT ERROR TYPEOUTS
*SW10=1      BELL ON ERROR
*SW09=1      LOOP ON ERROR
*CALL
*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER

```

```

8921 037714      SERROR:
8922 037714 104406      CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
8923 037716 105267 141161 7$:      INCB      SERFLG          ;; SET THE ERROR FLAG
8924 037722 001775      BEQ      7$          ;; DON'T LET THE FLAG GO TO ZERO
8925 037724 016777 141152 141210  MOV      $*STNM,DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
8926 037732 032777 002000 141200  BIT      #BIT10,SWR      ;; BELL ON ERROR?
8927 037740 001402      BEQ      1$          ;; NO - SKIP
8928 037742 104401 001306      TYPE      $BELL          ;; RING BELL
8929 037746 005267 141140 1$:      INC      $ERTTL          ;; COUNT THE NUMBER OF ERRORS
8930 037752 011667 141140      MOV      (SP),SERRPC      ;; GET ADDRESS OF ERROR INSTRUCTION
8931 037756 162767 000002 141132  SUB      #2,SERRPC
8932 037764 117767 141126 141122  MOV      $SERRPC,$ITEMB  ;; STRIP AND SAVE THE ERROR ITEM CODE
8933 037772 032777 020000 141140  BIT      #BIT13,SWR      ;; SKIP TYPEOUT IF SET
8934 040000 001004      BNE      20$          ;; SKIP TYPEOUTS
8935 040002 004767 002124      JSR      PC,ERTYPE      ;; GO TO USER ERROR ROUTINE
8936 040006 104401 001313      TYPE      ,SCLF
8937 040012      20$:
8938 040012 122767 000001 141316  CMPB     #APTENV,$ENV      ;; RUNNING IN APT MODE
8939 040020 001007      BNE      2$          ;; NO SKIP APT ERROR REPORT
8940 040022 116767 141066 000004  MOV      $ITEMB,21$      ;; SET ITEM NUMBER AS ERROR NUMBER
8941 040030 004767 000740      JSR      PC,$ATY4      ;; REPORT FATAL ERROR TO APT
8942 040034      000      21$:      .BYTE      0
8943 040035      000      .BYTE      0
8944 040036 000777      22$:      BR      22$          ;; APT ERROR LOOP
8945 040040 005777 141074 2$:      TST      $SWR          ;; HALT ON ERROR
8946 040044 100002      BPL          ;; SKIP IF CONTINUE
8947 040046 000000      HALT          ;; HALT ON ERROR!
8948 040050 104406      CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
8949 040052 032777 001000 141060 3$:      BIT      #BIT09,SWR      ;; LOOP ON ERROR SWITCH SET?
8950 040060 001402      BEQ      4$          ;; BR IF NO
8951 040062 016716 141022      MOV      $LPERR,(SP)      ;; FUDGE RETURN FOR LOOPING
8952 040066 005767 141212 4$:      TST      $ESCAPE          ;; CHECK FOR AN ESCAPE ADDRESS
8953 040072 001402      BEQ      5$          ;; BR IF NONE
8954 040074 016716 141204      MOV      $ESCAPE,(SP)      ;; FUDGE RETURN ADDRESS FOR ESCAPE
8955 040100      5$:
8956 040100 022737 037354 000042  CMP      #SENDAD,$#42      ;; ACT-11 AUTO-ACCEPT?

```

```

001001
000000
001110
001112
032777 001000 141020
001013
011637 001162
062737 177776 001162
122777 000377 141020
001002
062716 000002
000002
010046
010146
010246
010346
010446
010546
016646 000022
016646 000022
016646 000022
016646 000022
000002
012666 000022
012666 000022
012666 000022
012666 000022
012605
012604
012603
012602
012601

```

```

BNE 68 ::BRANCH IF NO
HALT ::YES
68:
BIT #BIT09,2SWP
BNE ERM10
MOV (SP),2#SREG0 ;SEE IF ERROR #377
ADD #2,2#SREG0
CMPB #377,2#SREG0
BNE ERM10
ADD #2,(SP)
ERM10: RTI

```

.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

```

*****
*SAVE RO-R5
*CALL:
* SAVREG
*UPON RETURN FROM $$SAVREG THE STACK WILL LOOK LIKE:
*
*TOP---(+16)
* +2---(+18)
* +4---R5
* +6---R4
* +8---R3
*+10---R2
*+12---R1
*+14---R0

```

```

$$SAVREG:
MOV R0,-(SP) ::PUSH R0 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 R2,SP),-(SP) ::SAVE PS OF MAIN FLOW
MOV R2,SP),-(SP) ::SAVE PC OF MAIN FLOW
MOV R3,SP),-(SP) ::SAVE PS OF CALL
MOV R2,SP),-(SP) ::SAVE PC OF CALL
RTI

```

```

*RESTORE RO-R5
*CALL:
* RESREG
$$RESREG:
MOV (SP)+,R2(SP) ::RESTORE PC OF CALL
MOV (SP)+,R2(SP) ::RESTORE PS OF CALL
MOV (SP)+,R2(SP) ::RESTORE PC OF MAIN FLOW
MOV (SP)+,R2(SP) ::RESTORE PS OF MAIN FLOW
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

```

040242 012600
040244 000002

MOV (SP)+,RO ;:POP STACK INTO RO
RTI

.SBTTL TYPE ROUTINE

*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.

*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE .MESADR ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*

040246 105767 140705
040252 100002
040254 000000
040256 000430
040260 010046
040262 017600 000002
040266 122767 000001 141042
040274 001011
040276 132767 000100 141033
040304 001405
040306 010067 000004
040312 004767 000446
040316 000000
040320 132767 000040 141011
040326 001003
040330 112046
040332 001005
040334 005726
040336 012600
040340 062716 000002
040344 000002
040346 122716 000011
040352 001430
040354 122716 000200
040360 001006
040362 005726
040364 104401
040366 001313
040370 105067 000130
040374 000755
040376 004767 000056
040402 126726 140550
040406 001350
040410 016746 140540
040414 105366 000001

\$TYPE: TSTB \$TPFLG ;: IS THERE A TERMINAL?
BPL 1\$;: BR IF YES
HALT ;: HALT HERE IF NO TERMINAL
BR 3\$;: LEAVE
1\$: MOV RO, -(SP) ;: SAVE RO
MOV #2(SP), RO ;: GET ADDRESS OF ASCIZ STRING
CMPB #APTENV, \$ENV ;: RUNNING IN APT MODE
BNE 62\$;: NO GO CHECK FOR APT CONSOLE
BITB #APTSPool, \$ENVM ;: SPOOL MESSAGE TO APT
BEQ 62\$;: NO GO CHECK FOR CONSOLE
MOV RO, 61\$;: SETUP MESSAGE ADDRESS FOR APT
JSR PC, \$ATY3 ;: SPOOL MESSAGE TO APT
0 ;: MESSAGE ADDRESS
61\$: BITB #APTC SUP, \$ENVM ;: APT CONSOLE SUPPRESSED
62\$: BNE 60\$;: YES, SKIP TYPE OUT
2\$: MOVB (RO)+, -(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)+, RO ;: RESTORE RO
3\$: ADD #2, (SP) ;: ADJUST RETURN PC
RTI ;: RETURN
4\$: CMPB #HT, (SP) ;: BRANCH IF <HT>
BEQ 8\$;: BRANCH IF NOT <CRLF>
5\$: CMPB #CRLF, (SP) ;: BRANCH IF NOT <CRLF>
BNE 5\$;: POP <CR><LF> EQUIV
TST (SP)+ ;: TYPE A CR AND LF
TYPE ;: TYPE A CR AND LF
\$CRLF ;: CLEAR CHARACTER COUNT
CLRB \$CHARCNT ;: CLEAR CHARACTER COUNT
BR 2\$;: GET NEXT CHARACTER
5\$: JSR PC, \$TYPEC ;: GO TYPE THIS CHARACTER
6\$: CMPB \$FILLC, (SP)+ ;: IS IT TIME FOR FILLER CHARS.?
BNE 2\$;: IF NO GO GET NEXT CHAR.
MOV \$NULL, -(SP) ;: GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
7\$: DECB 1(SP) ;: DOES A NULL NEED TO BE TYPED?


```

9069 040420 002770          BLT      65          ;; BR IF NO--GO POP THE NULL OFF OF STACK
9070 040422 004767 000032    JSR      PC,$TYPEC  ;; SO TYPE A NULL
9071 040426 105367 000072    DECB   $C+ARCNT    ;; DO NOT COUNT AS A COUNT
9072 040432 000770          BR       75          ;; LOOP
9073
9074          ;HORIZONTAL TAB PROCESSOR
9075
9076 040434 112716 000040    85:     MOVB     #' (SP)          ;; REPLACE TAB WITH SPACE
9077 040440 004767 000014    95:     JSR      PC,$TYPEC  ;; TYPE A SPACE
9078 040444 132767 000007 000052    BITB   #',$CHARCNT  ;; BRANCH IF NOT AT
9079 040452 001372          BNE     95          ;; TAB STOP
9080 040454 005726          TST    (SP)+        ;; POP SPACE OFF STACK
9081 040456 000724          BR      25          ;; GET NEXT CHARACTER
9082 040460 105777 140464    $TYPEC: TSTB   2$TPS    ;; WAIT UNTIL PRINTER IS READY
9083 040464 100375          BPL    $TYPEC
9084 040466 116677 000002 140456    MOVB   2(SP),2$TPB   ;; LOAD CHAR TO BE TYPED INTO DATA REG.
9085 040474 122766 000015 000002    CMPB   #CR,2(SP)    ;; IS CHARACTER A CARRIAGE RETURN?
9086 040502 001003          BNE    15          ;; BRANCH IF NO
9087 040504 105367 000014    CLAB   $CHARCNT     ;; YES--CLEAR CHARACTER COUNT
9088 040510 000406          BR     $TYPEX
9089 040512 122766 000012 000002    15:    CMPB   #LF,2(SP)   ;; IS CHARACTER A LINE FEED?
9090 040520 001402          BEQ   $TYPEX        ;; BRANCH IF YES
9091 040522 105227          INCB  (PC)+         ;; COUNT THE CHARACTER
9092 040524 000000    $CHARCNT: .WORD 0   ;; CHARACTER COUNT STORAGE
9093 040526 000207    $TYPEX: RTS      PC
9094
9095
9096          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
9097
9098          ;*****
9099          ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
9100          ;OCTAL (ASCII) NUMBER AND TYPE IT.
9101          ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
9102          ;CALL:
9103          ;     MOV     NUM,-(SP)          ;; NUMBER TO BE TYPED
9104          ;     TYPOS          ;; CALL FOR TYPEOUT
9105          ;     .BYTE  N          ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
9106          ;     .BYTE  M          ;; M=1 OR 0
9107          ;                                     ;; 1=TYPE LEADING ZEROS
9108          ;                                     ;; 0=SUPPRESS LEADING ZEROS
9109          ;
9110          ;$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
9111          ;$TYPOS OR $TYPOC
9112          ;CALL:
9113          ;     MOV     NUM,-(SP)          ;; NUMBER TO BE TYPED
9114          ;     TYPON          ;; CALL FOR TYPEOUT
9115          ;
9116          ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
9117          ;CALL:
9118          ;     MOV     NUM,-(SP)          ;; NUMBER TO BE TYPED
9119          ;     TYPOC          ;; CALL FOR TYPEOUT
9120          ;
9121 040530 017646 000000    $TYPOS: MOV     2(SP),-(SP)          ;; PICKUP THE MODE
9122 040534 116667 000001 000211    MOVB   1(SP),$OFILL  ;; LOAD ZERO FILL SWITCH
9123 040542 112667 000207    MOVB   (SP)+,$OMODE+1 ;; NUMBER OF DIGITS TO TYPE
9124 040546 062716 000002    ADD    #2,(SP)      ;; ADJUST RETURN ADDRESS

```

J13

```

9125 040552 000406 BR $TYPON
9126 040554 112767 000001 000171 $TYP0C: MOVB #1,$0FILL ;;SET THE ZERO FILL SWITCH
9127 040562 112767 000006 000165 MOVB #6,$0MODE+1 ;;SET FOR SIX(6) DIGITS
9128 040570 112767 000005 000154 $TYPON: MOVB #5,$0CNT ;;SET THE ITERATION COUNT
9129 040576 010346 MOV R3,-(SP) ;;SAVE R3
9130 040600 010446 MOV R4,-(SP) ;;SAVE R4
9131 040602 010546 MOV R5,-(SP) ;;SAVE R5
9132 040604 116704 000145 MOVB $0MODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
9133 040610 005404 NEG R4
9134 040612 062704 000006 ADD #6,R4 ;;SUBTRACT IT FOR MAX. ALLOWED
9135 040616 110467 000132 MOVB R4,$0MODE ;;SAVE IT FOR USE
9136 040622 116704 000125 MOVB $0FILL,R4 ;;GET THE ZERO FILL SWITCH
9137 040626 016605 000012 MOV 12(SP),R5 ;;PICKUP THE INPUT NUMBER
9138 040632 005003 CLR R3 ;;CLEAR THE OUTPUT WORD
9139 040634 006105 1$: ROL R5 ;;ROTATE MSB INTO "C"
9140 040636 000404 BR 3$ ;;GO DO MSB
9141 040640 006105 2$: ROL R5 ;;FORM THIS DIGIT
9142 040642 006105 ROL R5
9143 040644 006105 ROL R5
9144 040646 010503 MOV R5,R3
9145 040650 006103 3$: ROL R3 ;;GET LSB OF THIS DIGIT
9146 040652 105367 000076 DECB $0MODE ;;TYPE THIS DIGIT?
9147 040656 100016 BPL 7$ ;;BR IF NO
9148 040660 042703 177770 BIC #177770,R3 ;;GET RID OF JUNK
9149 040664 001002 BNE 4$ ;;TEST FOR 0
9150 040666 005704 TST R4 ;;SUPPRESS THIS 0?
9151 040670 001403 BEQ 5$ ;;BR IF YES
9152 040672 005204 4$: INC R4 ;;DON'T SUPPRESS ANYMORE 0'S
9153 040674 052703 000060 BIS #'0,R3 ;;MAKE THIS DIGIT ASCII
9154 040700 052703 000040 5$: BIS #' ,R3 ;;MAKE ASCII IF NOT ALREADY
9155 040704 110367 000040 MOVB R3,$5 ;;SAVE FOR TYPING
9156 040710 104401 040750 TYPE R5 ;;GO TYPE THIS DIGIT
9157 040714 105367 000032 7$: DECB $0CNT ;;COUNT BY 1
9158 040720 003347 BGT 2$ ;;BR IF MORE TO DO
9159 040722 002402 BLT 6$ ;;BR IF DONE
9160 040724 005204 INC R4 ;;INSURE LAST DIGIT ISN'T A BLANK
9161 040726 000744 BR 2$ ;;GO DO THE LAST DIGIT
9162 040730 012605 6$: MOV (SP)+,R5 ;;RESTORE R5
9163 040732 012604 MOV (SP)+,R4 ;;RESTORE R4
9164 040734 012603 MOV (SP)+,R3 ;;RESTORE R3
9165 040736 016666 000002 000004 MOV 2(SP),4(SP) ;;SET THE STACK FOR RETURNING
9166 040744 012616 MOV (SP)+,(SP)
9167 040746 000002 RTI ;;RETURN
9168 040750 000 8$: .BYTE 0 ;;STORAGE FOR ASCII DIGIT
9169 040751 000 .BYTE 0 ;;TERMINATOR FOR TYPE ROUTINE
9170 040752 000 $0CNT: .BYTE 0 ;;OCTAL DIGIT COUNTER
9171 040753 000 $0FILL: .BYTE 0 ;;ZERO FILL SWITCH
9172 040754 000000 $0MODE: .WORD 0 ;;NUMBER OF DIGITS TO TYPE
9173
9174 .SBTTL APT COMMUNICATIONS ROUTINE
9175
9176 ;*****
9177 040756 112767 000001 000236 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
9178 040764 112767 000001 000226 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
9179 040772 000403 BR $ATYC
9180 040774 112767 000001 000220 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
  
```

```

9181 041002          SATYC:
9182 041002 010046      MOV      RO,-(SP)      ;; PUSH RO ON STACK
9183 041004 010146      MOV      RI,-(SP)      ;; PUSH RI ON STACK
9184 041006 105767 000206  TSTB     $MFLG        ;; SHOULD TYPE A MESSAGE?
9185 041012 001450      BEQ      5$           ;; IF NOT: BR
9186 041014 122767 000001 140314  CMPB     $APTENV,$ENV  ;; OPERATING UNDER APT?
9187 041022 001031      BNE      3$           ;; IF NOT: BR
9188 041024 132767 000100 140305  BITB     $APTPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
9189 041032 001425      BEQ      3$           ;; IF NOT: BR
9190 041034 017600 000004      MOV      @4(SP),RO    ;; GET MESSAGE ADDR.
9191 041040 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDR.
9192 041046 005767 140244 1$:      TST      $MSGTYPE    ;; SEE IF DONE W/ LAST XMISSION?
9193 041052 001375      BNE      1$           ;; IF NOT: WAIT
9194 041054 010067 140252  MOV      RO,$MSGAD    ;; PUT ADDR IN MAILBOX
9195 041060 105720      TSTB     (RO)+        ;; FIND END OF MESSAGE
9196 041062 001376      BNE      2$           ;;
9197 041064 166700 140242  SUB      $MSGAD,RO    ;; SUB START OF MESSAGE
9198 041070 006200      ASR      RO           ;; GET MESSAGE LNTH IN WORDS
9199 041072 010067 140236  MOV      RO,$MSGGLT   ;; PUT LENGTH IN MAILBOX
9200 041076 012767 000004 140212  MOV      #4,$MSGTYPE  ;; TELL APT TO TAKE MSG.
9201 041104 000413      BR       5$           ;;
9202 041106 017667 000004 000016 3$:      MOV      @4(SP),4$   ;; PUT MSG ADDR IN JSR LINKAGE
9203 041114 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDRESS
9204 041122 016746 136650      MOV      177776,-(SP) ;; PUSH 177776 ON STACK
9205 041126 004767 177114  JSR      PC,$TYPE    ;; CALL TYPE MACRO
9206 041132 000000      .WORD   0
9207 041134      4$:
9208 041134 105767 000062 10$:      TSTB     $FFLG        ;; SHOULD REPORT FATAL ERRCR?
9209 041140 001416      BEQ      12$         ;; IF NOT: BR
9210 041142 005767 140170  TST      $ENV         ;; RUNNING UNDER APT?
9211 041146 001413      BEQ      12$         ;; IF NOT: BR
9212 041150 005767 140142 11$:      TST      $MSGTYPE    ;; FINISHED LAST MESSAGE?
9213 041154 001375      BNE      11$         ;; IF NOT: WAIT
9214 041156 017667 000004 140134  MOV      @4(SP),$FATAL ;; GET ERROR #
9215 041164 062766 000002 000004  ADD      #2,4(SP)     ;; BUMP RETURN ADDR.
9216 041172 005267 140120  INC      $MSGTYPE    ;; TELL APT TO TAKE ERROR
9217 041176 105067 000020 12$:      CLRB     $FFLG        ;; CLEAR FATAL FLAG
9218 041202 105067 000013  CLRB     $LFLG        ;; CLEAR LOG FLAG
9219 041206 105067 000006  CLRB     $MFLG        ;; CLEAR MESSAGE FLAG
9220 041212 012601      MOV      (SP)+,R1    ;; POP STACK INTO R1
9221 041214 012600      MOV      (SP)+,RO    ;; POP STACK INTO RO
9222 041216 000207      RTS      PC         ;; RETURN
9223 041220          000      $MFLG: .BYTE 0    ;; MESSG. FLAG
9224 041221          000      $LFLG: .BYTE 0    ;; LOG FLAG
9225 041222          000      $FFLG: .BYTE 0    ;; FATAL FLAG
9226          041224      .EVEN
9227          000200  APTSIZE=200
9228          000001  APTENV=001
9229          000100  APTPOOL=100
9230          000040  APTCSUP=040
9231
9232          .SBTTL  TTY INPUT ROUTINE
9233
9234          ;*****
9235          .ENABL  LSB
9236

```

```

*****
:SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
:ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
:SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
:WHEN OPERATING IN TTY FLAG MODE.
9237 041224 022767 000176 137706 $CKSWR: CMP $SWREG, SWR ;; IS THE SOFT-SWR SELECTED?
9238 041232 001074 BNE 15$ ;; BRANCH IF NO
9239 041234 105777 137704 TSTB 2$TKS ;; CHAR THERE?
9240 041240 100071 BPL 15$ ;; IF NO, DON'T WAIT AROUND
9241 041242 117746 137700 MOVB 2$TKB, -(SP) ;; SAVE THE CHAR
9242 041246 042716 177600 BIC #1C177, (SP) ;; STRIP-OFF THE ASCII
9243 041252 022726 000007 CMP #7, (SP)+ ;; IS IT A CONTROL G?
9244 041256 001062 BNE 15$ ;; NO, RETURN TO USER
9245 041260 126727 137650 000001 CMPB $AJTOB, #1 ;; ARE WE RUNNING IN AUTO-MODE?
9246 041266 001456 BEQ 15$ ;; BRANCH IF YES
9247 041270 104401 041633 $GTSWR: TYPE , $CNTLG ;; ECHO THE CONTROL-G (+G)
9248 041274 104401 041640 TYPE $MSWR ;; TYPE CURRENT CONTENTS
9249 041300 016746 136672 MOV $WREG, -(SP) ;; SAVE SWREG FOR TYPEOUT
9250 041304 104402 TYPOC ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9251 041306 104401 041651 TYPE , $MNEW ;; PROMPT FOR NEW SWR
9252 041312 005046 19$: CLR -(SP) ;; CLEAR COUNTER
9253 041314 005046 CLR -(SP) ;; THE NEW SWR
9254 041316 105777 137622 7$: TSTB 2$TKS ;; CHAR THERE?
9255 041322 100375 BPL 7$ ;; IF NOT TRY AGAIN
9256 041324 117746 137616 MOVB 2$TKB, -(SP) ;; PICK UP CHAR
9257 041330 042716 177600 BIC #1C177, (SP) ;; MAKE IT 7-BIT ASCII
9258 041334 021627 000025 9$: CMP (SP), #25 ;; IS IT A CONTROL-U?
9259 041340 001005 BNE 10$ ;; BRANCH IF NOT
9260 041342 104401 041626 TYPE $CNTLU ;; YES, ECHO CONTROL-U (+U)
9261 041346 062706 000006 20$: ADD #6, SP ;; IGNORE PREVIOUS INPUT
9262 041352 000757 BR 19$ ;; LET'S TRY IT AGAIN
9263 041354 021627 000015 10$: CMP (SP), #15 ;; IS IT A <CR>?
9264 041360 001022 BNE 16$ ;; BRANCH IF NO
9265 041362 005766 000004 TST 4(SP) ;; YES, IS IT THE FIRST CHAR?
9266 041366 001403 BEQ 11$ ;; BRANCH IF YES
9267 041370 016677 000002 137542 MOV 2(SP), 2$SWR ;; SAVE NEW SWR
9268 041376 062706 000006 11$: ADD #6, SP ;; CLEAR UP STACK
9269 041402 104401 001313 14$: TYPE $CRLF ;; ECHO <CR> AND <LF>
9270 041406 126727 137523 000001 CMPB $INTAG, #1 ;; RE-ENABLE TTY KBD INTERRUPTS?
9271 041414 001003 BNE 15$ ;; BRANCH IF NOT
9272 041416 012777 000100 137520 MOV #100, 2$TKS ;; RE-ENABLE TTY KBD INTERRUPTS
9273 041424 000002 15$: RTI ;; RETURN
9274 041426 004767 177026 16$: JSR PC, $TYPEC ;; ECHO CHAR
9275 041432 021627 000060 CMP (SP), #60 ;; CHAR < 0?
9276 041436 002420 BLT 18$ ;; BRANCH IF YES
9277 041440 021627 000067 CMP (SP), #67 ;; CHAR > ??
9278 041444 003015 BGT 18$ ;; BRANCH IF YES
9279 041446 042726 000060 BIC #60, (SP)+ ;; STRIP-OFF ASCII
9280 041452 005766 000002 TST 2(SP) ;; IS THIS THE FIRST CHAR

```

```

9293 041456 001403 BEQ 17$ :: BRANCH IF YES
9294 041460 006316 ASL (SP) :: NO, SHIFT PRESENT
9295 041462 006316 ASL (SP) :: CHAR OVER TO MAKE
9296 041464 006316 ASL (SP) :: ROOM FOR NEW ONE.
9297 041466 005266 000002 17$: INC 2(SP) :: KEEP COUNT OF CHAR
9298 041472 056616 177776 BIS -2(SP), (SP) :: SET IN NEW CHAR
9299 041476 000707 BR 7$ :: GET THE NEXT ONE
9300 041500 104401 001312 18$: TYPE $QUES :: TYPE ?<CR><LF>
9301 041504 000720 BR 20$ :: SIMULATE CONTROL-U
9302 .DSABL LSB
9303
9304
9305 :: *****
9306 :: *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9307 :: *CALL:
9308 :: * .RDCHR :: INPUT A SINGLE CHARACTER FROM THE TTY
9309 :: * RETURN HERE :: CHARACTER IS ON THE STACK
9310 :: * :: WITH PARITY BIT STRIPPED OFF
9311 ::
9312
9313 $RDCHR: MOV (SP), -(SP) :: PUSH DOWN THE PC
9314 041510 016666 000004 000002 MOV 4(SP), 2(SP) :: SAVE THE PS
9315 041516 105777 137422 15: TSTB 2$TKS :: WAIT FOR
9316 041522 100375 BPL 1$ :: A CHARACTER
9317 041524 117766 137416 000004 MOVB 2$TKB, 4(SP) :: READ THE TTY
9318 041532 042766 177600 000004 BIC #1C(177), 4(SP) :: GET RID OF JUNK IF ANY
9319 041540 026627 000004 000023 CMP 4(SP), #23 :: IS IT A CONTROL-S?
9320 041546 001013 BNE 3$ :: BRANCH IF NO
9321 041550 105777 137370 25: TSTB 2$TKS :: WAIT FOR A CHARACTER
9322 041554 100375 BPL 2$ :: LOOP UNTIL ITS THERE
9323 041556 117746 137364 MOVB 2$TKB, -(SP) :: GET CHARACTER
9324 041562 042716 177600 BIC #1C177, (SP) :: MAKE IT 7-BIT ASCII
9325 041566 022627 000021 CMP (SP)+, #21 :: IS IT A CONTROL-Q?
9326 041572 001366 BNE 2$ :: IF NOT DISCARD IT
9327 041574 000750 BR 1$ :: YES, RESUME
9328 041576 026627 000004 000140 35: CMP 4(SP), #140 :: IS IT UPPER CASE?
9329 041604 002407 BLT 4$ :: BRANCH IF YES
9330 041606 026627 000004 000175 CMP 4(SP), #175 :: IS IT A SPECIAL CHAR?
9331 041614 003003 BGT 4$ :: BRANCH IF YES
9332 041616 042766 000040 000004 BIC #40, 4(SP) :: MAKE IT UPPER CASE
9333 041624 000002 RTI :: GO BACK TO JSER
9334 041626 052536 005015 000 $CNTLU: .ASCIZ /↑U/<15><12> :: CONTROL "U"
9335 041633 136 006507 000012 $CNTLG: .ASCIZ /↑G/<15><12> :: CONTROL "G"
9336 041640 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
9337 041646 020075 000
9338 041651 040 047040 053505 $MNEW: .ASCIZ / NEW = /
9339 041656 036440 000040

```

.SBTTL TRAP DECODER

```

9340
9341
9342 :: *****
9343 :: *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
9344 :: *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
9345 :: *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
9346 :: *GO TO THAT ROUTINE.
9347
9348

```

9349 041662 010046
9350 041664 016600 000002
9351 041670 005740
9352 041672 111000
9353 041674 006300
9354 041676 016000 041716
9355 041702 000200
9356
9357
9358
9359

\$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 \$TRAPD(R0),R0 ;;INDEX TO TABLE
RTS R0 ;;GO TO ROUTINE

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

9360 041704 011646
9361 041706 016666 000004 000002
9362 041714 000002
9363
9364
9365

\$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.

9366
9367
9368
9369
9370
9371 041716 041704
9372 041720 040246
9373 041722 040554
9374 041724 040530
9375 041726 040570
9376
9377 041730 041274
9378
9379 041732 041224
9380 041734 041506
9381 041736 040152
9382 041740 040210
9383 041742 042666
9384 041744 042560
9385 000030
9386
9387
9388
9389
9390

; ROUTINE
;-----
\$TRAPD: .WORD \$TRAP2
\$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
\$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
\$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
\$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

\$GTSWR ;;CALL=GTSWR TRAP+5(104405) GET SOFT-SWR SETTING

\$CKSWR ;;CALL=CKSWR TRAP+6(104406) TEST FOR CHANGE IN SOFT-SWR
\$RDCHR ;;CALL=RDCHR TRAP+7(104407) TTY TYPEIN CHARACTER ROUTINE
\$SAVREG ;;CALL=SAVREG TRAP+10(104410) SAVE R0-R5 ROUTINE
\$RESREG ;;CALL=RESREG TRAP+11(104411) RESTORE R0-R5 ROUTINE
.RSET ;;CALL=RSETUP TRAP+12(104412) ROUTINE TO INITIALIZE AT END OF EACH TES
.LPER ;;CALL=LPER TRAP+13(104413) ROUTINE TO SET UP LOOP ON ERROR ADDRESS

\$TERM=-.\$TRAPD

.SBTTL POWER DOWN AND UP ROUTINES

9391 041746 012737 042124 000024
9392 041754 012737 000340 000026
9393 041762 010046
9394 041764 010146
9395 041766 010246
9396 041770 010346
9397 041772 010446
9398 041774 010546
9399 041776 017746 137136
9400 042002 010667 000122
9401 042006 012737 042020 000024
9402 042014 000000
9403 042016 000776
9404

: POWER DOWN ROUTINE
\$PWRDN: MOV \$SILLUP,2#\$PWRVEC ;;SET FOR FAST UP
MOV #340,2#\$PWRVEC+2 ;;PRIO:7
MOV R0,-(SP) ;;PUSH R0 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 2\$SWR,-(SP) ;;PUSH 2\$SWR ON STACK
MOV SP,\$SAVR6 ;;SAVE SP
MOV #2#\$PWRUP,2#\$PWRVEC ;;SET UP VECTOR
BR .-2 ;;HANG UP

9558	042528	012276			MOV	(R2)+, -(SP)		:FOUR OCTAL NUMBERS.
9559	042529	104402			TYPOC			
9560	042530	043002			TYPE			
9561	042531	012276			.WORD	SPACE		
9562	042532	104402			MOV	(R2)+, -(SP)		
9563	042533	043002			TYPOC			
9564	042534	104402			TYPE			
9565	042535	043002			.WORD	SPACE		
9566	042536	104402			MOV	(R2)+, -(SP)		
9567	042537	043002			TYPOC			
9568	042538	104402			TYPE			
9569	042539	043002			.WORD	SPACE		
9570	042540	011246			MOV	(R2), -(SP)		
9571	042541	104402			TYPOC			
9572	042542	00444			BR	ERT2		
9573	042543	122710	000004	105:	CMPB	#4, (R0)		:FORMAT FOUR?
9574	042544	001004			BNE	115		
9575	042545	013146			MOV	2(R1)+, -(SP)		:FORMAT FOUR SO TYPE
9576	042546	104402			TYPOS			:AN OCTAL NUMBER
9577	042547	016			.BYTE	16		:SUPPRESSING LEADING ZEROS.
9578	042548	000			.BYTE	0		
9579	042549	000435			BR	ERT2		
9580	042550	122710	000005	115:	CMPB	#5, (R0)		:FORMAT FIVE?
9581	042551	001005			BNE	135		
9582	042552	012137	042454		MOV	(R1)+, 2#125		:FORMAT FIVE SO TYPE AN
9583	042553	104402			TYPE			:ASCIZ STRING.
9584	042554	000000		125:	.WORD	0		
9585	042555	000427			BR	ERT3		
9586	042556	122710	000011	135:	CMPB	#11, (R0)		:FORMAT ELEVEN?
9587	042557	001005			BNE	155		
9588	042558	013137	042474		MOV	2(R1)+, 2#145		:FORMAT ELEVEN SO PICK
9589	042559	104402			TYPE			:A POINTER TO AN ASCIZ
9590	042560	000000		145:	.WORD	0		:STRING.
9591	042561	000417			BR	ERT3		
9592	042562	122710	000012	155:	CMPB	#12, (R0)		:FORMAT TWELVE?
9593	042563	001011			BNE	175		
9594	042564	013102			MOV	2(R1)+, R2		:FORMAT TWELVE SO TYPE
9595	042565	012703	000006		MOV	#6, R3		:TYPE SIX OCTAL NUMBERS
9596	042566	012246		165:	MOV	(R2)+, -(SP)		
9597	042567	104402			TYPOC			
9598	042568	104402			TYPE			
9599	042569	043002			.WORD	SPACE		
9600	042570	077305			SQB	R3, 165		
9601	042571	000401			BR	ERT2		
9602	042572	000000		175:	HALT			:UNDEFINED FORMAT FOR DATA????
9603	042573	104401		ERT2:	TYPE			:PRINT A TAB AFTER TYPING

E14

9800 042534 043005
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811
9812
9813 042614 000137 037142
9814
9815
9816
9817
9818
9819
9820
9821 042620 011637 001236
9822 042624 022626
9823 042626 104377
9824 042630 000442
9825 042632 104412
9826
9827
9828

```
.WORD STAB ;AN DATA TABLE ENTRY  
;OF ALL FORMATS EXCEPT  
;ASCIZ, FORMATS 5 OR 11  
ERT3: INC R0 ;POINT TO THE NEXT FORMAT  
TST (R0) ;END OF DATA TABLE.  
REQ ERT4  
BR ERT1  
ERT4: TYPE ;DONE.  
.WORD SCRLF ;RESTORE R1,R2 AND R3  
MOV (SP)+,R3  
MOV (SP)+,R2  
MOV (SP)+,R1  
ERT5: MOV (SP)+,R0 ;RESTORE RC.  
RTS PC ;AND RETURN.
```

.SBTTL FPP SPURIOUS TRAP TO 244 HANDLER

*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
*

```
FPPSPUR: MOV (SP),2*STMP2 ;SAVE PC OF TRAP.  
CMP (SP)+,(SP)+ ;RESTORE SP.  
STFPS R0 ;GET FPS  
MOV R0,2*STMP3  
STST R0 ;GET FEC  
MOV R0,2*STMP4  
IS: ERROR 377  
.WORD 441  
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
;THE USER TYPED CONTROL G?).  
JMP 2*SEOP
```

.SBTTL CPU SPURIOUS TRAP TO 4 HANDLER

*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
*

```
CPPSPUR: MOV (SP),2*STMP2 ;SAVE PC OF TRAP.  
CMP (SP)+,(SP)+  
IS: ERROR 377  
.WORD 442  
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
;SEE IF THE USER HAS EXPRESSED  
;THE DESIRE TO CHANGE THE SOFTWARE  
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
```

F14

9629
9630 042634 000137 037142 JMP 2#SEOP ;THE USER TYPED CONTROL G?).

.SBTTL CPU SPURIOUS TRAP TO 10 HANDLER

*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
*

9631
9632
9633
9634
9635
9636
9637
9638 042640 011637 001236 CPTWO: MOV (SP),2#STMP2 ;SAVE PC OF TRAP.
9639 042641 022526 CMP (SP)+,(SP)+
9640 042646 104377 IS: ERROR 377
9641 042650 000443 .WORD 443
9642 042652 104412 RSETUP

:GO INITIALIZE THE FPS AND STACK; AND
:SEE IF THE USER HAS EXPRESSED
:THE DESIRE TO CHANGE THE SOFTWARE
:VIRTUAL CONSOLE SWITCH REGISTER .HAS
:THE USER TYPED CONTROL G?).

9643
9644
9645
9646 042654 000137 037142 JMP 2#SEOP

.SBTTL SET LOOP ON ERROR ADDRESS ROUTINE

*

9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657 042660 011637 001110 LPER: MOV (SP),2#SLPERR
9658 042664 000002 RTI

.SBTTL FLAG RESET AND CONSOLE TEST ROUTINE

*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
*CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
*TELETYPE AND THE USER CAN MODIFY IT.
*

9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670 042666 023727 001140 177570 RSET: CMP 2#SWR,#177570 ;SEE IF THERE IS A PHYSICAL
9671
9672 042674 001001 BNE IS ;CONSOLE SWITCH REGISTER.
9673 042676 104406 CKSWR ;BRANCH IF NO.
9674 ;OTHERWISE TYPE THE CONTENTS
9675 ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
9676 ;AND GIVE THE USER A CHANCE TO
9677 ;MODIFY IT.

9678 042700 012737 042564 000244 IS: MOV #FPSPUR,2#FPVECT
9679 042706 012737 042620 000004 MOV #CPSPUR,2#ERRVECT
9680 042714 012737 042640 000010 MOV #CPTWO,2#10
9681 042722 011600 MOV (SP),R0 ;SAVE RETURN ADDRESS.
9682 042724 012706 001100 MOV #STACK,SP ;RESET THE STACK POINTER.
9683 042730 005004 CLR R4 ;CLEAR THE FPS.
9684 042732 170104 LDFPS R4
9685 042734 000110 JMP (R0) ;RETURN.

9695
9696
9697

.NLIST BEX

;THESE ARE SPECIAL MESSAGES:

042736	050200	053517	051105	POWERM:	.ASCIZ	<CR LF>'POWER FAILURE. PROGRAM RESTARTING.'
043032	020040	000		SPACE:	.ASCIZ	
043005	011	000		\$TAB:	.ASCIZ	<TAB>
043007	107	052117	051040	MS1:	.ASCIZ	'GOT RESULT:'<TAB><TAB>
043025	105	050130	041505	MS2:	.ASCIZ	'EXPECTED RESULT:'<TAB>
043047	101	020103	050117	MS3:	.ASCIZ	'AC OPERAND:'<TAB><TAB>
043065	123	052517	041522	MS4:	.ASCIZ	'SOURCE OPERAND:'<TAB>
	043047			MS10=MS3		
043107	105	050130	047117	MS11:	.ASCIZ	'EXPONENT OPERAND:'<TAB>

;THESE ARE ERROR MESSAGES:

(0)	043132	052123	020106	026101	EM1:	.ASCIZ	'STF A,AC7 DID NOT TRAP. FID=0.'
(1)	043171	123	043124	040440	EM2:	.ASCIZ	'STF A,AC7. FPS BAD. FID=0.'
(0)	043224	052123	020106	026101	EM3:	.ASCIZ	'STF A,AC7. FEC BAD. FID=0.'
(1)	043257	123	043124	040440	EM4:	.ASCIZ	'STF A,(R). RO BAD. FDST FAILED.'
(0)	043317				EMS:	.ASCII	'STF A,(R) FAILED.'
(1)	043317	123	043124	040440		.BYTE	0
(0)	043341				EM6:	.ASCII	'STF A,(R). FDST FAILED.'
(1)	043341	123	043124	040440		.ASCIZ	'<CR LF>'(BUT FD) ST 707 WENT TO 245 INSTEAD OF 244.'
(0)	043370	024200	052502	020124	EM7:	.ASCIZ	'STF A,(R)+. RO BAD. FDST FAILED.'
(1)	043445	123	043124	040440	EM10:	.ASCII	'STF A,(R)+ FAILED.'
(0)	043506					.BYTE	0
(1)	043506	052123	020106	026101	EM11:	.ASCIZ	'STD A,(R)+. RO BAD. FDST FAILED.'
(0)	043530	000			EM12:	.ASCII	'STD A,(R)+ FAILED.'
(1)	043531	123	042124	040440		.BYTE	0
(0)	043572				EM13:	.ASCIZ	'STD A,#N TRAP TO 4 IN FDST.'
(1)	043572	052123	020104	026101	EM14=EM13		
(0)	043614	000			EM15:	.ASCII	'STD A,#N FAILED.'
(1)	043615	123	042124	040440		.BYTE	0
(0)	043651				EM16:	.ASCIZ	'PC BAD AFTER STD A,#N.'
(1)	043651	123	042124	040440	EM17:	.ASCIZ	'STD A,-(R) TRAP TO 4 IN FDST.'
(0)	043671	000			EM20:	.ASCIZ	'STD A,-(R). RO BAD. FDST FAILED.'
(1)	043672	041520	041040	042101	EM21:	.ASCIZ	'STD A,-(R) FAILED.'
(0)	043721				EM22=EM21		
(1)	043721	123	042124	040440	EM23:	.ASCIZ	'STD A,2(R)+ TRAP TO 4 IN FDST.'
(0)	043757					.BYTE	0
(1)	043757	123	042124	040440			
(0)	044020						
(1)	044020	052123	020104	026101			
(0)	044042	000					
(1)	044042	044020					
(0)	044043						
(1)	044043	123	042124	040440			

H14

(0)	044102				EM24:		
(1)	044102	052123	020104	026101		.ASCIZ	\STD A,2(R)+. RO BAD. FDST FAILED.
(0)	044144				EM25:		
(1)	044144	052123	020104	026101		.ASCII	\STD A,2(R)+ FAILED.\
(0)	044167	000				.BYTE	0
(0)	044170				EM26:		
(1)	044170	052123	020104	026101		.ASCIZ	\STD A,2-(R) TRAP TO 4 IN FDST.\
(0)	044227				EM27:		
(1)	044227	123	042124	040440		.ASCIZ	\STD A,2-(R). RC BAD. FDST FAILED.\
(0)	044271				EM30:		
(1)	044271	123	042124	040440		.ASCII	\STD A,2-(R) FAILED.\
(0)	044314	000				.BYTE	0
(0)	044315				EM31:		
(1)	044315	123	042124	040440		.ASCIZ	\STD A,N(R) TRAP TO 4 IN FDST.\
(0)	044353				EM32:		
(1)	044353	123	042124	040440		.ASCIZ	\STD A,N(R). RO BAD. FDST FAILED.\
(0)	044414				EM33:		
(1)	044414	052123	020104	026101		.ASCII	\STD A,N(R) FAILED.\
(0)	044436	000				.BYTE	0
(0)	044437				EM34:		
(1)	044437	123	042124	040440		.ASCIZ	\STD A,2N(R) TRAP TO 4 IN FDST.\
(0)	044476				EM35:		
(1)	044476	052123	020104	026101		.ASCIZ	\STD A,2N(R). RO BAD. FDST FAILED.\
(0)	044540				EM36:		
(1)	044540	052123	020104	026101		.ASCII	\STD A,2N(R) FAILED.\
(0)	044563	000				.BYTE	0
(0)	044564				EM37:		
(1)	044564	052123	043103	020104		.ASCII	'STCFD A,(R) FAILED.'
(0)	044607	000				.BYTE	0
(0)	044610				EM40:		
(1)	044610	052123	043103	020104		.ASCII	\STCFD A,(R). FPS BAD.\
(0)	044635	000				.BYTE	0
(0)	044636				EM41:		
(1)	044636	052123	043103	020104		.ASCII	\STCFD A,(R). FEC BAD.\
(0)	044663	000				.BYTE	0
(0)	044664				EM42:		
(1)	044664	052123	043103	020104		.ASCII	'STCFD A,(R) FAILED.'
(0)	044707	200	047111	042526		.ASCIZ	<CRLF>'INVERT FDFL ST 767 FAILED.'
(0)	044743				EM43:		
(1)	044743	123	041524	042106		.ASCII	\STCFD A,(R). FPS BAD.\
(0)	044770	024200	052502	020124		.ASCIZ	<CRLF>\(BUT EZBT) ST 560 WENT TO 061 INSTEAD OF 261.\
(0)	045047				EM44:		
(1)	045047	123	041524	042106		.ASCII	'STCFD A,(R) FAILED.'
(0)	045072	046200	053517	047440		.ASCIZ	<CRLF>'LOW ORDER BITS OF X11 DID NOT GET C ST 766.'
(0)	045147				EM45:		
(1)	045147	123	041524	042106		.ASCII	'STCFD A,(R) FAILED.'
(0)	045172	024200	052502	020124		.ASCIZ	<CRLF>'(BUT OPIC) ST 251 FAILED.'
(0)	045225				EM46:		
(1)	045225	123	041524	042106		.ASCII	\STCFD A,(R). FPS BAD.\
(0)	045252	024200	052502	020124		.ASCIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 262 INSTEAD OF 062.\
(0)	045331				EM47:		
(1)	045331	123	041524	042106		.ASCII	'STCFD A,(R) FAILED.'
(0)	045354	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FD) ST 113 WENT TO 415 INSTEAD OF 414.\
(0)	045431				EM50:		
(1)	045431	123	041524	042106		.ASCII	'STCFD A,(R) FAILED.'
(0)	045454	051440	043511	020116		.ASCII	'SIGN BAD.'

(0)	045466	024200	052502	020124	EMS1:	.ASCIZ	<CRLF>\(BUT ENBT) ST 567 WENT TO 060 INSTEAD OF 460.\
(1)	045545	123	041524	043104		.ASCII	'STCDF A,(R) FAILED.'
(0)	045570	000				.BYTE	0
(1)	045571	123	042124	040440	EMS2:	.ASCII	\STD A,(R). FPS BAD.\
(0)	045614	000				.BYTE	0
(1)	045615	123	042124	040440	EMS3:	.ASCII	\STD A,(R). FEC BAD.\
(0)	045640	000				.BYTE	0
(1)	045641	123	041524	043104	EMS4:	.ASCII	'STCDF A,(R) FAILED.'
(0)	045664	044600	053116	051105		.ASCIZ	<CRLF>'INVERT FDFL ST 767 FAILED.'
(1)	045720	052123	042103	020106	EMS5:	.ASCII	'STCDF A,(R) FAILED.'
(0)	045743	200	047522	047125		.ASCII	<CRLF>'ROUND ERROR, OR'
(1)	045763	200	041050	052125		.ASCIZ	<CRLF>\(BUT BREAKOUT) ST 400 WENT TO 766 INSTEAD OF 767.\
(0)	046046				EMS6:	.ASCII	\STD A,(R). FPS BAD.\
(1)	046071	200	041050	052125		.ASCIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 062 INSTEAD OF 262.\
(0)	046150				EMS7:	.ASCII	\STD A,(R). FPS BAD.\
(1)	046173	040	044506	036526		.ASCII	'FIV=0.'
(0)	046202	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 123 INSTEAD OF 103.\
(1)	046260	052123	042103	020106	EM60:	.ASCII	'STCDF A,(R) FAILED.'
(0)	046303	040	044506	036526		.ASCII	'FIV=1.'
(1)	046312	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 103 INSTEAD OF 123.\
(0)	046370				EM61:	.ASCII	\STD A,(R). FPS BAD.\
(1)	046413	200	041050	052125		.ASCIZ	<CRLF>\(BUT FLAG) ST 147 WENT TO 361 INSTEAD OF 365.
(0)	046472	052123	043103	020104	EM62:	.ASCII	'STCFD A,AC6. FPS BAD.'
(1)	046517	200	041050	052125		.ASCIZ	<CRLF>\(BUT FDST) ST 767 WENT TO 567 INSTEAD OF 577.\
(0)	046576	052123	043103	020104	EM63:	.ASCIZ	'STCFD A,AC6. FEC BAD.'
(1)	046624	046103	042122	024040	EM64:	.ASCII	\CLRD (R) FAILED.\
(0)	046644	055200	051105	020117		.ASCIZ	<CRLF>'ZERO XII AT ST 770 FAILED.'
(1)	046700	046103	042122	024040	EM65:	.ASCII	\CLRD (R). FPS BAD.\
(0)	046722	000				.BYTE	0
(1)	046723	103	051114	020104	EM66:	.ASCIZ	\CLRD (R). RO BAD. FDST FAILED.\
(0)	046762				EM67:	.ASCII	\CLRD AC7. FPS BAD.\
(1)	046762	046103	042122	040440		.ASCIZ	<CRLF>\(BUT FDST) ST 770 WENT TO 607 INSTEAD OF 617.\
(0)	047004	024200	052502	020124	EM70:	.ASCII	\CLRD AC7. FEC BAD.\
(1)	047063	103	051114	020104		.BYTE	0
(0)	047105	000				.ASCIZ	'NEGF AC7. FPS BAD.'
(1)	047106	042516	043107	040440	EM176:	.ASCIZ	'NEGF AC7. FEC BAD.'
(0)	047131	116	043505	020106	EM71:	.ASCIZ	\NEGF A FAILED.\
(1)	047154	042516	043107	040440		.ASCIZ	\NEGF A. FPS BAD.\
(0)	047173				EM72:	.ASCIZ	\NEGF A. FPS BAD.\
(1)	047173	116	043505	020106	EM107:	.ASCIZ	\NEGD (R) TRAP TO 4 IN SRC MODE.\
(0)	047214						
(1)	047214	042516	042107	024040			

(0)	047254				EM73:	
(1)	047254	042516	042107	024040	.ASCIZ	\NEGD (R) FAILED.\
(0)	047275				EM74:	
(1)	047275	116	043505	020104	.ASCIZ	\NEGD (R). RO BAD.\
(0)	047317				EM75:	
(1)	047317	116	043505	020104	.ASCIZ	\NEGD (R). FPS BAD.\
(0)	047342				EM76:	
(1)	047342	041101	042123	024040	.ASCIZ	\ABSD (R)+ TRAP TO 4 IN SRC MODE.\
(0)	047403				EM77:	
(1)	047403	101	051502	020104	.ASCIZ	\ABSD (R)+ FAILED.\
(0)	047425				EM100:	
(1)	047425	101	051502	020104	.ASCIZ	\ABSD (R)+. RO BAD.\
(0)	047450				EM101:	
(1)	047450	041101	042123	024040	.ASCIZ	\ABSD (R)+. FPS BAD.\
(0)	047474				EM102:	
(1)	047474	041101	042123	026440	.ASCIZ	\ABSD -(R) TRAP TO 4 IN SRC MODE.\
(0)	047535				EM103:	
(1)	047535	101	051502	020104	.ASCIZ	\ABSD -(R) FAILED.\
(0)	047557				EM104:	
(1)	047557	101	051502	020104	.ASCIZ	\ABSD -(R). RO BAD.\
(0)	047602				EM105:	
(1)	047602	041101	042123	026440	.ASCIZ	\ABSD -(R). FPS BAD.\
(0)	047626				EM106:	
(1)	047626	041101	042123	040040	.ASCIZ	\ABSD @ (R)+ TRAP TO 4 IN SRC MODE.\
(0)	047670				EM110:	
(1)	047670	041101	042123	040040	.ASCIZ	\ABSD @ (R)+ FAILED.\
(0)	047713				EM111:	
(1)	047713	101	051502	020104	.ASCIZ	\ABSD @ (R)+. RO BAD.\
(0)	047737				EM112:	
(1)	047737	101	051502	020104	.ASCIZ	\ABSD @ (R)+. FPS BAD.\
(0)	047764				EM113:	
(1)	047764	042516	042107	040040	.ASCIZ	\NEGD @-(R) TRAP TO 4 IN SRC MODE.\
(0)	050026				EM114:	
(1)	050026	042516	042107	040040	.ASCIZ	\NEGD @-(R) FAILED.\
(0)	050051				EM115:	
(1)	050051	116	043505	020104	.ASCIZ	\NEGD @-(R). RO BAD.\
(0)	050075				EM116:	
(1)	050075	116	043505	020104	.ASCIZ	\NEGD @-(R). FPS BAD.\
(0)	050122				EM117:	
(1)	050122	041101	042123	047040	.ASCIZ	\ABSD N(R) TRAP TO 4 IN SRC MODE.\
(0)	050163				EM120:	
(1)	050163	101	051502	020104	.ASCIZ	\ABSD N(R) FAILED.\
(0)	050205				EM121:	
(1)	050205	101	051502	020104	.ASCIZ	\ABSD N(R). RO BAD.\
(0)	050230				EM122:	
(1)	050230	041101	042123	047040	.ASCIZ	\ABSD N(R). FPS BAD.\
(0)	050254				EM123:	
(1)	050254	042516	042107	040040	.ASCIZ	\NEGD @N(R) TRAP TO 4 IN SRC MODE.\
(0)	050316				EM124:	
(1)	050316	042516	042107	040040	.ASCIZ	\NEGD @N(R) FAILED.\
(0)	050341				EM125:	
(1)	050341	116	043505	020104	.ASCIZ	\NEGD @N(R). RO BAD.\
(0)	050365				EM126:	
(1)	050365	116	043505	020104	.ASCIZ	\NEGD @N(R). FPS BAD.\
(0)	050412				EM127:	
(1)	050412	042516	042107	047040	.ASCIZ	\NEGD N(R?) TRAP TO 4 IN SRC MODE.\

K14

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 179
 DFFPCA.P11 31-OCT-76 17:16 FLAG RESET AND CONSOLE TEST ROUTINE

(0)	050454				EM130:	
(1)	050454	042516	042107	047040	.ASCIZ	\NEGD N(R7) FAILED.\
(0)	050477				EM131:	
(1)	050477	116	043505	020104	.ASCIZ	\NEGD N(R7).. FPS BAD.\
(0)	050524				EM132:	
(1)	050524	041101	042123	040040	.ASCIZ	\ABSD 2N(R7) TRAP TO 4 IN SRC MODE.\
(0)	050567				EM133:	
(1)	050567	101	051502	020104	.ASCIZ	\ABSD 2N(R7) FAILED.\
(0)	050613				EM134:	
(1)	050613	101	051502	020104	.ASCIZ	\ABSD 2N(R7).. FPS BAD.\
	050641	116	043505	020104	EM135:	.ASCII 'NEGD A FAILED.'
	050657	200	047530	020122	.ASCIZ	<CRLF>'XOR SIGN BIT ST 336 FAILED.'
(0)	050714				EM136:	
(1)	050714	042516	042107	040440	.ASCIZ	\NEGD A FAILED.\
(0)	050733				EM137:	
(2)	050733	116	043505	020104	.ASCIZ	\NEGD A. FPS BAD.\
(0)	050754				EM140:	
(1)	050754	042516	042107	024040	.ASCIZ	\NEGD (R) FAILED.\
(0)	050775				EM141:	
(1)	050775	116	043505	020104	.ASCIZ	\NEGD (R). RO BAD. SPECIAL DEST FAILED.\
(0)	051044				EM142:	
(2)	051044	042516	042107	024040	.ASCIZ	\NEGD (R). FPS BAD.\
(0)	051067				EM143:	
(1)	051067	116	043505	020104	.ASCIZ	\NEGD (R)+ FAILED.\
(0)	051111				EM144:	
(1)	051111	116	043505	020104	.ASCIZ	\NEGD (R)+. RO BAD. SPECIAL DEST FAILED.\
(0)	051161				EM145:	
(2)	051161	116	043505	020104	.ASCIZ	\NEGD (R)+. FPS BAD.\
(0)	051205				EM146:	
(1)	051205	116	043505	020104	.ASCIZ	\NEGD -(R) FAILED.\
(0)	051227				EM147:	
(1)	051227	116	043505	020104	.ASCIZ	\NEGD -(R). RO BAD. SPECIAL DEST FAILED.\
(0)	051277				EM150:	
(2)	051277	116	043505	020104	.ASCIZ	\NEGD -(R). FPS BAD.\
(0)	051323				EM151:	
(1)	051323	116	043505	020104	.ASCIZ	\NEGD 2(R)+ FAILED.\
(0)	051346				EM152:	
(1)	051346	042516	042107	040040	.ASCIZ	\NEGD 2(R)+. RO BAD. SPECIAL DEST FAILED.\
(0)	051417				EM153:	
(2)	051417	116	043505	020104	.ASCIZ	\NEGD 2(R)+. FPS BAD.\
(0)	051444				EM154:	
(1)	051444	042516	042107	040040	.ASCIZ	\NEGD 2-(R) FAILED.\
(0)	051467				EM155:	
(1)	051467	116	043505	020104	.ASCIZ	\NEGD 2-(R). RO BAD. SPECIAL DEST FAILED.\
(0)	051540				EM156:	
(2)	051540	042516	042107	040040	.ASCIZ	\NEGD 2-(R). FPS BAD.\
(0)	051565				EM157:	
(1)	051565	116	043505	020106	.ASCIZ	\NEGF (R)+ FAILED.\
	051607	116	043505	020106	EM160:	.ASCII 'NEGF (R)+. RO BAD.'
	051631	102	042101	041440	.ASCIZ	'BAD CONSTANT USED. SPECIAL DEST FAILED.'
(0)	051701				EM161:	
(2)	051701	116	043505	020106	.ASCIZ	\NEGF (R)+. FPS BAD.\
(0)	051725				EM162:	
(1)	051725	116	043505	020104	.ASCIZ	\NEGD (R7)+ FAILED.\
(0)	051750				EM163:	
(2)	051750	042516	042107	024040	.ASCIZ	\NEGD (R7)+. FPS BAD.\

(0)	051775	120	020103	040502	EM164:	.ASCIZ	'PC BAD AFTER NEG0 (R7)+. BAD CONSTANT USED.'
(1)	052051				EM215:	.ASCII	\PC BAD AFTER NEG0 N(R). BAD CONSTANT USED 746 746.\
(1)	052133	200	051117	024040		.ASCIZ	<CRLF>'OR (BLT F0ST) IN SPECIAL DEST FAILED.'
(0)	052202				EM216:	.ASCIZ	\NEG0 N(R) FAILED.\
(1)	052202	042516	042107	047040		.ASCIZ	\NEG0 N(R). RO BAD. SPECIAL DEST FAILED.\
(0)	052224				EM217:	.ASCIZ	\NEG0 N(R). FPS BAD.\
(1)	052224	042516	042107	047040		.ASCIZ	\NEG0 N(R). FPS BAD.\
(0)	052274				EM220:	.ASCIZ	\NEG0 N(R). FPS BAD.\
(2)	052274	042516	042107	047040		.ASCIZ	\NEG0 N(R). FPS BAD.\
(0)	052320				EM221:	.ASCII	\PC BAD AFTER NEG0 N(R). BAD CONSTANT USED 747 747.\
(1)	052320	041520	041040	042101		.ASCIZ	<CRLF>'OR (BUT F0ST) IN SPECIAL DEST FAILED.'
(1)	052403	200	051117	024040		.ASCIZ	<CRLF>'OR (BUT F0ST) IN SPECIAL DEST FAILED.'
(0)	052452				EM222:	.ASCIZ	\NEG0 N(R) FAILED.\
(1)	052452	042516	042107	040040		.ASCIZ	\NEG0 N(R). RO BAD. SPECIAL DEST FAILED.\
(0)	052475				EM223:	.ASCIZ	\NEG0 N(R). FPS BAD.\
(1)	052475	116	043505	020104		.ASCIZ	\NEG0 N(R). FPS BAD.\
(0)	052546				EM224:	.ASCIZ	\NEG0 N(R). FPS BAD.\
(2)	052546	042516	042107	040040		.ASCIZ	\NEG0 N(R). FPS BAD.\
(0)	052573				EM165:	.ASCIZ	\NEG0 (R) FAILED.\
(1)	052573	116	043505	020104		.ASCIZ	\ABSD (R) FAILED.\
(0)	052614				EM166:	.ASCIZ	\TSTD (R) FAILED.\
(1)	052614	041101	042123	024040		.ASCIZ	\TSTD (R). FPS BAD.\
(0)	052635				EM167:	.ASCIZ	\TSTD (R). FPS BAD.\
(1)	052635	124	052123	020104		.ASCIZ	\TSTD (R). FPS BAD.\
(0)	052656				EM170:	.ASCIZ	\NEG0 (R). FEC BAD.\
(1)	052656	042516	042107	024040		.ASCIZ	\NEG0 (R). FEC BAD.\
(0)	052701				EM171:	.ASCIZ	\ABSD (R). FPS BAD.\
(1)	052701	101	051502	020104		.ASCIZ	\ABSD (R). FPS BAD.\
(0)	052724				EM172:	.ASCIZ	\TSTD (R). FPS BAD.\
(1)	052724	051524	042124	024040		.ASCIZ	\TSTD (R). FPS BAD.\
(0)	052747				EM173:	.ASCIZ	\NEG0 (R). FEC BAD.\
(1)	052747	116	043505	020104		.ASCIZ	\ABSD (R). FEC BAD.\
(0)	052772				EM174:	.ASCIZ	\ABSD (R). FEC BAD.\
(1)	052772	041101	042123	024040		.ASCIZ	\TSTD (R). FEC BAD.\
(0)	053015				EM175:	.ASCIZ	\TSTD (R). FEC BAD.\
(1)	053015	124	052123	020104		.ASCIZ	\NEG0 (R) FAILED.\
(0)	053040				EM200:	.ASCIZ	<CRLF>'XOR SIGN BIT FAILED ST 336.'
(1)	053040	042516	042107	024040		.ASCIZ	<CRLF>'XOR SIGN BIT FAILED ST 336.'
(0)	053060	054200	051117	051440		.ASCIZ	\NEG0 (R). FPS BAD.\
(0)	053115				EM201:	.ASCII	\NEG0 (R). FPS BAD.\
(1)	053115	116	043505	020104		.ASCII	\NEG0 (R). FPS BAD.\
(1)	053137	200	041050	052125		.ASCIZ	<CRLF>\(BUT ENBT) ST 336 WENT TO 053 INSTEAD OF 453.\
(0)	053216				EM202:	.ASCIZ	<CRLF>\(BUT ENBT) ST 336 WENT TO 453 INSTEAD OF 053.\
(1)	053216	042516	042107	024040		.ASCIZ	\NEG0 (R). FPS BAD.\
(1)	053240	024200	052502	020124		.ASCIZ	<CRLF>\(BUT ENBT) ST 336 WENT TO 453 INSTEAD OF 053.\
(0)	053317				EM203:	.ASCII	\ABSD (R) FAILED.\
(1)	053317	101	051502	020104		.ASCII	\ABSD (R) FAILED.\
(0)	053337	200	041050	052125		.ASCII	<CRLF>'(BUT OPIB) ST 055 WENT TO 336 INSTEAD OF 335, OR'
(1)	053420	024200	052502	020124		.ASCIZ	<CRLF>\(BUT ENBT) ST 335 WENT TO 452 INSTEAD OF 052.\
(0)	053477				EM204:	.ASCIZ	\ABSD (R) FAILED.\
(1)	053477	101	051502	020104		.ASCII	\ABSD (R) FAILED.\
(0)	053517	200	047530	020122		.ASCIZ	<CRLF>'XOR SIGN BIT FAILED ST 452.'
(0)	053554				EM205:	.ASCIZ	\TSTD (R) FAILED.\
(1)	053554	051524	042124	024040		.ASCII	\TSTD (R) FAILED.\
(1)	053574	024200	052502	020124		.ASCIZ	<CRLF>\(BUT OPIB) ST 055 WENT TO 336 INSTEAD OF 334.\

M14

(0)	053653				EM206:	
(1)	053653	124	052123	020104	.ASCII	\TSTD (R). FPS BAD.\
(1)	053675	200	041050	052125	.ASCIZ	<CRLF>\(BUT ENBT) ST 334 WENT TO 453 INSTEAD OF 053.\
(0)	053754				EM207:	
(1)	053754	051524	042124	024040	.ASCII	\TSTD (R) FAILED.\
(1)	053774	024200	052502	020124	.ASCIZ	<CRLF>\(BUT OP1B) ST 057 WENT TO 335 INSTEAD OF 334.\
(0)	054053				EM210:	
(1)	054053	124	052123	020104	.ASCII	\TSTD (R) FAILED.\
(1)	054073	200	041050	052125	.ASCIZ	<CRLF>\(BUT ENBT) ST 334 WENT TO 053 INSTEAD OF 453.\
(0)	054152				EM211:	
(1)	054152	051524	042124	024040	.ASCII	\TSTD (R) FAILED.\
(1)	054172	024200	052502	020124	.ASCIZ	<CRLF>\(BUT OP1B) ST 255 WENT TO 311 OR 312 INSTEAD OF 310.\
(0)	054260				EM212:	
(1)	054260	051524	042124	024040	.ASCII	\TSTD (R). FPS BAD.\
(1)	054302	024200	052502	020124	.ASCIZ	<CRLF>\(BUT ENBT) ST 310 WENT TO 402 INSTEAD OF 002.\
(0)	054361				EM213:	
(1)	054361	124	052123	020104	.ASCII	\TSTD (R). FPS BAD.\
(1)	054403	040	044506	053125	.ASCII	' FIUV=0, OPERAND=-0.'
(1)	054427	200	041050	052125	.ASCIZ	<CRLF>\(BUT FIUV) ST 257 WENT TO 355 INSTEAD OF 255.\
(0)	054506				EM214:	
(1)	054506	051524	042124	024040	.ASCII	\TSTD (R). FPS BAD.\
(1)	054530	043040	052511	036526	.ASCII	' FIUV=1, OPERAND=-0.'
(1)	054554	024200	052502	020124	.ASCIZ	<CRLF>\(BUT FIUV) ST 257 WENT TO 255 INSTEAD OF 355.\
(0)	054633				EM225:	
(1)	054633	114	043104	051520	.ASCIZ	\LDFPS (R). RO BAD.\
(0)	054656				EM226:	
(1)	054656	042114	050106	020123	.ASCIZ	\LDFPS (R). FPS BAD.\
(0)	054702				EM227:	
(1)	054702	042114	050106	020123	.ASCIZ	\LDFPS (R) TRAPPED TO 4.\
(0)	054732				EM230:	
(1)	054732	042114	050106	020123	.ASCIZ	\LDFPS (R)+. RO BAD.\
(0)	054756				EM231:	
(1)	054756	042114	050106	020123	.ASCIZ	\LDFPS (R)+. FPS BAD.\
(0)	055003				EM232:	
(1)	055003	114	043104	051520	.ASCIZ	\LDFPS (R)+ TRAPPED TO 4.\
(0)	055034				EM233:	
(1)	055034	042114	050106	020123	.ASCIZ	\LDFPS -(R). RO BAD.\
(0)	055060				EM234:	
(1)	055060	042114	050106	020123	.ASCIZ	\LDFPS -(R). FPS BAD.\
(0)	055105				EM235:	
(1)	055105	114	043104	051520	.ASCIZ	\LDFPS -(R) TRAPPED TO 4.\
(0)	055136				EM236:	
(1)	055136	042114	050106	020123	.ASCIZ	\LDFPS @ (R)+. RO BAD.\
(0)	055163				EM237:	
(1)	055163	114	043104	051520	.ASCIZ	\LDFPS @ (R)+. FPS BAD.\
(0)	055211				EM240:	
(1)	055211	114	043104	051520	.ASCIZ	\LDFPS @ (R)+ TRAPPED TO 4.\
(0)	055243				EM241:	
(1)	055243	114	043104	051520	.ASCIZ	\LDFPS @-(R). RO BAD.\
(0)	055270				EM242:	

N14

(1)	055270	042114	050106	020123		.ASCIZ	\LDFPS 2-(R). FPS BAD.\
(0)	055316				EM243:		
(1)	055316	042114	050106	020123		.ASCIZ	\LDFPS 2-(R) TRAPPED TO 4.\
(0)	055350				EM244:		
(1)	055350	042114	050106	020123		.ASCIZ	\LDFPS N(R). RO BAD.\
(0)	055374				EM245:		
(1)	055374	042114	050106	020123		.ASCIZ	\LDFPS N(R). FPS BAD.\
(0)	055421				EM246:		
(1)	055421	120	020103	040502		.ASCIZ	\PC BAD AFTER LDFPS N(R).\
(0)	055452				EM247:		
(1)	055452	042114	050106	020123		.ASCIZ	\LDFPS N(R) TRAPPED TO 4.\
(0)	055503				EM250:		
(1)	055503	114	043104	051520		.ASCIZ	\LDFPS 2N(R). RO BAD.\
(0)	055530				EM251:		
(1)	055530	042114	050106	020123		.ASCIZ	\LDFPS 2N(R). FPS BAD.\
(0)	055556				EM252:		
(1)	055556	041520	041040	042101		.ASCIZ	\PC BAD AFTER LDFPS 2N(R).\
(0)	055610				EM253:		
(1)	055610	042114	050106	020123		.ASCIZ	\LDFPS 2N(R) TRAPPED TO 4.\
(0)	055642				EM254:		
(1)	055642	041520	041040	042101		.ASCIZ	\PC BAD AFTER LDCLD (R)+,A.\
(0)	055676				EM255:		
(1)	055676	042114	046103	020104		.ASCIZ	\LDCLD (R)+,A TRAPPED TO 4.\
(0)	055732				EM256:		
(1)	055732	042114	046103	020104		.ASCIZ	\LDCLD (R)+,A. RO BAD.\
(0)	055760				EM257:		
(1)	055760	042114	046103	020104		.ASCIZ	\LDCLD (R)+,A. FPS BAD.\
(0)	056007				EM260:		
(1)	056007	114	041504	043111		.ASCII	\LDCIF OR LDCLF (R),A FAILED.\
	056043	000				.BYTE	0
(0)	056044				EM261:		
(1)	056044	042114	044503	020106		.ASCII	\LDCIF OR LDCLF (R),A. FPS BAD.\
	056102	000				.BYTE	0
(0)	056103				EM262:		
(1)	056103	114	041504	043111		.ASCII	\LDCIF (R),A FAILED.\
(1)	056126	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
(0)	056203				EM263:		
(1)	056203	114	041504	043114		.ASCII	\LDCLF (R),A. FPS BAD.\
	056230	000				.BYTE	0
(0)	056231				EM264:		
(1)	056231	114	041504	043111		.ASCII	\LDCIF (R),A FAILED.\
	056254	052600	042523	020104		.ASCIZ	<CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
(0)	056326				EM265:		
(1)	056326	042114	044503	020106		.ASCII	\LDCIF OR LDCLF (R),A FAILED.\
	056362	051600	052105	051440		.ASCIZ	<CRLF>'SET SIGN BIT FAILED ST 146.'

B15

(0)	056417				EM266:	
(1)	056417	114	041504	042111	.ASCII	\LDCLF OR LDCLF (R), A FAILED.
(1)	056453	200	041050	052125	.ASCII2	<CR LF> \BL* 372 WENT TO 152 INSTEAD OF 112.
(0)	056530				EM267:	
(1)	056530	042114	046103	020106	.ASCII	\LDCLF (R), A FAILED.
(1)	056555	200	051525	042106	.ASCII2	<CR LF> \USED CONSTANT 217 INSTEAD OF 237 ST 107.
(0)	056627				EM270:	
(1)	056627	114	041504	042114	.ASCII	\LDCLF (R), A FAILED.
(1)	056652	051040	052517	042116	.ASCII2	*ROUND ERROR.*
(0)	056673				EM271:	
(1)	056673	042114	046103	020106	.ASCII	\LDCLF (R), A FAILED.\
(1)	056713	040	051124	047125	.ASCII2	*TRUNCATION ERROR.*
(0)	056736				EM272:	
(1)	056736	042114	044503	020106	.ASCII	\LDCLF OR LDCLF (R), A FAILED.\
(1)	056772	051200	032061	047040	.ASCII2	<CR LF> \R14 NOT INCREMENTED ST 630.*
(0)	057027				EM273:	
(1)	057027	114	041504	042111	.ASCII	\LDCID OR LDCLD (R), A FAILED.\
(1)	057063	200			.BYTE	0
(0)	057064				EM274:	
(1)	057064	042114	044503	020104	.ASCII	\LDCID OR LDCLD (R), A. FPS BAD.\
(1)	057122	200			.BYTE	0
(0)	057123				EM275:	
(1)	057123	114	041504	042111	.ASCII	\LDCID (R), A FAILED.\
(1)	057146	024200	052502	020124	.ASCII2	<CR LF> \ (BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
(0)	057223				EM276:	
(1)	057223	114	041504	042111	.ASCII	\LDCID (R), A FAILED.\
(1)	057246	052600	042523	020104	.ASCII2	<CR LF> \USED CONSTANT 237 INSTEAD OF 217 ST 107.*
(0)	057320				EM277:	
(1)	057320	042114	044503	020104	.ASCII	\LDCID (R), A FAILED.\
(1)	057343	200	042523	020124	.ASCII2	<CR LF> \SET SIGN FAILED ST 146.*
(0)	057374				EM300:	
(1)	057374	042114	046103	020104	.ASCII	\LDCLD (R), A FAILED.\
(1)	057417	200	051525	042105	.ASCII2	<CR LF> \USED CONSTANT 217 INSTEAD OF 237 ST 107.*
(0)	057471				EM301:	
(1)	057471	114	042504	050130	.ASCII	\LDEXP (R), A FAILED.\
(1)	057514	200			.BYTE	0
(0)	057515				EM302:	
(1)	057515	114	042504	050130	.ASCII	\LDEXP (R), A. FPS BAD.\
(1)	057542	200			.BYTE	0
(1)	057543	114	042504	050130	.ASCII2	*LDEXP (R), A. FEC BAD.*
(0)	057571				EM304:	
(1)	057571	114	042504	050130	.ASCII	\LDEXP (R), A FAILED.\
(1)	057614	042600	041530	051505	.ASCII2	<CR LF> \EXCESS 200 CALCULATION ST 624 BAD.*

C15

(0)	057760				EM305:		
(1)	057760	042114	054105	020120	.ASCII	\LDEXP (R) A. FPS BAC.\	
	057763	250	052502	020124	.ASCII	(\CRF)\(BUT EZBT,XNBT) ST 625 DID NOT GO TO 304.\	
(0)	057763				EM306:		
(1)	057763	114	042504	050130	.ASCII	\LDEXP (R) A. FPS BAC.\	
	060010	024200	052502	020124	.ASCII	(\CRF)\(BUT EZBT) ST 544 WENT TO 504 INSTEAD OF 704. OR	
(1)	060071	200	041050	052125	.ASCII2	(\CRF)\(BUT EZBT) ST 704 WENT TO 264 INSTEAD OF 064.\	
(0)	060153				EM307:		
(1)	060153	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	060173	200	041050	052125	.ASCII2	(\CRF)\(BUT EZBT) ST 704 WENT TO 064 INSTEAD OF 264.\	
(0)	060253				EM310:		
(1)	060253	042114	054105	020120	.ASCII	\LDEXP (R) A. FPS BAC.\	
(1)	060277	200	041050	052125	.ASCII2	(\CRF)\(BUT FIU) ST 264 WENT TO 115 INSTEAD OF 155.\	
(0)	060355				EM311:		
(1)	060355	114	042504	050130	.ASCII	\LDEXP (R) A FAILED.\	
(1)	060400	024200	052502	020124	.ASCII2	(\CRF)\(BUT FIU) ST 264 WENT TO 155 INSTEAD OF 115.\	
(0)	060456				EM312:		
(1)	060456	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	060501	200	041050	052125	.ASCII2	(\CRF)\(BUT EZBT) ST 544 WENT TO 704 INSTEAD OF 504.\	
(0)	060560				EM313:		
(1)	060560	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	060603	200	041050	052125	.ASCII2	(\CRF)\(BUT FIU) ST 504 WENT TO 155 INSTEAD OF 115.\	
(0)	060661				EM314:		
(1)	060661	114	042504	050130	.ASCII	\LDEXP (R) A FAILED.\	
(1)	060704	024200	052502	020124	.ASCII2	(\CRF)\(BUT FIV) ST 104 WENT TO 116 INSTEAD OF 136.\	
(0)	060762				EM315:		
(1)	060762	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	061005	200	041050	052125	.ASCII2	(\CRF)\(BUT FIV) ST 104 WENT TO 136 INSTEAD OF 116.\	
(0)	061063				EM316:		
(1)	061063	114	042504	050130	.ASCII	\LDEXP (R) A FAILED.\	
(1)	061106	024200	052502	020124	.ASCII2	(\CRF)\(BUT FIV) ST 144 WENT TO 116 INSTEAD OF 136.\	
(0)	061164				EM317:		
(1)	061164	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	061207	200	041050	052125	.ASCII2	(\CRF)\(BUT FIV) ST 144 WENT TO 136 INSTEAD OF 116.\	
(0)	061265				EM320:		
(1)	061265	114	042504	050130	.ASCII	\LDEXP (R) A FAILED.\	
(1)	061310	024200	052502	020124	.ASCII2	(\CRF)\(BUT FIV) ST 344 WENT TO 116 INSTEAD OF 136.\	
(0)	061366				EM321:		
(1)	061366	042114	054105	020120	.ASCII	\LDEXP (R) A FAILED.\	
(1)	061411	200	041050	052125	.ASCII2	(\CRF)\(BUT FIV) ST 344 WENT TO 136 INSTEAD OF 116.\	
(0)	061467				EM322:		
(1)	061467	123	041524	044504	.ASCII	\STCDI OR STCDL (R) A FAILED.	
	06:523	000			.BYTE	0	

D15

(0) 061524 EM323: .ASCII STCDL (R),A. FPS BAD.\
(1) 061524 052123 042103 020111 .ASCII P *E 0
061562 200

061563 123 041524 044504 EM324: .ASCII 'STCDI OR STCDL (R),A. FEC BAC.'

(0) 061622 EM325: .ASCII \STCDL (R),A. FPS BAD.\
(1) 061622 052123 042103 020114 .ASCII <CRLF> 'CLEAR FLAG ST 774 FAILED, OR'
(1) 061704 024200 052502 020124 .ASCII <CRLF> '(BUT FLAG) ST 662 WENT TO 365 INSTEAD OF 361.\

061622 EM326=EM325

(0) 061763 EM327: .ASCII \STCDL (R),A. FAILED.\
(1) 061763 123 041524 046104 .ASCII <CRLF> '(BUT ENBT) ST 632 WENT TO 473 INSTEAD OF 073.\

062006 024200 052502 020124

(0) 062065 EM330: .ASCII STCDL (R),A. FPS BAD.\
(1) 062065 123 041524 046104 .ASCII <CRLF> '(BUT FIC) ST 004 WENT TO 305 INSTEAD OF 315.\

062112 024200 052502 020124

(0) 062173 EM331: .ASCII \STCDL (R),A. FPS BAD.\
(1) 062173 052123 042103 020114 .ASCII <CRLF> '(BUT FIC) ST 004 WENT TO 315 INSTEAD OF 305.\

062215 200 041050 052126

061524 EM333=EM323

(0) 062273 EM334: .ASCII \STCDL (R),A. FPS BAD.\
(1) 062273 123 041524 044504 .ASCII <CRLF> 'USED CONSTANT 37 INSTEAD OF 17 ST 66.'

062320 052600 042523 020104

(0) 062367 EM335: .ASCII \STCDI (R),A. FAILED.\
(1) 062367 123 041524 044504 .ASCII <CRLF> '(BUT ENBT) ST 632 WENT TO 073 INSTEAD OF 473.\

062412 024200 052502 020124

(0) 062471 EM336: .ASCII \STCDI (R),A. FPS BAD.\
(1) 062471 123 041524 044504 .ASCII <CRLF> 'SET FN ST 473 FAILED.'

062516 051600 052105 043040

(0) 062545 EM337: .ASCII \STCDL (R),A. FAILED.\
(1) 062545 123 041524 046104 .ASCII <CRLF> '(BUT COUT) ST 275 WENT TO 074 INSTEAD OF 274.\

062570 024200 052502 020124

(0) 062647 EM340: .ASCII \STCDL (R),A. FAILED.\
(1) 062647 123 041524 046104 .ASCII <CRLF> '(BUT COUT) ST 275 WENT TO 274 INSTEAD OF 074.\

062672 024200 052502 020124

(0) 062751 EM341: .ASCII \STCDL (R),A. FPS BAD.\
(1) 062751 123 041524 046104 .ASCII <CRLF> '(BUT EZBT) ST 377 WENT TO 633 INSTEAD OF 433.\

062776 024200 052502 020124

(0) 063055 EM342: .ASCII \STCDL (R),A. FAILED.\
(1) 063055 123 041524 046104 .ASCII <CRLF> '(BUT COUT) ST 360 WENT TO 654 INSTEAD OF 454.\

063100 024200 052502 020124

E15

MAINDEC-11-FPP34-A PDP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 186
 OFFPCA.P11 31-OCT-76 17:16 FLAG RESET AND CONSOLE TEST ROUTINE

(0)	063157				EM343:		
(1)	063157	024200	041524	046104	.ASCII	\STCOL (R) A FAILED.\	
(1)	063202	200	052502	020124	.ASCIIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 531 INSTEAD OF 431.	
(0)	063261				EM344:		
(1)	063261	024200	041524	046104	.ASCII	\STCOL (R) A FAILED.\	
(1)	063304	200	052502	020124	.ASCII	<CRLF>\(BUT COUT) ST 360 WENT TO 454 INSTEAD OF 654. OR	
(1)	063365	200	041050	052125	.ASCIIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 431 INSTEAD OF 531.	
(0)	063444				EM332:		
(1)	063444	052123	042103	020111	.ASCII	\STCDI (R) A FAILED.\	
(1)	063467	200	051524	042105	.ASCIIZ	<CRLF>\USED CONSTANT 37 INSTEAD OF 17 ST 66.'	
(0)	063536				EM345:		
(1)	063536	052123	042103	020111	.ASCII	\STCDI (R) A FAILED.\	
(1)	063561	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FL) ST 633 WENT TO 655 INSTEAD OF 654.	
(0)	063636				EM346:		
(1)	063636	052123	043103	020114	.ASCII	\STCFL (R) A FAILED.\	
(1)	063661	200	042532	047522	.ASCIIZ	<CRLF>\ZERO LOW ORDER PART OF X11 FAILED ST 773.'	
(0)	063734				EM347:		
(1)	063734	052123	054105	020120	.ASCII	\STEXP A,(R) FAILED.\	
(1)	063757	000			.BYTE	0	
(0)	063760				EM350:		
(1)	063760	052123	054105	020120	.ASCII	\STEXP A,(R). FPS BAD.\	
(1)	064005	000			.BYTE	0	
	064006	047515	042522	052040	EM351:	.ASCII	'MORE THAN ONE WORD.'
	064031	127	044522	052124	.ASCIIZ	'WRITTEN BY STEXP A,(R).'\<CRLF>'ZERO FDFL ST 347 FAILED.'	
(0)	064112				EM352:		
(1)	064112	052123	054105	020120	.ASCII	\STEXP A,(R). FPS BAD.\	
(1)	064137	200	041050	052125	.ASCIIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 071 INSTEAD OF 471.\	
(0)	064216				EM353:		
(1)	064216	052123	054105	020120	.ASCII	\STEXP A,(R). FPS BAD.\	
(1)	064243	200	041050	052125	.ASCIIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 072 INSTEAD OF 272.\	
(0)	064322				EM354:		
(1)	064322	052123	054105	020120	.ASCII	\STEXP A,(R). FPS BAD.\	
(1)	064347	200	041050	052125	.ASCIIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 272 INSTEAD OF 072.\	
(0)	064426				EM355:		
(1)	064426	052123	054105	020120	.ASCII	\STEXP A,(R). FPS BAD.\	
(1)	064453	200	041050	052125	.ASCIIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 471 INSTEAD OF 071.\	
	064532	052123	052123	024040	EM356:	.ASCII	'STST (R) GOT BAD FEC.\<CRLF>
	064560	043101	042524	020122	.ASCIIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'	
	064630	052123	052123	024040	EM357:	.ASCII	'STST (R) GOT BAD FEA.\<CRLF>
	064656	043101	042524	020122	.ASCIIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'	
	064726	047117	054514	047440	EM360:	.ASCII	'ONLY ONE WORD WRITTEN BY STS* (R).'

F15

MAINDEC-11-FPP34-A POP :1 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 197
 DFFPCA.P11 31-OCT-76 17:16 FLAG RESET AND CONSOLE TEST ROUTINE

	064771	123	052105	043040		.ASCIZ	'SET FDFL ST 636 FAILED.'
(0)	065021				EM401:		
(1)	065021	123	043124	051520		.ASCIZ	\STFPS (R). RO BAD.\
(0)	065044				EM402:		
(1)	065044	052123	050106	020123		.ASCIZ	\STFPS (R) FAILED.\
(0)	065066	047515	042522	052040	EM403:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R).'
(1)	065066	024200	052502	020124		.ASCIZ	<CR LF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.
(0)	065222				EM404:		
(1)	065222	052123	050106	020123		.ASCIZ	\STFPS (R) TRAPPED TO 4.
(0)	065250				EM405:		
(1)	065250	052123	050106	020123		.ASCIZ	\STFPS (R)+. RO BAD.\
(0)	065274				EM406:		
(1)	065274	052123	050106	020123		.ASCIZ	\STFPS (R)+ FAILED.\
(0)	065317	115	051117	020105	EM407:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R)+.'
(1)	065370	024200	052502	020124		.ASCIZ	<CR LF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	065452				EM410:		
(1)	065452	052123	050106	020123		.ASCIZ	\STFPS (R)+ TRAPPED TO 4.\
(0)	065503				EM411:		
(1)	065503	123	043124	051520		.ASCIZ	\STFPS -(R). RO BAD.\
(0)	065527				EM412:		
(1)	065527	123	043124	051520		.ASCIZ	\STFPS -(R) FAILED.\
(0)	065552	047515	042522	052040	EM413:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS -(R).'
(1)	065623	200	041050	052125		.ASCIZ	<CR LF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	065705				EM414:		
(1)	065705	123	043124	051520		.ASCIZ	\STFPS -(R) TRAPPED TO 4.\
(0)	065736				EM415:		
(1)	065736	052123	050106	020123		.ASCIZ	\STFPS 2(R)+. RO BAD.\
(0)	065763				EM416:		
(1)	065763	123	043124	051520		.ASCIZ	\STFPS 2(R)+ FAILED.\
(0)	066007	123	043124	051520	EM417:	.ASCIZ	'STFPS 2(R)+ DID NOT DEFER THE WRITE.'
(0)	066055				EM420:		
(1)	066055	123	043124	051520		.ASCIZ	\STFPS 2(R)+ TRAPPED TO 4.\
(0)	066107				EM421:		
(1)	066107	123	043124	051520		.ASCIZ	\STFPS 2-(R). RO BAD.\
(0)	066134				EM422:		
(1)	066134	052123	050106	020123		.ASCIZ	\STFPS 2-(R) FAILED.\
(0)	066160	052123	050106	020123	EM423:	.ASCIZ	'STFPS 2-(R) DID NOT DEFER THE WRITE.'
(0)	066226				EM424:		
(1)	066226	052123	050106	020123		.ASCIZ	\STFPS 2-(R) TRAPPED TO 4.\
(0)	066260				EM425:		
(1)	066260	052123	050106	020123		.ASCIZ	\STFPS N(R). RO BAD.\
(0)	066304				EM426:		
(1)	066304	052123	050106	020123		.ASCIZ	\STFPS N(R) FAILED.\
(0)	066327	115	051117	020105	EM427:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS N(R).'
(1)	066400	024200	052502	020124		.ASCIZ	<CR LF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	066462				EM430:		
(1)	066462	052123	050106	020123		.ASCIZ	\STFPS N(R) TRAPPED TO 4.\
(0)	066513	120	020103	040502	EM431:	.ASCII	'PC BAD AFTER STFPS N(R). BAD CONSTANT USED.'

G15

(0)	066566				EM432:		
(1)	066566	052123	050106	020123	.ASCIZ	\STFPS 2N(R). RO BAD.\	
(0)	066613				EM433:		
(1)	066613	123	043124	051520	.ASCIZ	\STFPS 2N(R) FAILED.\	
(0)	066637	115	051117	020105	EM434:	.ASCII	*MORE THAN ONE WORD WRITTEN BY STFPS 2N(R).*
(1)	066711	200	041050	052123	.ASCIZ	<CALF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.	
(0)	066773				EM435:		
(1)	066773	123	043124	051520	.ASCIZ	\STFPS 2N(R) TRAPPED TO 4.\	
	067025	120	020103	040502	EM436:	.ASCIZ	*PC BAD AFTER STFPS 2N(R). BAD CONSTANT USED.*
(0)	067102				EM437:		
(1)	067102	052123	042103	020114	.ASCIZ	\STCOL A,(R)+. RO BAD.\	
(0)	067130				EM440:		
(1)	067130	052123	042103	020114	.ASCIZ	\STCOL A,-(R). RO BAD.\	
	067156	052123	052123	024040	EM361:	.ASCIZ	*STST (R). FPS BAD.*

000000	EM362=0
000000	EM363=0
000000	EM364=0
000000	EM365=0
000000	EM366=0
000000	EM367=0
000000	EM370=0
000000	EM371=0
000000	EM372=0
000000	EM373=0
000000	EM374=0
000000	EM375=0
000000	EM376=0
000000	EM377=0
000000	EM400=0

067201	125	042516	050130	EM441:	.ASCIZ	*UNEXPECTED FPP TRAP TO 244.*
067235	125	042516	050130	EM442:	.ASCIZ	*UNEXPECTED CPU TRAP TO 4.*
067267	125	042516	050130	EM443:	.ASCIZ	*UNEXPECTED CPU TRAP TO 10.*

:THESE ARE DATA TABLE HEADERS:

(0)	067322	020040	042524	052123	DH1:	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*
	067362	043011	051520	004456		.ASCIZ	<TAB>*FPS.*<TAB>*FEC.*
(1)	067375	040	052040	051505	DH2:	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*
(0)	067435	011	047507	020124		.ASCIZ	<TAB>*GOT FPS.*<TAB>*EXPECTED FPS.*
(1)	067465	040	052040	051505	DH3:	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*
	067525	011	047507	020124		.ASCIZ	<TAB>*GOT FEC.*<TAB>*EXPECTED FEC.*
(0)	067555	040	052040	051505	DH4:	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*
(1)	067615	011	047507	020124		.ASCIZ	<TAB>*GOT RO.*<TAB>*EXPECTED RO.*
(0)	067644				DH5:		
(1)	067644	020040	042524	052123	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*	
	067704	000			.BYTE	0	
		067644			DH6=DH5		
		067555			DH7=DH4		

H15

MAINDEC-11-FPP34-A POP 11 34 FPP DIAGNOSTIC MACY11 27(1006) 31-OCT-76 17:35 PAGE 199
DFFPCA.P11 31-OCT-76 17:16 FLAG RESET AND CONSOLE TEST ROUTINE

	067644			DH10=DH5	
	067555			DH11=DH4	
	067644			DH12=DH5	
067705	040	052040	051505	DH13: .ASCIZ	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF TRAP.*
	067705			DH14=DH13	
	067644			DH15=DH5	
(0)	067745			DH16:	
(1)	067745	052040	051505	.ASCII	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*
	070005	047507	020124	.ASCIZ	<TAB>*GOT PC.*<TAB>*EXPECTED PC.*
	067705			DH17=DH13	
	067555			DH20=DH4	
	067644			DH21=DH5	
	067644			DH22=DH5	
	067705			DH23=DH13	
	067555			DH24=DH4	
	067644			DH25=DH5	
	067705			DH26=DH13	
	067555			DH27=DH4	
	067644			DH30=DH5	
	067705			DH31=DH13	
	067555			DH32=DH4	
	067644			DH33=DH5	
	067705			DH34=DH13	
	067555			DH35=DH4	
	067644			DH36=DH5	
070034	020040	042524	052123	DH37: .ASCIZ	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*<TAB>*GOT FPS.*<TAB>*EXPEC
070124	020040	042524	052123	DH40=DH37	
	070034			DH41: .ASCIZ	* TEST.*<TAB>*PC OF CALL.*<TAB>*PC OF ERROR.*<TAB>*FPS.*<TAB>*GOT FEC.
	070034			DH42=DH37	
	070034			DH43=DH37	
	070034			DH44=DH37	
	070034			DH45=DH37	
	070034			DH46=DH37	
	070034			DH47=DH37	
	070034			DH50=DH37	
	070034			DH51=DH37	
	070034			DH52=DH37	
	070124			DH53=DH41	
	070034			DH54=DH37	
	070034			DH55=DH37	
	070034			DH56=DH37	
	070034			DH57=DH37	
	070034			DH60=DH37	
	070034			DH61=DH37	
	067375			DH62=DH2	
	067465			DH63=DH3	
	067644			DH64=DH5	
	067375			DH65=DH2	
	067555			DH66=DH4	
	067375			DH67=DH2	
	067465			DH70=DH3	
	067375			DH176=DH2	
	067465			DH177=DH3	
	067644			DH71=DH5	
	067375			DH72=DH2	
	067705			DH107=DH13	

067745	DH164=DH16
067745	DH215=DH16
067644	DH216=DH45
067555	DH217=DH44
067375	DH220=DH2
067745	DH221=DH16
067644	DH222=DH5
067555	DH223=DH4
067375	DH224=DH2
070034	DH165=DH37
070034	DH166=DH37
070034	DH167=DH37
070034	DH170=DH37
070034	DH171=DH37
070034	DH172=DH37
070124	DH173=DH41
070124	DH174=DH41
070124	DH175=DH41
070034	DH200=DH37
070034	DH201=DH37
070034	DH202=DH37
070034	DH203=DH37
070034	DH204=DH37
070034	DH205=DH37
070034	DH206=DH37
070034	DH207=DH37
070034	DH210=DH37
070034	DH211=DH37
070034	DH212=DH37
070034	DH213=DH37
070034	DH214=DH37

070221	067555	052040	051505
	067375		
	067555		
	067375		
	070221		
	067555		
	067375		
	070221		
	067555		
	067375		
	070221		
	067555		
	067375		
	067745		
	070221		
	067555		
	067375		
	067745		
	070221		
	067745		

DH225=DH4
DH226=DH2
DH227: .ASCIZ ' TEST.' <TAB> 'PC OF CALL.' <TAB> 'PC OF TRAP.'
DH230=DH4
DH231=DH2
DH232=DH227
DH233=DH4
DH234=DH2
DH235=DH227
DH236=DH4
DH237=DH2
DH240=DH227
DH241=DH4
DH242=DH2
DH243=DH227
DH244=DH4
DH245=DH2
DH246=DH16
DH247=DH227
DH250=DH4
DH251=DH2
DH252=DH16
DH253=DH227
DH254=DH16

K15

MAINDEC-11-FPP34-A
OFFPCA.P11 31-OCT-76

PDP 11 34 FPP DIAGNOSTIC
17:16

MACY11 27(1006) 31-OCT-76 17:35 PAGE 192
FLAG RESET AND CONSOLE TEST ROUTINE

070221	DH255=DH227
067555	DH256=DH4
067375	DH257=DH2
070034	DH260=DH37
070034	DH261=DH37
070034	DH262=DH37
070034	DH263=DH37
070034	DH264=DH37
070034	DH265=DH37
070034	DH266=DH37
070034	DH267=DH37
070034	DH270=DH37
070034	DH271=DH37
070034	DH272=DH37
070034	DH273=DH37
070034	DH274=DH37
070034	DH275=DH37
070034	DH276=DH37
070034	DH277=DH37
070034	DH300=DH37
070034	DH301=DH37
070034	DH302=DH37
070124	DH303=DH41
070034	DH304=DH37
070034	DH305=DH37
070034	DH306=DH37
070034	DH307=DH37
070034	DH310=DH37
070034	DH311=DH37
070034	DH312=DH37
070034	DH313=DH37
070034	DH314=DH37
070034	DH315=DH37
070034	DH316=DH37
070034	DH317=DH37
070034	DH320=DH37
070034	DH321=DH37
070034	DH322=DH37
070034	DH323=DH37
070124	DH324=DH41
070034	DH325=DH37
070034	DH326=DH37
070034	DH327=DH37
070034	DH330=DH37
070034	DH331=DH37
070034	DH332=DH37
070034	DH333=DH37
070034	DH334=DH37
070034	DH335=DH37
070034	DH336=DH37
070034	DH337=DH37
070034	DH340=DH37
070034	DH341=DH37
070034	DH342=DH37
070034	DH343=DH37
070034	DH344=DH37

070034				DH345=DH37	
070034				DH346=DH37	
070034				DH347=DH37	
070034				DH350=DH37	
067705				DH351=DH13	
070034				DH352=DH37	
070034				DH353=DH37	
070034				DH354=DH37	
070034				DH355=DH37	
067555				DH356=DH11	
070261	040	052040	051505	DH357: .ASCII	' TEST.' <TAB> 'PC OF CALL.' <TAB> 'PC OF ERROR.'
070321	011	047507	020124	.ASCIZ	<TAB> 'GOT FEA.' <TAB> 'EXPECTED FEA.'
067705				DH360=DH13	
067375				DH361=DH2	
000000				DH362=0	
000000				DH363=0	
000000				DH364=0	
000000				DH365=0	
000000				DH366=0	
000000				DH367=0	
000000				DH370=0	
000000				DH371=0	
000000				DH372=0	
000000				DH373=0	
000000				DH374=0	
000000				DH375=0	
000000				DH376=0	
000000				DH377=0	
000000				DH400=0	
067555				DH401=DH4	
067375				DH402=DH2	
067705				DH403=DH13	
070221				DH404=DH227	
067555				DH405=DH4	
067375				DH406=DH2	
067705				DH407=DH13	
070221				DH410=DH227	
067555				DH411=DH4	
067375				DH412=DH2	
067705				DH413=DH13	
070221				DH414=DH227	
067555				DH415=DH4	
067375				DH416=DH2	
067705				DH417=DH13	
070221				DH420=DH227	
067555				DH421=DH4	
067375				DH422=DH2	
067705				DH423=DH13	
070221				DH424=DH227	
067555				DH425=DH4	
067375				DH426=DH2	
067705				DH427=DH13	
070221				DH430=DH227	
067705				DH431=DH13	

M15

	067555			DH432=DH4	
	067375			DH433=DH2	
	067705			DH434=DH13	
	070221			DH435=DH227	
	067705			DH436=DH13	
	067555			DH437=DH4	
	067555			DH440=DH4	
070351	040	052040	051505	DH441: .ASCIZ	: TEST.<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FEC.'
070417	040	052040	05:505	DH442: .ASCIZ	: TEST.<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
	070417			DH443=DH442	

: THESE ARE FORMAT SPECIFICATIONS FOR THE DATA TABLES: :

070460	004	000	005	DF1: .BYTE	4,0,5,0,5,0,0
070467	004	000	005	DF2: .BYTE	4,0,5,0,5,0,5,0
	070467			DF3=DF2	
	070467			DF4=DF2	
070477	004	000	005	DF5: .BYTE	4,0,5,0,5,5,2,5,5,2
070511	004	000	005	DF6: .BYTE	4,0,5,0
	070467			DF7=DF4	
	070477			DF10=DF5	
	070467			DF11=DF4	
070515	004	000	005	DF12: .BYTE	4,0,5,0,5,5,3,5,5,3
	070511			DF13=DF6	
	070511			DF14=DF6	
	070515			DF15=DF12	
	070467			DF16=DF2	
	070511			DF17=DF6	
	070467			DF20=DF2	
	070515			DF21=DF12	
	070515			DF22=DF12	
	070511			DF23=DF6	
	070467			DF24=DF2	
	070515			DF25=DF12	
	070511			DF26=DF6	
	070467			DF27=DF2	
	070515			DF30=DF12	
	070511			DF31=DF6	
	070467			DF32=DF2	
	070515			DF33=DF12	
	070511			DF34=DF6	
	070467			DF35=DF2	
	070515			DF36=DF12	
070527	004	000	005	DF37: .BYTE	4,0,5,0,5,0,5,0,5,5,3,5,5,3,5,5,3
	070527			DF40=DF37	
070550	004	000	005	DF41: .BYTE	4,0,5,0,5,0,0,0,5,5,3,5,5,3,5,5,3
	070527			DF42=DF37	
	070527			DF43=DF37	
	070527			DF44=DF37	
	070527			DF45=DF37	
	070527			DF46=DF37	
	070527			DF47=DF37	
	070527			DF50=DF37	
	070527			DF51=DF37	
	070527			DF52=DF37	

N15

MAINDEC-11-FPP34-A
DFFPCA.P11

31-OCT-76 17:16

PDP 11 34 FPP DIAGNOSTIC

FLAG RESET AND CONSOLE TEST ROUTINE

MACY11 27(1006) 31-OCT-76 17:35 PAGE 195

070527			DF53=DF37	
070527			DF54=DF37	
070527			DF55=DF37	
070527			DF56=DF37	
070527			DF57=DF37	
070527			DF60=DF37	
070527			DF61=DF37	
070467			DF62=DF2	
070467			DF63=DF2	
070477			DF64=DF5	
070467			DF65=DF2	
070467			DF66=DF2	
070467			DF67=DF2	
070467			DF70=DF2	
070467			DF176=DF2	
070467			DF177=DF2	
070571	004	000	005	DF71: .BYTE 4,0,5,0,5,5,3,5,5,3,5,5,3
070467				DF72=DF2
070511				DF107=DF6
070571				DF73=DF71
070467				DF74=DF2
070467				DF75=DF2
070511				DF76=DF6
070571				DF77=DF71
070467				DF100=DF2
070467				DF101=DF2
070511				DF102=DF6
070571				DF103=DF71
070467				DF104=DF2
070467				DF105=DF2
070511				DF106=DF6
070571				DF110=DF71
070467				DF111=DF2
070467				DF112=DF2
070511				DF113=DF6
070571				DF114=DF71
070467				DF115=DF2
070467				DF116=DF2
070511				DF117=DF6
070571				DF120=DF71
070467				DF121=DF2
070467				DF122=DF2
070511				DF123=DF6
070571				DF124=DF71
070467				DF125=DF2
070467				DF126=DF2
070511				DF127=DF6
070571				DF130=DF71
070467				DF131=DF2
070511				DF132=DF6
070571				DF133=DF71
070467				DF134=DF2
070515				DF135=DF12
070515				DF136=DF12
070467				DF137=DF2
070515				DF140=DF12

C16

070620
070621
070622
070623
070624
070625
070626
070627
070628
070629
070630
070631
070632
070633
070634
070635
070636
070637
070638
070639
070640
070641
070642
070643
070644
070645
070646
070647
070648
070649
070650

DF233=DF233
DF234=DF234
DF235=DF235
DF236=DF236
DF237=DF237
DF238=DF238
DF239=DF239
DF240=DF240
DF241=DF241
DF242=DF242
DF243=DF243
DF244=DF244
DF245=DF245
DF246=DF246
DF247=DF247
DF248=DF248
DF249=DF249
DF250=DF250
DF251=DF251
DF252=DF252
DF253=DF253
DF254=DF254
DF255=DF255
DF256=DF256
DF257=DF257
DF258=DF258
DF259=DF259
DF260=DF260

070622 004 000 005 DF260: .BYTE 4.0.5.0.5.0.5.0.5.5.2.5.5.2.5.5.2
070622 DF261=DF260
070622 DF262=DF260
070622 DF263=DF260
070622 DF264=DF260
070622 DF265=DF260
070622 DF266=DF260
070622 DF267=DF260
070622 DF270=DF260
070622 DF271=DF260
070622 DF272=DF260

070643 004 000 005 DF273: .BYTE 4.0.5.0.5.0.5.0.5.5.2.5.5.3.5.5.3
070643 DF274=DF273
070643 DF275=DF273
070643 DF276=DF273
070643 DF277=DF273
070643 DF300=DF273

070664 004 000 005 DF301: .BYTE 4.0.5.0.5.0.5.0.5.5.3.5.5.0.5.5.3.5.5.3
070664 DF302=DF301

070710 004 000 005 DF303: .BYTE 4.0.5.0.5.0.0.0.5.5.3.5.5.0.5.5.3.5.5.3
070664 DF304=DF301
070664 DF305=DF301
070664 DF306=DF301
070664 DF307=DF301
070664 DF310=DF301
070664 DF311=DF301
070664 DF312=DF301
070664 DF313=DF301
070664 DF314=DF301
070664 DF315=DF301
070664 DF316=DF301
070664 DF317=DF301
070664 DF320=DF301

```

070664 DF321=DF301
070734 004 000 005 DF322: .BYTE 4.0.5.0.5.0.5.0.5.5.3.5.5.2.5.5.2
070755 004 000 005 DF323=DF322
DF324: .BYTE 4.0.5.0.5.0.0.0.5.5.3.5.5.2.5.5.2
DF325=DF322
DF326=DF322
DF327=DF322
DF330=DF322
DF331=DF322
DF332=DF322
DF333=DF322
DF334=DF322
DF335=DF322
DF336=DF322
DF337=DF322
DF340=DF322
DF341=DF322
DF342=DF322
DF343=DF322
DF344=DF322
DF345=DF322
DF346=DF322
070776 004 000 005 DF347: .BYTE 4.0.5.0.5.0.5.0.5.5.3.5.5.0.5.5.0
DF350=DF347
DF351=DF227
DF352=DF347
DF353=DF347
DF354=DF347
DF355=DF347
DF356=DF225
DF357=DF225
DF360=DF227
DF361=DF225
000000 DF362=0
000000 DF363=0
000000 DF364=0
000000 DF365=0
000300 DF366=0
000000 DF367=0
000000 DF370=0
000000 DF371=0
000000 DF372=0
000000 DF373=0
000000 DF374=0
000000 DF375=0
000000 DF376=0
000000 DF377=0
000000 DF400=0
070606 DF401=DF225
070606 DF402=DF225
070616 DF403=DF227
070616 DF404=DF227

```

```

071017 004 000 005 OF441: .BYTE 4,0,5,0,5,0
071017 071017 OF442=OF441
071017 071017 OF443=OF441

```

071026 .EVEN

;THESE ARE THE ERROR MESSAGE DATA TABLES:

```

071026 001232 001234 043005 DT1: .WORD STMP0,STMP1,STAB,STMP2,STAB,STMP3,STMP4,0
071046 001232 001234 043005 DT2: .WORD STMP0,STMP1,STAB,STMP2,STAB,STMP3,STAB,STMP5,0
071070 001232 001234 043005 DT3: .WORD STMP0,STMP1,STAB,STMP2,STAB,STMP4,STAB,STMP6,0
071112 001232 001234 043005 DT4: .WORD STMP0,STMP1,STAB,STMP2,STAB,STMP4,STAB,STMP3,0
071134 001232 001234 043005 DT5: .WORD STMP0,STMP1,STAB,STMP2,SCRLF,MS1,STMP3
071152 001313 043025 001242 DT6: .WORD SCRLF,MS2,STMP4,0
071162 001232 001234 043005 DT7: .WORD STMP0,STMP1,STAB,STMP2,0
071112 DT7=DT4
071134 DT10=DT5
071112 DT11=DT4
071134 DT12=DT5
071162 DT13=DT6
071162 DT14=DT6
071134 DT15=DT5
071174 001232 001234 043005 DT16: .WORD STMP0,STMP1,STAB,STMP2,STAB,STMP4,STAB,STMP3,0
071162 DT17=DT6
071174 C*20=DT16
071134 DT21=DT5
071134 DT22=DT5

```

F16

MAINDEC-11-F0034-A
OFFPCA.P11 31-OCT-76

DDP 11 24 FPP DIAGNOSTIC
17:16

MACY11 27(1006) 31-OCT-76 17:35 PAGE 200
FLAG RESET AND CONSOLE TEST ROUTINE

071216	001234	043005	DT23=DT6		
071240	001240	001313	DT24=DT16		
			DT25=DT5		
			DT26=DT6		
			DT27=DT16		
			DT30=DT5		
			DT31=DT6		
			DT32=DT16		
			DT33=DT5		
			DT34=DT6		
			DT35=DT16		
			DT36=DT5		
			DT37:	.WORD	STMP0, STMP1, STAB, STMP2, STAB, STMP7, STAB, STMP10, SCRLF
				.WORD	MS4, STMP3, SCRLF, MS1, STMP4, SCRLF, MS2, STMP5, 0
			DT40=DT37		
			DT41:	.WORD	STMP0, STMP1, STAB, STMP2, STAB, STMP7, STMP11, STMP12
				.WORD	SCRLF, MS4, STMP3, SCRLF, MS1, STMP4, SCRLF, MS2, STMP5, 0
			DT42=DT37		
			DT43=DT37		
			DT44=DT37		
			DT45=DT37		
			DT46=DT37		
			DT47=DT37		
			DT50=DT37		
			DT51=DT37		
			DT52=DT37		
			DT53=DT41		
			DT54=DT37		
			DT55=DT37		
			DT56=DT37		
			DT57=DT37		
			DT60=DT37		
			DT61=DT37		
			DT62=DT16		
			DT63=DT16		
			DT64=DT5		
			DT65=DT16		
			DT66=DT4		
			DT67=DT4		
			DT70=DT4		
			DT176=DT4		
			DT177=DT4		
			DT71:	.WORD	STMP0, STMP1, STAB, STMP2, SCRLF, MS3, STMP3, SCRLF, MS1
				.WORD	STMP5, SCRLF, MS2, STMP4, 0
			DT72=DT4		
			DT107=DT6		
			DT73:	.WORD	STMP0, STMP1, STAB, STMP2, SCRLF, MS4, STMP4
				.WORD	SCRLF, MS1, STMP5, SCRLF, MS2, STMP3, 0
			DT74=DT4		
			DT75=DT2		
			DT76=DT6		
			DT77=DT73		
			DT100=DT4		
			DT101=DT2		
			DT102=DT6		
			DT103=DT73		

G16

MAINDEC-11-FPP34-A
DFFPCA.P11 31-OCT-76 17:16

POP 1: 34 FPP DIAGNOSTIC
FLAG RESET AND CONSOLE TEST ROUTINE

MACY11 27(1006) 31-OCT-76 17:35 PAGE 201

071112	DT104=OT4
071046	DT105=OT2
071162	DT106=OT6
071362	DT110=OT73
071112	DT111=OT4
071046	DT112=OT2
071162	DT113=OT6
071362	DT114=OT73
071112	DT115=OT4
071046	DT116=OT2
071162	DT117=OT6
071362	DT120=OT73
071112	DT121=OT4
071046	DT122=OT2
071162	DT123=OT6
071362	DT124=OT73
071112	DT125=OT4
071046	DT126=OT2
071162	DT127=OT6
071362	DT130=OT73
071046	DT131=OT2
071162	DT132=OT6
071362	DT133=OT73
071046	DT134=OT2
071134	DT135=OT5
071134	DT136=OT5
071174	DT137=OT16
071134	DT140=OT5
071112	DT141=OT4
071112	DT142=OT4
071134	DT143=OT5
071112	DT144=OT4
071112	DT145=OT4
071134	DT146=OT5
071112	DT147=OT4
071112	DT150=OT4
071134	DT151=OT5
071112	DT152=OT4
071112	DT153=OT4
071134	DT154=OT5
071112	DT155=OT4
071112	DT156=OT4
071134	DT157=OT5
071112	DT160=OT4
071112	DT161=OT4
071134	DT162=OT5
071112	DT163=OT4
071112	DT164=OT4
071112	DT215=OT4
071134	DT216=OT5
071112	DT217=OT4
071112	DT220=OT4
071112	DT221=OT4
071134	DT222=OT5
071112	DT223=OT4
071112	DT224=OT4

H16

MAINDEC-11-FPP34-A
OFFPCA.P11

POP 1: 34 FPP DIAGNOSTIC
17:16

MACY11 27(1006) 31-OCT-76
FLAG RESET AND CONSOLE TEST ROUTINE

17:35 PAGE 202

071216	DT165=DT37
071216	DT166=DT37
071216	DT167=DT37
071216	DT170=DT37
071216	DT171=DT37
071216	DT172=DT37
071262	DT173=DT41
071262	DT174=DT41
071262	DT175=DT41
071216	DT200=DT37
071216	DT201=DT37
071216	DT202=DT37
071216	DT203=DT37
071216	DT204=DT37
071216	DT205=DT37
071216	DT206=DT37
071216	DT207=DT37
071216	DT210=DT37
071216	DT211=DT37
071216	DT212=DT37
071216	DT213=DT37
071216	DT214=DT37

071112	DT225=DT4
071112	DT226=DT4
071162	DT227=DT6
071112	DT230=DT4
071112	DT231=DT4
071162	DT232=DT6
071112	DT233=DT4
071112	DT234=DT4
071162	DT235=DT6
071112	DT236=DT4
071112	DT237=DT4
071162	DT240=DT6
071112	DT241=DT4
071112	DT242=DT4
071162	DT243=DT6
071112	DT244=DT4
071112	DT245=DT4
071112	DT246=DT4
071162	DT247=DT6
071112	DT250=DT4
071112	DT251=DT4
071112	DT252=DT4
071162	DT253=DT6
071112	DT254=DT4
071162	DT255=DT6
071112	DT256=DT4
071112	DT257=DT4

071216	DT260=DT37
071216	DT261=DT37
071216	DT262=DT37
071216	DT263=DT37
071216	DT264=DT37

071216				DT265=DT37	
071216				DT266=DT37	
071216				DT267=DT37	
071216				DT270=DT37	
071216				DT271=DT37	
071216				DT272=DT37	
071216				DT273=DT37	
071216				DT274=DT37	
071216				DT275=DT37	
071216				DT276=DT37	
071216				DT277=DT37	
071216				DT300=DT37	
071416	001232	001234	043005	DT301: .WORD	STMP0,STMP1,STAB,STMP2,STAB,STMP7,STAB,STMP10
071436	001313	043047	001240	.WORD	\$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4
071452	001313	043007	001246	.WORD	\$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0
071470	001232	001234	043005	DT302=DT301	
071510	001313	043047	001240	DT303: .WORD	STMP0,STMP1,STAB,STMP2,STAB,STMP7,STMP11,STMP12
071524	001313	043007	001246	.WORD	\$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4
				.WORD	\$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0
071416				DT304=DT301	
071416				DT305=DT301	
071416				DT306=DT301	
071416				DT307=DT301	
071416				DT310=DT301	
071416				DT311=DT301	
071416				DT312=DT301	
071416				DT313=DT301	
071416				DT314=DT301	
071416				DT315=DT301	
071416				DT316=DT301	
071416				DT317=DT301	
071416				DT320=DT301	
071416				DT321=DT301	
071542	001232	001234	043005	DT322: .WORD	STMP0,STMP1,STAB,STMP2,STAB,STMP7,STAB,STMP10
071562	001313	043047	001240	.WORD	\$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0
	071542			DT323=DT322	
071606	001232	001234	043005	DT324: .WORD	STMP0,STMP1,STAB,STMP2,STAB,STMP7,STMP11,STMP12
071626	001313	043047	001240	.WORD	\$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0
	071542			DT325=DT322	
	071542			DT326=DT322	
	071542			DT327=DT322	
	071542			DT330=DT322	
	071542			DT331=DT322	
	071542			DT332=DT322	
	071542			DT333=DT322	
	071542			DT334=DT322	
	071542			DT335=DT322	
	071542			DT336=DT322	
	071542			DT337=DT322	
	071542			DT340=DT322	
	071542			DT341=DT322	
	071542			DT342=DT322	
	071542			DT343=DT322	
	071542			DT344=DT322	

071542
071542

-DT345=DT322
DT346=DT322

071542
071542
071162
071542
071542
071542
071542
071046
071070
071162
071416

DT347=DT322
DT350=DT322
DT351=DT6
DT352=DT322
DT353=DT322
DT354=DT322
DT355=DT322
DT356=DT2
DT357=DT3
DT360=DT6
DT361=DT302

000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000
000000

DT362=0
DT363=0
DT364=0
DT365=0
DT366=0
DT367=0
DT370=0
DT371=0
DT372=0
DT373=0
DT374=0
DT375=0
DT376=0
DT377=0
DT400=0

071112
071112
071162
071162
071112
071112
071162
071162
071112
071112
071162
071162
071112
071112
071162
071162
071112
071112
071162
071162
071112
071112
071162
071162
071112
071112
071162
071162
071112
071112

DT401=DT4
DT402=DT4
DT403=DT6
DT404=DT6
DT405=DT4
DT406=DT4
DT407=DT6
DT410=DT6
DT411=DT4
DT412=DT4
DT413=DT6
DT414=DT6
DT415=DT4
DT416=DT4
DT417=DT6
DT420=DT6
DT421=DT4
DT422=DT4
DT423=DT6
DT424=DT6
DT425=DT4
DT426=DT4
DT427=DT6
DT430=DT6
DT431=DT6

K16

MAINDEC-11-FPP34-A
DFFPCA.P11

31-OCT-76

PDP 11-34 FPP DIAGNOSTIC
17:16

FLAG RESET AND CONSOLE TEST ROUTINE

MACY11 27(1006) 31-OCT-76 17:35 PAGE 205

	071112			DT432=DT4	
	071113			DT433=DT4	
	071162			DT434=DT6	
	071163			DT435=DT6	
	071164			DT436=DT6	
	071112			DT437=DT4	
	071113			DT440=DT4	
071652	001232	001234	043005	DT441: .WORD	STMP0,STMP1,STAB,STMP2,STAB,STMP3,0
071670	001232	001234	043005	DT442: .WORD	STMP0,STMP1,STAB,STMP2,0
	071670			DT443=DT442	

000001

;12345

.END

M16

MAINDEC-11-FPP34-A
DFFPCA.P11

PDP 11 34 FPP DIAGNOSTIC
31-OCT-76 17:16

MACY11 27(1006) 31-OCT-76 17:35 PAGE 208

SYMBOL TABLE

DF26 = 070511	DF336 = 070734	DF414 = 070616	DF74 = 070467	DH152 = 067555
DF260 = 070622	DF337 = 070734	DF415 = 070606	DF75 = 070467	DH153 = 067375
DF261 = 070622	DF34 = 070511	DF416 = 070606	DF76 = 070511	DH154 = 067644
DF262 = 070622	DF340 = 070734	DF417 = 070616	DF77 = 070571	DH155 = 067555
DF263 = 070622	DF341 = 070734	DF42 = 070527	DH1 = 067322	DH156 = 067375
DF264 = 070622	DF342 = 070734	DF420 = 070616	DH10 = 067644	DH157 = 067644
DF265 = 070622	DF343 = 070734	DF421 = 070606	DH100 = 067555	DH16 = 067745
DF266 = 070622	DF344 = 070734	DF422 = 070606	DH101 = 067375	DH160 = 067555
DF267 = 070622	DF345 = 070734	DF423 = 070616	DH102 = 067705	DH161 = 067375
DF27 = 070467	DF346 = 070734	DF424 = 070616	DH103 = 067644	DH162 = 067644
DF270 = 070622	DF347 = 070776	DF425 = 070606	DH104 = 067555	DH163 = 067375
DF271 = 070622	DF35 = 070467	DF426 = 070606	DH105 = 067375	DH164 = 067745
DF272 = 070622	DF350 = 070776	DF427 = 070616	DH106 = 067705	DH165 = 070034
DF273 = 070643	DF351 = 070616	DF43 = 070527	DH107 = 067705	DH166 = 070034
DF274 = 070643	DF352 = 070776	DF430 = 070616	DH11 = 067555	DH167 = 070034
DF275 = 070643	DF353 = 070776	DF431 = 070616	DH110 = 067644	DH170 = 070034
DF276 = 070643	DF354 = 070776	DF432 = 070606	DH111 = 067555	DH171 = 070034
DF277 = 070643	DF355 = 070776	DF433 = 070606	DH112 = 067375	DH172 = 070034
DF3 = 070467	DF356 = 070606	DF434 = 070616	DH113 = 067705	DH173 = 070124
DF30 = 070515	DF357 = 070606	DF435 = 070616	DH114 = 067644	DH174 = 070124
DF300 = 070643	DF36 = 070515	DF436 = 070616	DH115 = 067555	DH175 = 070124
DF301 = 070664	DF360 = 070616	DF437 = 070606	DH116 = 067375	DH176 = 067375
DF302 = 070664	DF361 = 070606	DF44 = 070527	DH117 = 067705	DH177 = 067465
DF303 = 070710	DF362 = 000000	DF440 = 070606	DH12 = 067644	DH2 = 067375
DF304 = 070664	DF363 = 000000	DF441 = 071017	DH120 = 067644	DH20 = 067555
DF305 = 070664	DF364 = 000000	DF442 = 071017	DH121 = 067555	DH200 = 070034
DF306 = 070664	DF365 = 000000	DF443 = 071017	DH122 = 067375	DH201 = 070034
DF307 = 070664	DF366 = 000000	DF45 = 070527	DH123 = 067705	DH202 = 070034
DF31 = 070511	DF367 = 000000	DF46 = 070527	DH124 = 067644	DH203 = 070034
DF310 = 070664	DF37 = 070527	DF47 = 070527	DH125 = 067555	DH204 = 070034
DF311 = 070664	DF370 = 000000	DF5 = 070477	DH126 = 067375	DH205 = 070034
DF312 = 070664	DF371 = 000000	DF50 = 070527	DH127 = 067705	DH206 = 070034
DF313 = 070664	DF372 = 000000	DF51 = 070527	DH13 = 067705	DH207 = 070034
DF314 = 070664	DF373 = 000000	DF52 = 070527	DH130 = 067644	DH21 = 067644
DF315 = 070664	DF374 = 000000	DF53 = 070527	DH131 = 067375	DH210 = 070034
DF316 = 070664	DF375 = 000000	DF54 = 070527	DH132 = 067705	DH211 = 070034
DF317 = 070664	DF376 = 000000	DF55 = 070527	DH133 = 067644	DH212 = 070034
DF32 = 070467	DF377 = 000000	DF56 = 070527	DH134 = 067375	DH213 = 070034
DF320 = 070664	DF4 = 070467	DF57 = 070527	DH135 = 067644	DH214 = 070034
DF321 = 070664	DF40 = 070527	DF6 = 070511	DH136 = 067644	DH215 = 067745
DF322 = 070734	DF400 = 000000	DF60 = 070527	DH137 = 067375	DH216 = 067644
DF323 = 070734	DF401 = 070606	DF61 = 070527	DH14 = 067705	DH217 = 067555
DF324 = 070755	DF402 = 070606	DF62 = 070467	DH140 = 067644	DH22 = 067644
DF325 = 070734	DF403 = 070616	DF63 = 070467	DH141 = 067555	DH220 = 067375
DF326 = 070734	DF404 = 070616	DF64 = 070477	DH142 = 067375	DH221 = 067745
DF327 = 070734	DF405 = 070606	DF65 = 070467	DH143 = 067644	DH222 = 067644
DF33 = 070515	DF406 = 070606	DF66 = 070467	DH144 = 067555	DH223 = 067555
DF330 = 070734	DF407 = 070616	DF67 = 070467	DH145 = 067375	DH224 = 067375
DF331 = 070734	DF41 = 070550	DF7 = 070467	DH146 = 067644	DH225 = 067555
DF332 = 070734	DF410 = 070616	DF70 = 070467	DH147 = 067555	DH226 = 067375
DF333 = 070734	DF411 = 070606	DF71 = 070571	DH15 = 067644	DH227 = 070221
DF334 = 070734	DF412 = 070606	DF72 = 070467	DH150 = 067375	DH23 = 067705
DF335 = 070734	DF413 = 070616	DF73 = 070571	DH151 = 067644	

EM1205	053554
EM1206	053653
EM1207	053754
EM121	044020
EM1210	054053
EM1211	054152
EM1212	054260
EM1213	054361
EM1214	054506
EM1215	054551
EM1216	052202
EM1217	052224
EM122	044020
EM1220	052274
EM1221	052320
EM1222	052452
EM1223	052475
EM1224	052546
EM1225	054633
EM1226	054656
EM1227	054702
EM123	044043
EM1230	054732
EM1231	054756
EM1232	055002
EM1233	055034
EM1234	055060
EM1235	055105
EM1236	055136
EM1237	055163
EM124	044102
EM1240	055211
EM1241	055243
EM1242	055270
EM1243	055316
EM1244	055350
EM1245	055374
EM1246	055421
EM1247	055452
EM125	044144
EM1250	055503
EM1251	055530
EM1252	055556
EM1253	055610
EM1254	055642
EM1255	055676
EM1256	055732
EM1257	055760
EM126	044170
EM1260	056007
EM1261	056044
EM1262	056103
EM1263	056203

EM1264	056231
EM1265	056326
EM1266	056417
EM1267	056532
EM127	044227
EM1270	056627
EM1271	056670
EM1272	056736
EM1273	057027
EM1274	057064
EM1275	057123
EM1276	057223
EM1277	057320
EM13	043224
EM130	044271
EM1300	057374
EM1301	057471
EM1302	057515
EM1303	057543
EM1304	057571
EM1305	057660
EM1306	057763
EM1307	060150
EM131	044315
EM1310	060252
EM1311	060355
EM1312	060456
EM1313	060560
EM1314	060661
EM1315	060762
EM1316	061063
EM1317	061164
EM132	044353
EM1320	061265
EM1321	061366
EM1322	061467
EM1323	061524
EM1324	061563
EM1325	061622
EM1326	061622
EM1327	061763
EM133	044414
EM1330	062065
EM1331	062170
EM1332	063444
EM1333	061524
EM1334	062273
EM1335	062367
EM1336	062471
EM1337	062545
EM134	044437
EM1340	062647
EM1341	062751

EM1342	063055
EM1343	063157
EM1344	063261
EM1345	063336
EM1346	063636
EM1347	063734
EM135	044476
EM1350	063760
EM1351	064006
EM1352	064112
EM1353	064216
EM1354	064322
EM1355	064426
EM1356	064532
EM1357	064630
EM136	044540
EM1360	064726
EM1361	067156
EM1362	000000
EM1363	000000
EM1364	000000
EM1365	000000
EM1366	000000
EM1367	000000
EM137	044564
EM1370	000000
EM1371	000000
EM1372	000000
EM1373	000000
EM1374	000000
EM1375	000000
EM1376	000000
EM1377	000000
EM14	043257
EM140	044610
EM1400	000000
EM1401	065021
EM1402	065044
EM1403	065066
EM1404	055220
EM1405	065250
EM1406	065274
EM1407	065317
EM141	044636
EM1410	065452
EM1411	065503
EM1412	065527
EM1413	065552
EM1414	065705
EM1415	065736
EM1416	065763
EM1417	066007
EM142	044664

EM1420	066055
EM1421	066107
EM1422	066134
EM1423	066160
EM1424	066226
EM1425	066260
EM1426	066304
EM1427	066327
EM143	044743
EM1430	066462
EM1431	066513
EM1432	066566
EM1433	066613
EM1434	066637
EM1435	066773
EM1436	067025
EM1437	067102
EM144	045047
EM1440	067130
EM1441	067201
EM1442	067235
EM1443	067267
EM145	045147
EM146	045225
EM147	045331
EM148	043317
EM150	045431
EM151	045545
EM152	045571
EM153	045615
EM154	045647
EM155	045720
EM156	046046
EM157	046150
EM16	043341
EM160	046260
EM161	046370
EM162	046472
EM163	046576
EM164	046624
EM165	046700
EM166	046723
EM167	046762
EM17	043445
EM170	047063
EM171	047154
EM172	047173
EM173	047254
EM174	047275
EM175	047317
EM176	047342
EM177	047403
EM10	040150

EM178	044743
EM179	066513
EM180	066566
EM181	066613
EM182	066637
EM183	066773
EM184	067025
EM185	067102
EM186	045047
EM187	067130
EM188	067201
EM189	067235
EM190	067267
EM191	045147
EM192	045225
EM193	045331
EM194	043317
EM195	045431
EM196	045545
EM197	045571
EM198	045615
EM199	045647
EM200	045720
EM201	046046
EM202	046150
EM203	043341
EM204	046260
EM205	046370
EM206	046472
EM207	046576
EM208	046624
EM209	046700
EM210	046723
EM211	046762
EM212	043445
EM213	047063
EM214	047154
EM215	047173
EM216	047254
EM217	047275
EM218	047317
EM219	047342
EM220	047403
EM221	040150

ERRVECT= 030004
ERTYPE 022132
ERT1 022314
ERT2 022336
ERT3 022358
ERT4 022420
ERT5 022442
ERT6 022464
ERT7 022486
ERT8 022508
ERT9 022530
ERT10 022552
ERT11 022574
ERT12 022596
ERT13 022618
ERT14 022640
ERT15 022662
ERT16 022684
ERT17 022706
ERT18 022728
ERT19 022750
ERT20 022772
ERT21 022794
ERT22 022816
ERT23 022838
ERT24 022860
ERT25 022882
ERT26 022904
ERT27 022926
ERT28 022948
ERT29 022970
ERT30 022992
ERT31 023014
ERT32 023036
ERT33 023058
ERT34 023080
ERT35 023102
ERT36 023124
ERT37 023146
ERT38 023168
ERT39 023190
ERT40 023212
ERT41 023234
ERT42 023256
ERT43 023278
ERT44 023300
ERT45 023322
ERT46 023344
ERT47 023366
ERT48 023388
ERT49 023410
ERT50 023432
ERT51 023454
ERT52 023476
ERT53 023498
ERT54 023520
ERT55 023542
ERT56 023564
ERT57 023586
ERT58 023608
ERT59 023630
ERT60 023652
ERT61 023674
ERT62 023696
ERT63 023718
ERT64 023740
ERT65 023762
ERT66 023784
ERT67 023806
ERT68 023828
ERT69 023850
ERT70 023872
ERT71 023894
ERT72 023916
ERT73 023938
ERT74 023960
ERT75 023982
ERT76 024004
ERT77 024026
ERT78 024048
ERT79 024070
ERT80 024092
ERT81 024114
ERT82 024136
ERT83 024158
ERT84 024180
ERT85 024202
ERT86 024224
ERT87 024246
ERT88 024268
ERT89 024290
ERT90 024312
ERT91 024334
ERT92 024356
ERT93 024378
ERT94 024400
ERT95 024422
ERT96 024444
ERT97 024466
ERT98 024488
ERT99 024510
ERT100 024532
ERT101 024554
ERT102 024576
ERT103 024598
ERT104 024620
ERT105 024642
ERT106 024664
ERT107 024686
ERT108 024708
ERT109 024730
ERT110 024752
ERT111 024774
ERT112 024796
ERT113 024818
ERT114 024840
ERT115 024862
ERT116 024884
ERT117 024906
ERT118 024928
ERT119 024950
ERT120 024972
ERT121 024994
ERT122 025016
ERT123 025038
ERT124 025060
ERT125 025082
ERT126 025104
ERT127 025126
ERT128 025148
ERT129 025170
ERT130 025192
ERT131 025214
ERT132 025236
ERT133 025258
ERT134 025280
ERT135 025302
ERT136 025324
ERT137 025346
ERT138 025368
ERT139 025390
ERT140 025412
ERT141 025434
ERT142 025456
ERT143 025478
ERT144 025500
ERT145 025522
ERT146 025544
ERT147 025566
ERT148 025588
ERT149 025610
ERT150 025632
ERT151 025654
ERT152 025676
ERT153 025698
ERT154 025720
ERT155 025742
ERT156 025764
ERT157 025786
ERT158 025808
ERT159 025830
ERT160 025852
ERT161 025874
ERT162 025896
ERT163 025918
ERT164 025940
ERT165 025962
ERT166 025984
ERT167 026006
ERT168 026028
ERT169 026050
ERT170 026072
ERT171 026094
ERT172 026116
ERT173 026138
ERT174 026160
ERT175 026182
ERT176 026204
ERT177 026226
ERT178 026248
ERT179 026270
ERT180 026292
ERT181 026314
ERT182 026336
ERT183 026358
ERT184 026380
ERT185 026402
ERT186 026424
ERT187 026446
ERT188 026468
ERT189 026490
ERT190 026512
ERT191 026534
ERT192 026556
ERT193 026578
ERT194 026600
ERT195 026622
ERT196 026644
ERT197 026666
ERT198 026688
ERT199 026710
ERT200 026732
ERT201 026754
ERT202 026776
ERT203 026798
ERT204 026820
ERT205 026842
ERT206 026864
ERT207 026886
ERT208 026908
ERT209 026930
ERT210 026952
ERT211 026974
ERT212 026996
ERT213 027018
ERT214 027040
ERT215 027062
ERT216 027084
ERT217 027106
ERT218 027128
ERT219 027150
ERT220 027172
ERT221 027194
ERT222 027216
ERT223 027238
ERT224 027260
ERT225 027282
ERT226 027304
ERT227 027326
ERT228 027348
ERT229 027370
ERT230 027392
ERT231 027414
ERT232 027436
ERT233 027458
ERT234 027480
ERT235 027502
ERT236 027524
ERT237 027546
ERT238 027568
ERT239 027590
ERT240 027612
ERT241 027634
ERT242 027656
ERT243 027678
ERT244 027700
ERT245 027722
ERT246 027744
ERT247 027766
ERT248 027788
ERT249 027810
ERT250 027832
ERT251 027854
ERT252 027876
ERT253 027898
ERT254 027920
ERT255 027942
ERT256 027964
ERT257 027986
ERT258 028008
ERT259 028030
ERT260 028052
ERT261 028074
ERT262 028096
ERT263 028118
ERT264 028140
ERT265 028162
ERT266 028184
ERT267 028206
ERT268 028228
ERT269 028250
ERT270 028272
ERT271 028294
ERT272 028316
ERT273 028338
ERT274 028360
ERT275 028382
ERT276 028404
ERT277 028426
ERT278 028448
ERT279 028470
ERT280 028492
ERT281 028514
ERT282 028536
ERT283 028558
ERT284 028580
ERT285 028602
ERT286 028624
ERT287 028646
ERT288 028668
ERT289 028690
ERT290 028712
ERT291 028734
ERT292 028756
ERT293 028778
ERT294 028800
ERT295 028822
ERT296 028844
ERT297 028866
ERT298 028888
ERT299 028910
ERT300 028932
ERT301 028954
ERT302 028976
ERT303 028998
ERT304 029020
ERT305 029042
ERT306 029064
ERT307 029086
ERT308 029108
ERT309 029130
ERT310 029152
ERT311 029174
ERT312 029196
ERT313 029218
ERT314 029240
ERT315 029262
ERT316 029284
ERT317 029306
ERT318 029328
ERT319 029350
ERT320 029372
ERT321 029394
ERT322 029416
ERT323 029438
ERT324 029460
ERT325 029482
ERT326 029504
ERT327 029526
ERT328 029548
ERT329 029570
ERT330 029592
ERT331 029614
ERT332 029636
ERT333 029658
ERT334 029680
ERT335 029702
ERT336 029724
ERT337 029746
ERT338 029768
ERT339 029790
ERT340 029812
ERT341 029834
ERT342 029856
ERT343 029878
ERT344 029900
ERT345 029922
ERT346 029944
ERT347 029966
ERT348 029988
ERT349 030010
ERT350 030032
ERT351 030054
ERT352 030076
ERT353 030098
ERT354 030120
ERT355 030142
ERT356 030164
ERT357 030186
ERT358 030208
ERT359 030230
ERT360 030252
ERT361 030274
ERT362 030296
ERT363 030318
ERT364 030340
ERT365 030362
ERT366 030384
ERT367 030406
ERT368 030428
ERT369 030450
ERT370 030472
ERT371 030494
ERT372 030516
ERT373 030538
ERT374 030560
ERT375 030582
ERT376 030604
ERT377 030626
ERT378 030648
ERT379 030670
ERT380 030692
ERT381 030714
ERT382 030736
ERT383 030758
ERT384 030780
ERT385 030802
ERT386 030824
ERT387 030846
ERT388 030868
ERT389 030890
ERT390 030912
ERT391 030934
ERT392 030956
ERT393 030978
ERT394 031000
ERT395 031022
ERT396 031044
ERT397 031066
ERT398 031088
ERT399 031110
ERT400 031132
ERT401 031154
ERT402 031176
ERT403 031198
ERT404 031220
ERT405 031242
ERT406 031264
ERT407 031286
ERT408 031308
ERT409 031330
ERT410 031352
ERT411 031374
ERT412 031396
ERT413 031418
ERT414 031440
ERT415 031462
ERT416 031484
ERT417 031506
ERT418 031528
ERT419 031550
ERT420 031572
ERT421 031594
ERT422 031616
ERT423 031638
ERT424 031660
ERT425 031682
ERT426 031704
ERT427 031726
ERT428 031748
ERT429 031770
ERT430 031792
ERT431 031814
ERT432 031836
ERT433 031858
ERT434 031880
ERT435 031902
ERT436 031924
ERT437 031946
ERT438 031968
ERT439 031990
ERT440 032012
ERT441 032034
ERT442 032056
ERT443 032078
ERT444 032100
ERT445 032122
ERT446 032144
ERT447 032166
ERT448 032188
ERT449 032210
ERT450 032232
ERT451 032254
ERT452 032276
ERT453 032298
ERT454 032320
ERT455 032342
ERT456 032364
ERT457 032386
ERT458 032408
ERT459 032430
ERT460 032452
ERT461 032474
ERT462 032496
ERT463 032518
ERT464 032540
ERT465 032562
ERT466 032584
ERT467 032606
ERT468 032628
ERT469 032650
ERT470 032672
ERT471 032694
ERT472 032716
ERT473 032738
ERT474 032760
ERT475 032782
ERT476 032804
ERT477 032826
ERT478 032848
ERT479 032870
ERT480 032892
ERT481 032914
ERT482 032936
ERT483 032958
ERT484 032980
ERT485 033002
ERT486 033024
ERT487 033046
ERT488 033068
ERT489 033090
ERT490 033112
ERT491 033134
ERT492 033156
ERT493 033178
ERT494 033200
ERT495 033222
ERT496 033244
ERT497 033266
ERT498 033288
ERT499 033310
ERT500 033332
ERT501 033354
ERT502 033376
ERT503 033398
ERT504 033420
ERT505 033442
ERT506 033464
ERT507 033486
ERT508 033508
ERT509 033530
ERT510 033552
ERT511 033574
ERT512 033596
ERT513 033618
ERT514 033640
ERT515 033662
ERT516 033684
ERT517 033706
ERT518 033728
ERT519 033750
ERT520 033772
ERT521 033794
ERT522 033816
ERT523 033838
ERT524 033860
ERT525 033882
ERT526 033904
ERT527 033926
ERT528 033948
ERT529 033970
ERT530 033992
ERT531 034014
ERT532 034036
ERT533 034058
ERT534 034080
ERT535 034102
ERT536 034124
ERT537 034146
ERT538 034168
ERT539 034190
ERT540 034212
ERT541 034234
ERT542 034256
ERT543 034278
ERT544 034300
ERT545 034322
ERT546 034344
ERT547 034366
ERT548 034388
ERT549 034410
ERT550 034432
ERT551 034454
ERT552 034476
ERT553 034498
ERT554 034520
ERT555 034542
ERT556 034564
ERT557 034586
ERT558 034608
ERT559 034630
ERT560 034652
ERT561 034674
ERT562 034696
ERT563 034718
ERT564 034740
ERT565 034762
ERT566 034784
ERT567 034806
ERT568 034828
ERT569 034850
ERT570 034872
ERT571 034894
ERT572 034916
ERT573 034938
ERT574 034960
ERT575 034982
ERT576 035004
ERT577 035026
ERT578 035048
ERT579 035070
ERT580 035092
ERT581 035114
ERT582 035136
ERT583 035158
ERT584 035180
ERT585 035202
ERT586 035224
ERT587 035246
ERT588 035268
ERT589 035290
ERT590 035312
ERT591 035334
ERT592 035356
ERT593 035378
ERT594 035400
ERT595 035422
ERT596 035444
ERT597 035466
ERT598 035488
ERT599 035510
ERT600 035532
ERT601 035554
ERT602 035576
ERT603 035598
ERT604 035620
ERT605 035642
ERT606 035664
ERT607 035686
ERT608 035708
ERT609 035730
ERT610 035752
ERT611 035774
ERT612 035796
ERT613 035818
ERT614 035840
ERT615 035862
ERT616 035884
ERT617 035906
ERT618 035928
ERT619 035950
ERT620 035972
ERT621 035994
ERT622 036016
ERT623 036038
ERT624 036060
ERT625 036082
ERT626 036104
ERT627 036126
ERT628 036148
ERT629 036170
ERT630 036192
ERT631 036214
ERT632 036236
ERT633 036258
ERT634 036280
ERT635 036302
ERT636 036324
ERT637 036346
ERT638 036368
ERT639 036390
ERT640 036412
ERT641 036434
ERT642 036456
ERT643 036478
ERT644 036500
ERT645 036522
ERT646 036544
ERT647 036566
ERT648 036588
ERT649 036610
ERT650 036632
ERT651 036654
ERT652 036676
ERT653 036698
ERT654 036720
ERT655 036742
ERT656 036764
ERT657 036786
ERT658 036808
ERT659 036830
ERT660 036852
ERT661 036874
ERT662 036896
ERT663 036918
ERT664 036940
ERT665 036962
ERT666 036984
ERT667 037006
ERT668 037028
ERT669 037050
ERT670 037072
ERT671 037094
ERT672 037116
ERT673 037138
ERT674 037160
ERT675 037182
ERT676 037204
ERT677 037226
ERT678 037248
ERT679 037270
ERT680 037292
ERT681 037314
ERT682 037336
ERT683 037358
ERT684 037380
ERT685 037402
ERT686 037424
ERT687 037446
ERT688 037468
ERT689 037490
ERT690 037512
ERT691 037534
ERT692 037556
ERT693 037578
ERT694 037600
ERT695 037622
ERT696 037644
ERT697 037666
ERT698 037688
ERT699 037710
ERT700 037732
ERT701 037754
ERT702 037776
ERT703 037798
ERT704 037820
ERT705 037842
ERT706 037864
ERT707 037886
ERT708 037908
ERT709 037930
ERT710 037952
ERT711 037974
ERT712 037996
ERT713 038018
ERT714 038040
ERT715 038062
ERT716 038084
ERT717 038106
ERT718 038128
ERT719 038150
ERT720 038172
ERT721 038194
ERT722 038216
ERT723 038238
ERT724 038260
ERT725 038282
ERT726 038304
ERT727 038326
ERT728 038348
ERT729 038370
ERT730 038392
ERT731 038414
ERT732 038436
ERT733 038458
ERT734 038480
ERT735 038502
ERT736 038524
ERT737 038546
ERT738 038568
ERT739 038590
ERT740 038612
ERT741 038634
ERT742 038656
ERT743 038678
ERT744 038700
ERT745 038722
ERT746 038744
ERT747 038766
ERT748 038788
ERT749 038810
ERT750 038832
ERT751 038854
ERT752 038876
ERT753 038898
ERT754 038920
ERT755 038942
ERT756 038964
ERT757 038986
ERT758 039008
ERT759 039030
ERT760 039052
ERT761 039074
ERT762 039096
ERT763 039118
ERT764 039140
ERT765 039162
ERT766 039184
ERT767 039206
ERT768 039228
ERT769 039250
ERT770 039272
ERT771 039294
ERT772 039316
ERT773 039338
ERT774 039360
ERT775 039382
ERT776 039404
ERT777 039426
ERT778 039448
ERT779 039470
ERT780 039492
ERT781 039514
ERT782 039536
ERT783 039558
ERT784 039580
ERT785 039602
ERT786 039624
ERT787 039646
ERT788 039668
ERT789 039690
ERT790 039712
ERT791 039734
ERT792 039756
ERT793 039778
ERT794 039800
ERT795 039822
ERT796 039844
ERT797 039866
ERT798 039888
ERT799 039910
ERT800 039932
ERT801 039954
ERT802 039976
ERT803 039998
ERT804 040020
ERT805 040042
ERT806 040064
ERT807 040086
ERT808 040108
ERT809 040130
ERT810 040152
ERT811 040174
ERT812 040196
ERT813 040218
ERT814 040240
ERT815 040262
ERT816 040284
ERT817 040306
ERT818 040328
ERT819 040350
ERT820 040372
ERT821 040394
ERT822 040416
ERT823 040438
ERT824 040460
ERT825 040482
ERT826 040504
ERT827 040526
ERT828 040548
ERT829 040570
ERT830 040592
ERT831 040614
ERT832 040636
ERT833 040658
ERT834 040680
ERT835 040702
ERT836 040724
ERT837 040746
ERT838 040768
ERT839 040790
ERT840 040812
ERT841 040834
ERT842 040856
ERT843 040878
ERT844 040900
ERT845 040922
ERT846 040944
ERT847 040966
ERT848 040988
ERT849 041010
ERT850 041032
ERT851 041054
ERT852 041076
ERT853 041098
ERT854 041120
ERT855 041142
ERT856 041164
ERT857 041186
ERT858 041208
ERT859 041230
ERT860 041252
ERT861 041274
ERT862 041296
ERT863 041318
ERT864 041340
ERT865 041362
ERT866 041384
ERT867 041406
ERT868 041428
ERT869 041450
ERT870 041472
ERT871 041494
ERT872 041516
ERT873 041538
ERT874 041560
ERT875 041582
ERT876 041604
ERT877 041626
ERT878 041648
ERT879 041670
ERT880 041692
ERT881 041714
ERT882 041736
ERT883 041758
ERT884 041780
ERT885 041802
ERT886 041824
ERT887 041846
ERT888 041868
ERT889 041890
ERT890 041912
ERT891 041934
ERT892 041956
ERT893 041978
ERT894 042000
ERT895 042022
ERT896 042044
ERT897 042066
ERT898 042088
ERT899 042110
ERT900 042132
ERT901 042154
ERT902 042176
ERT903 042198
ERT904 042220
ERT905 042242
ERT906 042264
ERT907 042286
ERT908 042308
ERT909 042330
ERT910 042352
ERT911 042374
ERT912 042396
ERT913 042418
ERT914 042440
ERT915 042462
ERT916 042484
ERT917 042506
ERT918 042528
ERT919 042550
ERT920 042572
ERT921 042594
ERT922 042616
ERT923 042638
ERT924 042660
ERT925 042682
ERT926 042704
ERT927 042726
ERT928 042748
ERT929 042770
ERT930 042792
ERT931 042814
ERT932 042836
ERT933 042858
ERT934 042880
ERT935 042902
ERT936 042924
ERT937 042946
ERT938 042968
ERT939 042990
ERT940 0430

FO1

MAINDEC-11-FPP34-A
 DFFPCA.P11 31-OCT-76 17:16

POP 1: 34 FPP DIAGNOSTIC
 SYMBOL TABLE

MACY11 27(1006) 31-OCT-76 17:35 PAGE 213

PPP10	007136	RRBTP1	020630	SSC25	033760	TAB	=	000011	TST54	020000
PPP15	007156	RRBTP2	020640	SSC30	034010	TBITVE	=	000014	TST55	020000
PPP2	006756	RRBTP3	020650	SSSA1	010252	TCCBFO	=	025650	TST56	020000
PPP3	007020	RRB1	020510	SSSBFO	010242	TCCBF1	=	025660	TST57	020000
PPP4	007040	RRB10	020652	SSSCON	010410	TCCDON	=	025670	TST58	020000
PROGNUM	=	RRB15	020672	SSSTP1	010262	TCC1	=	025536	TST59	020000
PRC	=	RRB2	020564	SSSTP2	010272	TCC2	=	025560	TST60	020000
PR1	=	RRB20	020710	SSS1	010104	TCC3	=	025624	TST61	020000
PR2	=	RRCDON	033542	SSS10	010302	TKVEC	=	000060	TST62	020000
PR3	=	RACTBO	033410	SSS11	010324	TPVEC	=	000064	TST63	020000
PR4	=	RACTB1	033414	SSS15	010334	TRAVE	=	000034	TST64	020000
PR5	=	RACTB2	033424	SSS2	010162	TRAVEC	=	000014	TST65	020000
PR6	=	RC1	033302	SSS20	010352	TST1	=	006576	TST66	020000
PR7	=	RC10	033430	SSS25	010372	TST10	=	011132	TST67	020000
PS	=	RC15	033450	STACK	=	TST11	=	011366	TST7	020000
PSM	=	RC2	033356	START	=	TST12	=	011640	TST70	020000
PARVEC	=	RC20	033472	STCOFS	=	TST13	=	012474	TST71	020000
QOBON	=	RC25	033514	STCOY	=	TST14	=	013330	TST72	020000
QCBTP1	020402	RRDON	010100	STCOFS	=	TST15	=	013430	TST73	020000
QCBTP2	020422	RRREXP	027730	STCFT	=	TST16	=	013642	TST74	020000
QCB1	020266	RRRTP1	007710	STCIBF	=	TST17	=	013742	TST75	020000
QCB10	020432	RRRTP2	007720	STCSUB	=	TST2	=	006736	TST76	020000
QCB15	020452	RRR1	007562	STKLMT	=	TST20	=	014042	TTBON	020000
QCB2	020336	RRR10	007740	STXBF	=	TST21	=	014260	TTBTP1	020000
QCB20	020470	RRR11	007762	STXSUB	=	TST22	=	014550	TTBTP2	020000
QCCON	033276	RRR12	010002	SWR	=	TST23	=	015040	TTB1	020000
QCTBO	033144	RRR15	010034	SWREG	=	TST24	=	015330	TTB10	020000
QCTB1	033150	RRR2	007640	SWC	=	TST25	=	015630	TTB15	020000
QCTB2	033160	RRR25	010014	SWC	=	TST26	=	016130	TTB2	020000
QCC1	033040	RRR3	007642	SWC1	=	TST27	=	016424	TTB20	020000
QCC10	033164	RRR4	007666	SWC2	=	TST3	=	007176	TTCON	020000
QCC15	033204	RSETUP	=	SWC3	=	TST30	=	016726	TTCTBO	020000
QCC2	033112	R6	=	SWC4	=	TST31	=	017170	TTCTB1	020000
QCC20	033226	R7	=	SWC5	=	TST32	=	017442	TTCTB2	020000
QCC25	033250	SAVREG	=	SWC6	=	TST33	=	017644	TTCTB3	020000
QCBFO	007340	SPACE	=	SWC7	=	TST34	=	020054	TTCTB4	020000
QCBF1	007354	SSBON	021146	SWC8	=	TST35	=	020264	TTCTB5	020000
QCCON	007556	SSBTP1	021052	SWC9	=	TST36	=	020506	TTCTB6	020000
QCBTP1	007370	SSBTP2	021062	SW1	=	TST37	=	020726	TTCTB7	020000
QCC1	007200	SSBTP3	021072	SW10	=	TST4	=	007560	TTCTB8	020000
QCC10	007400	SSB1	020730	SW11	=	TST40	=	021150	TTCTB9	020000
QCC2	007216	SSB10	021074	SW12	=	TST41	=	021360	TTCTB10	020000
QCC20	007420	SSB15	021114	SW13	=	TST42	=	021606	TTCTB11	020000
QCC22	007434	SSB2	021006	SW14	=	TST43	=	022050	TTCTB12	020000
QCC23	007466	SSB20	021132	SW15	=	TST44	=	022322	TTCTB13	020000
QCC24	007514	SSCON	034014	SW2	=	TST45	=	023604	TTCTB14	020000
QCC25	007540	SSCTBO	033660	SW3	=	TST46	=	023764	TTCTB15	020000
QCC3	007254	SSCTB1	033664	SW4	=	TST47	=	024142	TTCTB16	020000
QCC4	007302	SSC1	033546	SW5	=	TST5	=	010102	TTCTB17	020000
ROCHR	=	SSC10	033674	SW6	=	TST50	=	024336	TTCTB18	020000
RESREG	=	SSC15	033714	SW7	=	TST51	=	024546	TTCTB19	020000
RESVEC	=	SSC2	033616	SW8	=	TST52	=	024750	TTCTB20	020000
RRBON	020724	SSC20	033736	SW9	=	TST53	=	025166	TTCTB21	020000

TYPE	N	107401	WAB2	022402	YYBCON	022320	SCM1	N	000024	SICNT	001104
TYPOC	N	107402	WAB3	022460	YYBTP1	022204	SCM2	N	000050	SILLOP	042124
TYPON	N	107404	WAB4	022536	YYBTP2	022214	SCM3	N	000024	SINTAG	001125
TYP05	N	107403	WAB5	022614	YYBTP3	022224	SCM4	N	000024	SITEMB	001114
UUBCON		021604	WAB6	022672	YYB1	022552	SCNTLG		041633	SIF	001314
UUBTP1		021436	WAB7	022750	YYB10	022246	SCNTLU		041626	SIFLG	041221
UUBTP2		021510	WAB8	022826	YYB15	022266	SCPLOP		001344	SLOOP	037422
UUB1		021362	WAB9	023104	YTB2	022132	SCRLF		001313	SLPACR	001106
UUB10		021520	WACCON	036152	YTB20	022304	SCOW0		001402	SLPERA	001110
UUB15		021556	WAC1	034514	YTB25	022226	SCOW1		001404	SMADR1	001350
UUB2		021434	WAC10	035154	YYCON	026734	SCOW10		001426	SMADR2	001354
UUB20		021540	WAC11	035220	YYC1	036226	SCOW11		001430	SMADR3	001360
UUCBFO		034376	WAC12	035264	YYC2	036264	SCOW12		001432	SMADR4	001364
UUCCON		034404	WAC13	035320	YYC3	036322	SCOW13		001434	SMAY1	001316
UUCTP1		034364	WAC14	035374	YYC4	036360	SCOW14		001436	SMAMS1	001346
UUC1		034304	WAC15	035440	YYC5	036416	SCOW15		001440	SMAMS2	001352
UUC2		034334	WAC16	035504	YYC6	036454	SCOW2		001406	SMAMS3	001356
UUC3		034344	WAC17	035550	YYYDON	013326	SCOW3		001410	SMAMS4	001362
UUUA1		011016	WAC18	035614	YYY1	012476	SCOW4		001412	SMAPCR	006074
UUUA2		011020	WAC2	034560	YYY2	012552	SCOW5		001414	SMFLG	041220
UUUA3		011022	WAC4	034624	YYY3	012626	SCOW6		001416	SMFEN	041651
UUBFO		011034	WAC5	034670	YYY4	012702	SCOW7		001420	SMSCAD	001332
UUCON		011130	WAC6	034734	YYYS	012756	SCOW8		001422	SMSQLG	001334
UUCTP1		011030	WAC7	035000	ZZCBF	037126	SCOW9		001424	SMSTY	001316
UUL1		010664	WAC8	035044	ZZCON	037140	SCDVCT		001326	SMWAR	041640
UUL10		011040	WAC9	035110	ZZC1	036740	SCDVM		001374	SMTYP1	001347
UUL11		011062	WUBFO	011516	ZZC10	037100	SCDAGN		037364	SMTYP2	001353
UUL15		011074	WUBF1	011536	ZZC12	037114	SENDAD		037354	SMTYP3	001357
UUL2		010750	WUKON	011636	ZZC15	037120	SENCT		037176	SMTYP4	001363
UUL20		011112	WUTP1	011526	ZZC2	036750	SENULL		037430	SMXCNT	037712
UUL3		010774	WWW1	011370	ZZC3	036776	SENV		001336	SNULL	001154
VVBOON		014040	WWW10	011546	ZZC5	037074	SENVN		001337	SNWTST =	000001
VVB1		012744	WWW11	011570	ZZZCON	013426	SEOP		037142	SOCNT	040752
VVB10		014006	WWW15	011602	ZZZ1	013332	SEOPCT		037170	SCODE	040754
VVB15		014024	WWW2	011460	ZZZ10	013374	SEFLG		001103	SOVER	037676
VVB2		013762	WWW20	011620	ZZZ15	013412	SERMAX		001115	SPASS	001324
VVCBFO		034502	XXBON	022046	ZZZ2	013350	SERROR		037714	SPASTM	006100
VVCON		034510	XXBTP1	021734	ZZZ11	013374	SERRPC		001116	SPARAD	042106
VVCTP1		034470	XXBTP2	021744	ZZZ10	013374	SERRTB		001442	SPARDN	041746
VVC1		034410	XXB1	021610	ZZZ15	013412	SERTTL		001112	SPWRMG	042102
VVC2		034440	XXB10	021774	ZZZ20	013450	SESCAP		001304	SPWRUP	042020
VVC3		034450	XXB15	022014	ZZZ25	013488	SETABL		001336	SGUES	001312
VVBF0		011254	XXB2	021662	ZZZ30	013526	SETEND		001442	SROCHR	041506
VVCON		011364	XXB20	022032	ZZZ35	013564	SFATAL		001320	SRCST =	000001
VVTP1		011264	XXB25	021754	ZZZ40	013602	SFFLG		041222	SREGAD	001160
VV1		011134	XXCON	036222	ZZZ45	013640	SFILLC		001156	SREG0	001162
VV10		011274	XXC1	036156	ZZZ50	013678	SFILLS		001155	SREG1	001164
VV11		011316	XXXCON	012472	ZZZ55	013716	SGADR		001120	SREG10	001202
VV15		011330	XXX1	011642	ZZZ60	013754	SGDAT		001124	SREG11	001204
VV2		011216	XXX2	011716	ZZZ65	013792	SGET42		037326	SREG12	001206
VV20		011346	XXX3	011772	ZZZ70	013830	SGTSWR		041274	SREG13	001210
WABOON		023602	XXX4	012046	ZZZ75	013868	SHD =		000003	SREG14	001212
WAB1		022324	XXX5	012122	ZZZ80	013906	SHIBTS		006072	SREG15	001214

H01

MAINDEC-11-FPP34-A
DFFPCA.P11

PDP 11 34 FPP DIAGNOSTIC
31-OCT-76 17:16 SYMBOL TABLE

MACY11 27(1006) 31-OCT-76 17:35 PAGE 215

\$REG16	001216	\$SCOPE	037434	STMP1	001234	STMP6	001246	STYPOS	040530
\$REG17	001220	\$SETUP =	000137	STMP10	001252	STMP7	001250	\$UNIT	001330
\$REG18	001166	\$STUP =	177777	STMP11	001254	\$IN =	000076	\$UNITM	006102
\$REG19	001222	\$SVLAC	037642	STMP12	001256	STPB	001152	\$USWR	001342
\$REG20	001224	\$SYPC =	006072	STMP13	001260	STPFLG	001157	\$VECT1	001366
\$REG21	001226	\$SWR =	177400	STMP14	001262	STPS	001150	\$VECT2	001370
\$REG22	001230	\$SWREG	001340	STMP15	001264	STRAP	041662	\$XTSTR	037446
\$REG23	001170	\$SWRMC =	000000	STMP16	001266	STRAP2	041704	\$SET4 =	000001
\$REG24	001172	\$SWRMS =	000200	STMP17	001270	STRP =	000014	\$OFILL	040753
\$REG25	001174	\$TAB	043005	STMP2	001236	STRPAD	041716	.LPER =	071702
\$REG26	001176	\$TBIT	037426	STMP20	001272	STSTM	006076	.RSET	042660
\$REG27	001200	\$TERM =	000030	STMP21	001274	STSTM	001102	.S =	006072
\$RESRE	040210	\$TESTN	001322	STMP22	001276	\$TYPE	040246		
\$R*NO	037424	\$TIMES	001302	STMP23	001300	\$TYPEC	040460		
\$R*RN	037420	\$TKB	001146	STMP3	001240	\$TYPEX	040526		
\$SAVRE	040152	\$TKS	001144	STMP4	001242	\$TYPOC	040554		
\$SAVR6	042130	\$TMP0	001232	STMP5	001244	\$TYPOH	040570		

. ABS. 071702 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DFFPCA.BIN, DSKZ:DFFPCA.SEO/SOL+DSKZ:DFFPCA.P11
RUN-TIME: 106 94 8 SECONDS
RUN-TIME RATIO: 1179/208=5.6
CORE USED: 31K (61 PAGES)

