

FP11.F

FP11F FLTG PNT PRT C
CKFPCB0

AH-F638B-MC
FICHE 1 OF 2

MAR 1980
COPYRIGHT © 1980
MADE IN USA



The main body of the document is a large grid of data, likely a flight log or a technical record. It consists of approximately 15 columns and 25 rows of data points. Each cell in the grid contains small, faint text or numbers, which are difficult to read due to the low resolution and high density of the data. The data appears to be organized in a structured format, possibly representing time, location, or specific parameters over a period of time.

FP11-F

FP11F FLTG PNT PRT C
CKFPCBO

AH-F638B-MC
FICHE 2 OF 2

MAR 1980
COPYRIGHT © 1980
MADE IN USA



12

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45
46	47	48
49	50	51
52	53	54
55	56	57
58	59	60
61	62	63
64	65	66
67	68	69
70	71	72
73	74	75
76	77	78
79	80	81
82	83	84
85	86	87
88	89	90
91	92	93
94	95	96
97	98	99
100	101	102



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

.REM &

IDENTIFICATION

PRODUCT CODE: AC-F636B-MC
PRODUCT NAME: CKFPCB0 FP11F FLTG PNT PRT C
PRODUCT DATE: DECEMBER, 1979
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: ANTHONY VEZZA, DAN MILLEVILLE

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

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

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

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

HISTORY

NO CHANGES TO THE 11/34 FLOATING POINT DIAGNOSTIC PART 'A' WERE FOUND TO BE NEEDED TO ADAPT IT FOR USE ON THE 11/44.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'B' VERSION COVER THE 11/44:

1. TEST 22 - PROCESSOR LOOKS TO SEE IF APT IS CONTROLLING THE TEST, AND IF IT IS, CHECKS TO SEE IF THE USER HAS SELECTED THIS TEST BY CHECKING BIT 7 IN THE SWITCH REGISTER. IT HAS ALSO BEEN CHANGED SO THAT IF BIT 7 IS *ONE*, THE CODE WILL SELECT THE TEST.

THE FOLLOWING WAS ADDED TO THE 11/34 FLOATING POINT DIAGNOSTIC TO MAKE THE 'C' VERSION COVER THE 11/44:

1. TEST 76 - CHECKS THAT FP PROCESSOR DOESN'T ACCESS D-SPACE UNTIL CONDITIONS WARRANT.
2. TEST 77 - CHECKS THAT SR1 MATCHES WHAT ACTUALLY HAPPENED TO THE REGISTER OF THE INSTRUCTION, AND THAT THE VALUE OF AUTO INCREMENT/DECREMENT WAS PROPER.

THE FOLLOWING WAS ADDED TO THE 'C' VERSION TO FURTHER INTENSIFY THE TEST:

1. TEST 77 - A BYTE TABLE OF EXPECTED DATA FOR SR1 CHECKS TO MAKE SURE THAT THE VALUE OF THE INCREMENT/DECREMENT IS PROPER FOR THAT INSTRUCTION.

77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND OPERATOR INTERACTION
- 5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.3 OPERATOR ACTION
- 6. ERRORS
 - 6.1 SUMMARY
 - 6.2 ERROR RECOVERY
- 7. RESTRICTIONS
 - 7.1 STARTING RESTRICTIONS
 - 7.2 OPERATING RESTRICTIONS
- 8. MISCELLANEOUS
 - 8.1 EXECUTION TIMES
 - 8.2 STACK POINTER
 - 8.3 PASS COUNT
 - 8.4 T-BIT TRAPPING
 - 8.5 SOFTWARE SWITCH REGISTER
 - 8.6 INTERRUPTS TESTS
 - 8.7 ACT, APT AND XXDP COMPATIBILITY
- 9. PROGRAM DESCRIPTION
 - 9.1 CKFPCBO
- 10. LISTING
 - 10.1 CKFPCBO

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
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184

1.

ABSTRACT

THE THREE PROGRAMS:

CKFPAAO CKFPBAO CKFPCBO

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/44 FP11-F 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 161 (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 FP11-F. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. CKFPAAO

CKFPAAO 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. CKFPBAO

CKFPBAO TESTS:

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

C. CKFPCBO

CKFPCBO TESTS:

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

185 ABSF AND ABS
186 TSTF AND TSTD
187 NEGF, ABSF AND TSTF (ALL SOURCE MODES)
188 NEGF, ABSF AND TSTF (ALL SOURCE MODES)
189 LDFPS (ALL SOURCE MODES)
190 LDCIF AND LDCLF
191 LDCID AND LDCLD
192 LDEXP
193 STFPS (ALL DESTINATION MODES)
194 STCFI AND STCFI
195 STCDL AND STCDI
196 STEXP
197 STST
198 I AND D SPACE TESTS (ALL MODES AND REGS 0 AND 7)
199 AUTO INCREMENT/DECREMENT CHECK - SR1 (ALL MODES AND REGS 1 AND 7)
200
201

202 2. REQUIREMENTS
203

204
205 2.1 EQUIPMENT
206 A PDP 11/44 WITH CONSOLE AND AN FP11-F FLOATING
207 POINT PROCESSOR. NOTE THAT A SPECIAL INTERRUPTS TEST
208 MODULE IS BEING DESIGNED FOR USE IN THE MANUFAC-
209 TURING ENVIRONMENT. WHEN THIS DEVICE IS PRESENT THE
210 PROGRAM CKFPBAO WILL MAKE USE OF IT TO TEST THE FPP
211 INTERRUPT ON BUS REQUEST FUNCTIONS.
212

213 2.2 STORAGE
214
215 ALL THREE PROGRAM REQUIRE A MEMORY SYSTEM OF AT
216 LEAST 16K TO LOAD AND RUN.
217

218 2.3 PRELIMINARY PROGRAMS
219
220 THESE THREE DIAGNOSTICS WILL ASSUME THAT THE PDP
221 11/44 CENTRAL PROCESSOR IS FAULTLESS, THEREFORE WHEN
222 IN DOUBT RUN THE PDP 11/44 PROCESSOR DIAGNOSTICS
223 BEFORE THESE FP11-F DIAGNOSTICS.
224

225
226 3. LOADING PROCEDURE
227

228
229 THE PROGRAMS WILL BE SUPPLIED ON THE USUAL
230 DIAGNOSTIC MEDIA. REFER TO THE XXDP OPERATING
231 MANUAL FOR FURTHER INFORMATION.
232

233 4. STARTING PROCEDURE
234

235
236 4.1 CONTROL SWITCH SETTINGS

237
238 SEE SECTION 5.1
239

240 4.2 PROGRAM AND OPERATOR ACTION
241

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

256
 257 5. OPERATING PROCEDURE
 258 -----
 259

260 5.1 OPERATIONAL SWITCH SETTINGS

261 THE SWITCH SETTING ARE:

	OCTAL	
265 SW<15>=1...	100000	HALT ON ERROR
266 SW<14>=1...	40000	LOOP ON CURRENT TEST
267 SW<13>=1...	20000	INHIBIT ERROR TYPE OUTS
268 SW<12>=1...	10000	INHIBIT T-BIT TRAPPING
269 SW<11>=1...	4000	INHIBIT ITERATIONS
270 SW<10>=1...	2000	RING TTY BELL ON ERROR
271 SW<9>=1....	1000	LOOP ON ERROR
272 SW<8>=1....	400	LOOP ON TEST SPECIFIED IN SW<6> THROUGH SW<0>
274 SW<7>=1....	200	PRINT ERROR SUMMARY EVEN IF SW<13>=1, THIS APPLIES ONLY TO PROGRAM CKFPAAO.
277 SW<7>=1....	200	SELECT CORRECT INTERRUPT TEST IN PROGRAM CKFPBAO.

280
 281 6. ERRORS
 282 -----
 283

284 6.1 SUMMARIES

285 IN PROGRAM CKFPAAO TESTS 1 AND 11 HAVE A SPECIAL
 286 ERROR SUMMARY FEATURE. THESE TWO TEST RUN MANY
 287 TEST PATTERNS THROUGH THE LOGIC. AFTER AN ERROR
 288 IS ENCOUNTERED, ONLY THE FIRST FIVE ERRORS ARE
 289 REPORTED (TYPED ON THE TTY). EVERY ERROR THOUGH IS
 290 LOGGED AND AN ERROR SUMMARY IS PRINTED WHEN THE
 291 TEST IS COMPLETE. NOTE THAT IS SW<13>=1 THIS
 292 SUMMARY WILL NOT BE TYPED UNLESS SW<7>=1. IN OTHER
 293 WORDS TO GET JUST AN ERROR SUMMARY FROM EITHER OF
 294 THESE TWO TESTS 1 AND 11 IN PROGRAM CKFPAAO BOTH
 295 SWITCHES 13 AND 7 MUST = 1.
 296

297 6. ERROR RECOVERY
 298

299
300 SW<15:9>=0... MOST ERRORS WILL CAUSE EXECUTION TO
301 GO TO THE START OF THE NEXT TEST
302 AFTER THE MESSAGE IS TYPED. A FEW
303 TESTS ARE IN SECTIONS. IN THESE
304 TESTS AN ERROR WILL CAUSE EXECUTION
305 TO GO TO THE NEXT SECTION AFTER THE
306 MESSAGE IS TYPED.
307 SW<15>=1.. THE PROGRAM WILL HALT AFTER TYPING
308 THE ERROR MESSAGE. PRESSING THE
309 CONSOLE CONTINUE WILL CAUSE THE
310 PROGRAM TO CONTINUE AS IF SW<15>=0.
311
312 7. RESTRICTIONS
313 -----
314
315 NONE
316
317
318 8. MISCELLANEOUS
319 -----
320
321 8.1 EXECUTION TIMES
322
323 LESS THAN 10 SECONDS FOR EACH PROGRAM ON ANY PASS.
324
325 8.2 STACK POINTER
326
327 THE STACK POINTER IS INITIALIZED TO 1100 IN EACH OF
328 THE THREE PROGRAMS.
329
330 8.3 PASS COUNT
331 THE PROGRAM MAKES ONE PASS FOR EACH END OF PASS
332 MESSAGE TYPED. THE END OF PASS MESSAGE DESCRIBES
333 THE TOTAL NUMBER OF PASSES COMPLETED AND THE TOTAL
334 NUMBER OF ERRORS SINCE THE LAST END OF PASS MESSAGE.
335
336 8.4 T-BIT TRAPPING
337
338 IF SW<12>=0 EACH PROGRAM WILL RUN WITH TRACE TRAPS
339 ON EVERY OTHER PASS. FIRST PASS WILL NOT ENABLE
340 TRACE TRAPS. NOTE SW<12>=1 DISABLES T-BIT TRAPS.
341
342 8.5 SOFTWARE SWITCH REGISTER
343
344 IF THE USER DESIRES, A SOFTWARE SWITCH REGISTER CAN
345 BE EXAMINED OR MODIFIED AT ANY TIME BY THE USER IF
346 HE TYPES CNTRL/G WHILE THE PROGRAM IS RUNNING. THIS
347 CNTRL/G WILL CAUSE THE CONTENTS OF THE SOFTWARE
348 SWITCH REGISTER TO BE TYPED ON THE TTY AND ASK THE
349 USER FOR A NEW VALUE. WHEN THE USER TYPES A VALUE
350 AND CARRIAGE RETURN THEN THE PROGRAM WILL RESUME
351 TESTING AT THE SAME POINT AT WHICH IT LEFT OFF WHEN
352 THE USER TYPED CNTRL/G. NOTE THAT WHEN NOT RUNNING
353 UNDER ACT, APT OR CHAIN THE USER WILL BE ASKED
354 FOR A SOFTWARE SWITCH REGISTER VALUE AFTER LOADING
355 ADDRESS 200 AND STARTING THE PROGRAM THE FIRST TIME

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
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412

THE PROGRAM IS RUN AFTER LOADING ONLY IF THE
CONSOLE SWITCH REGISTER CONTAINS 177777.

8.6 INTERRUPTS TEST

IN PROGRAM CKFPBAO 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>=1 THIS TEST
WILL BE RUN. IF SW<7>=0, 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

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.

413
414 TEST 5 FDST MODE 4 TEST
415 -----
416 THIS IS A TEST OF STD WITH FDST MODE 4.
417
418 TEST 6 FDST MODE 3 TEST
419 -----
420 THIS IS A TEST OF FDST MODE 3 USING STD.
421
422 TEST 7 FDST MODE 5 TEST
423 -----
424 THIS IS A TEST OF FDST MODE 5 USING STD.
425
426 TEST 10 FDST MODE 6, INDEX MODE, TEST
427 -----
428 THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING
429 STD.
430
431 TEST 11 FDST MODE 7, INDEX DEFERRED MODE, TEST
432 -----
433 THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE,
434 USING STD.
435
436 TEST 12 STCFD TEST
437 -----
438 THIS IS A TEST OF THE STCFD INSTRUCTION.
439
440 TEST 13 STCDF TEST
441 -----
442 THIS IS A TEST OF THE STCDF INSTRUCTION.
443
444 TEST 14 STCFD WITH ILLEGAL ACCUMULATOR TEST
445 -----
446 THIS TEST STCFD WITH ILLEGAL AC 6.
447
448 TEST 15 CLRD TEST
449 -----
450 THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.
451
452 TEST 16 CLRD WITH ILLEGAL ACCUMULATOR TEST
453 -----
454 THIS IS A TEST OF CLRD WITH ILLEGAL AC7.
455
456 TEST 17 NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
457 -----
458 THIS IS A TEST OF THE SPECIAL DEST FLOWS USING THE
459
460
461
462
463
464
465
466
467
468
469

470 NEGD INST WITH MODE ZERO AND ILLEGAL AC7.
471
472 TEST 20 NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
473 -----
474
475 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
476 THE NEGD INSTRUCTION IS USED TO TEST MODE 0
477
478 TEST 21 NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
479 -----
480
481 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
482 THE NEGD INSTRUCTION IS USED TO TEST MODE 1
483
484 TEST 22 NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
485 -----
486
487 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
488 THE ABSD INSTRUCTION IS USED TO TEST MODE 2
489
490 TEST 23 NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
491 -----
492
493 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
494 THE ABSD INSTRUCTION IS USED TO TEST MODE 4
495
496 TEST 24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
497 -----
498
499 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
500 THE ABSD INSTRUCTION IS USED TO TEST MODE 3
501
502 TEST 25 NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
503 -----
504
505 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
506 THE NEGD INSTRUCTION IS USED TO TEST MODE 5
507
508 TEST 26 NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
509 -----
510
511 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
512 THE ABSD INSTRUCTION IS USED TO TEST MODE 6
513
514 TEST 27 NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
515 -----
516
517 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
518 THE ABSD INSTRUCTION IS USED TO TEST MODE 6
519
520 TEST 30 NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
521 -----
522
523 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
524 THE NEGD INSTRUCTION IS USED TO TEST MODE 6
525
526 TEST 31 NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST

527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583

TEST 32

SPECIAL DEST, MODE 0, TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE ABSD INSTRUCTION IS USED TO TEST MODE 7

TEST 33

SPECIAL DEST, MODE 1, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 0 USING THE NEGD INSTR.

TEST 34

SPECIAL DEST, MODE 2, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 1 USING THE NEGD INSTR.

TEST 35

SPECIAL DEST, MODE 4, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 2 USING THE NEGD INSTR.

TEST 36

SPECIAL DEST, MODE 3, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 4 USING THE NEGD INSTR.

TEST 37

SPECIAL DEST, MODE 5, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 3 USING THE NEGD INSTR.

TEST 40

SPECIAL DEST, FLOATING MODE 2, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 5 USING THE NEGD INSTR.

TEST 41

SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 2 USING THE NEGF INSTR.

TEST 42

SPECIAL DEST, MODE 6, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 2(IMMEDIATE) USING THE NEGD INSTR.

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION
FLOWS MODE 6 USING THE NEGD INSTR.

584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640

TEST 43 SPECIAL DEST, MODE 7, TEST

THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS MODE 7 USING THE NEGD INSTR.

TEST 44 NEGD, ABSD AND TSTD TEST

THIS IS A TEST OF THE NEGD ABSD AND TSTD INSTRUCTIONS.

TEST 45 SOURCE MODES, MODE 1 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 1 USING THE LDFPS INSTRUCTION.

TEST 46 SOURCE MODES, MODE 2 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 2 USING THE LDFPS INSTRUCTION.

TEST 47 SOURCE MODES, MODE 4 (FL=0), TEST

THIS IS A TEST OF SOURCE MODE 4 USING THE LDFPS INSTRUCTION.

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

THIS IS A TEST OF SOURCE MODE 3 USING THE LDFPS INSTRUCTION.

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

THIS IS A TEST OF SOURCE MODE 5 USING THE LDFPS INSTRUCTION.

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

THIS IS A TEST OF SOURCE MODE 6 USING THE LDFPS INSTRUCTION.

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

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

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

641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697

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 INSTRUCTION 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.

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

698
699
700
701
702
703
704
705

THIS IS A TEST OF DESTINATION MODE 6 USING THE STFPS
INSTRUCTION

TEST 67 DESTINATION MODES, MODE 7 (FL=0), TEST

,

707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
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

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 STCDL WITH REGISTER 0

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.

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

TEST 76 D-SPACE NON-ACCESS TEST

THIS IS A TEST THAT ENABLES D-SPACE, BUT MAKES IT NON-RESIDENT, CAUSING A MEMORY MANAGEMENT TRAP SHOULD IT BE ACCESSED DURING AN INSTRUCTION THAT WILL NOT NORMALLY ACCESS D-SPACE.

TEST 77 AUTO INCREMENT/DECREMENT TEST

THIS IS A TEST THAT ENABLES D-SPACE, BUT MAKES IT NON-RESIDENT IN THE AREA OF THE TEST, FORCING A MEMORY MANAGEMENT TRAP FOR EVERY FPP INSTRUCTION IN

764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790

THE TEST. SR1 IS THEN EXAMINED FOR PROPER CONTENTS.
SHOULD THE FPP INSTRUCTION FAIL TO ABORT, THE NEXT
INSTRUCTION IS AN IOT TRAP, AND CALLS AN ERROR TO
ANNOUNCE THE FPP INSTRUCTION'S FAILING TO CAUSE AN
ABORT, NOT ALLOWING PROPER EXAMINATION OF SR1.

10.

LISTING

&

000443
000003

MNUMBER=443
PROGNUM=3

.LIST ME
.NLIST MD,MC,CND

1021 000000
1028
1029
1030
1031
1032
1033

```
.ENABL ABS  
.MCALL .HEADER, .SWRHI, .EQUAT, .SETUP, .SCATCH, .SACT11, .SCMTAG  
.MCALL .SEOP, .SSCOPE, .SEORR, .SSAVE, .STYPE, .STYPOCT  
.MCALL .STYPDEC, .STRAP, .SPOWER, .SAPTHDR, .SAPTBL  
.MCALL .SAPTYPE, .SREAD  
.MCALL .EQUIV ;REMOVE FOR PDP-10
```

```
.TITLE CKFPCBO FP11F FLTG PNT PRT C  
:*COPYRIGHT (C) 1979  
:*DIGITAL EQUIPMENT CORP.  
:*MAYNARD, MASS. 01754  
:*  
:*  
:*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
:*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.  
:*
```

000001
160000

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

1034
1035
1036
1037
1038
1039
1040
1041
1042
1043

000244
000250
177400
000200
000011
000015

```
FPVECT=244  
MMVECT=250  
$SWR=177400  
$SWRMSK=200  
TAB=11  
CRLF=15
```

001100
104000
000004

```
.SBTTL BASIC DEFINITIONS  
:*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***  
STACK= 1100  
ERROR=EMT  
SCOPE=IOT
```

000011
000012
000015
000200
177776
;77776
177774
177772
177570
177570

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

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

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

000000
000040
000100

```
:*PRIORITY LEVEL DEFINITIONS  
PR0= 0 ;:PRIORITY LEVEL 0  
PR1= 40 ;:PRIORITY LEVEL 1  
PR2= 100 ;:PRIORITY LEVEL 2
```

```
000140 PR3= 140 ;:PRIORITY LEVEL 3
000200 PR4= 200 ;:PRIORITY LEVEL 4
000240 PR5= 240 ;:PRIORITY LEVEL 5
000300 PR6= 300 ;:PRIORITY LEVEL 6
000340 PR7= 340 ;:PRIORITY LEVEL 7
```

;* 'SWITCH REGISTER' SWITCH DEFINITIONS

```
100000 SW15= 100000
040000 SW14= 40000
020000 SW13= 20000
010000 SW12= 10000
004000 SW11= 4000
002000 SW10= 2000
001000 SW09= 1000
000400 SW08= 400
000200 SW07= 200
000100 SW06= 100
000040 SW05= 40
000020 SW04= 20
000010 SW03= 10
000004 SW02= 4
000002 SW01= 2
000001 SW00= 1
```

```
SW9=SW09
SW8=SW08
SW7=SW07
SW6=SW06
SW5=SW05
SW4=SW04
SW3=SW03
SW2=SW02
SW1=SW01
SW0=SW00
```

;* DATA BIT DEFINITIONS (BIT00 TO BIT15)

```
100000 BIT15= 100000
040000 BIT14= 40000
020000 BIT13= 20000
010000 BIT12= 10000
004000 BIT11= 4000
002000 BIT10= 2000
001000 BIT09= 1000
000400 BIT08= 400
000200 BIT07= 200
000100 BIT06= 100
000040 BIT05= 40
000020 BIT04= 20
000010 BIT03= 10
000004 BIT02= 4
000002 BIT01= 2
000001 BIT00= 1
```

```
BIT9=BIT09
BIT8=BIT08
BIT7=BIT07
BIT6=BIT06
BIT5=BIT05
BIT4=BIT04
BIT3=BIT03
BIT2=BIT02
```

```

000002          BIT1=BIT01
000001          BIT0=BIT00
;*BASIC 'CPU' TRAP VECTOR ADDRESSES
000004          ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
000010          RESVEC= 10         ;; RESERVED AND ILLEGAL INSTRUCTIONS
000014          TBITVEC=14        ;; 'T' BIT
000014          TRTVEC= 14         ;; TRACE TRAP
000014          BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
000020          IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024          PWRVEC= 24         ;; POWER FAIL
000030          EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
000034          TRAPVEC=34        ;; 'TRAP' TRAP
000060          TKVEC= 60          ;; TTY KEYBOARD VECTOR
000064          TPVEC= 64          ;; TTY PRINTER VECTOR
000240          PIRQVEC=240       ;; PROGRAM INTERRUPT REQUEST VECTOR
    
```

.SBTTL FPP REGISTER DEFINITIONS

```

1044          000000          AC0          =%0
1045          000001          AC1          =%1
1046          000002          AC2          =%2
1047          000003          AC3          =%3
1048          000004          AC4          =%4
1049          000005          AC5          =%5
1050          000006          AC6          =%6
1051          000007          AC7          =%7
1052          172300          KIPDR0       =172300
1053          172302          KIPDR1       =172302
1054          172304          KIPDR2       =172304
1055          172306          KIPDR3       =172306
1056          172310          KIPDR4       =172310
1057          172316          KIPDR7       =172316
1058          172340          KIPAR0       =172340
1059          172342          KIPAR1       =172342
1060          172344          KIPAR2       =172344
1061          172346          KIPAR3       =172346
1062          172350          KIPAR4       =172350
1063          172356          KIPAR7       =172356
1064          172320          KDPDR0       =172320
1065          172322          KDPDR1       =172322
1066          172324          KDPDR2       =172324
1067          172326          KDPDR3       =172326
1068          172330          KDPDR4       =172330
1069          172336          KDPDR7       =172336
1070          172360          KDPAR0       =172360
1071          172362          KDPAR1       =172362
1072          172364          KDPAR2       =172364
1073          172366          KDPAR3       =172366
1074          172370          KDPAR4       =172370
1075          172376          KDPAR7       =172376
1076          177572          MMR0         =177572
1077          177574          SR1          =177574
1078          177576          MMR2         =177576
1079          172516          MMR3         =172516
1080          117760          DATA        =117760
1081          000020          IOTRAP       =000020
    
```

.SBTTL TRAP CATCHER

1082
1083
1085
1086

```
000000          .=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
000174          .=174
000174 000000  DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
000176 000000  SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
000200 000137 006106 .SBTTL STARTING ADDRESS(ES)
                JMP      @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

1087

.SBTTL COMMON TAGS

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

001100 001100
001100 000000
001102 000
001103 000
001104 000000
001106 000000
001110 000000
001112 000000
001114 000
001115 001
001116 000000
001120 000000
001122 000000
001124 000000
001126 000000
001130 000000
001132 000000
001134 000
001135 000
001136 000000
001140 177570
001142 177570
001144 177560
001146 177562
001150 177564
001152 177566
001154 000
001155 002
001156 012
001157 000
001160 000000

001162 000024
001164 000000
001166 000000
001170 000000
001172 000000
001174 000000
001176 000000
001200 000000
001202 000000
001204 000000
001206 000000
001210 000000
001212 000000
001214 000000
001216 000000
001220 000000
001222 000000
001224 000000
001226 000000

.=1100
\$CMTAG: .WORD 0
\$TSTNM: .BYTE 0
\$ERFLG: .BYTE 0
\$ICNT: .WORD 0
\$LPADR: .WORD 0
\$LPERR: .WORD 0
\$ERTTL: .WORD 0
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
\$DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$STPFLG: .BYTE 0
\$REGAD: .WORD 0

\$REG0: .REPT \$CM3
\$REG1: .WORD 0
\$REG2: .WORD 0
\$REG3: .WORD 0
\$REG4: .WORD 0
\$REG5: .WORD 0
\$REG6: .WORD 0
\$REG7: .WORD 0
\$REG10: .WORD 0
\$REG11: .WORD 0
\$REG12: .WORD 0
\$REG13: .WORD 0
\$REG14: .WORD 0
\$REG15: .WORD 0
\$REG16: .WORD 0
\$REG17: .WORD 0
\$REG20: .WORD 0
\$REG21: .WORD 0
\$REG22: .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 (\$REG0) WAS OBTAINED

:::CONTAINS ((\$REGAD)+0)
:::CONTAINS ((\$REGAD)+2)
:::CONTAINS ((\$REGAD)+4)
:::CONTAINS ((\$REGAD)+6)
:::CONTAINS ((\$REGAD)+10)
:::CONTAINS ((\$REGAD)+12)
:::CONTAINS ((\$REGAD)+14)
:::CONTAINS ((\$REGAD)+16)
:::CONTAINS ((\$REGAD)+20)
:::CONTAINS ((\$REGAD)+22)
:::CONTAINS ((\$REGAD)+24)
:::CONTAINS ((\$REGAD)+26)
:::CONTAINS ((\$REGAD)+30)
:::CONTAINS ((\$REGAD)+32)
:::CONTAINS ((\$REGAD)+34)
:::CONTAINS ((\$REGAD)+36)
:::CONTAINS ((\$REGAD)+40)
:::CONTAINS ((\$REGAD)+42)
:::CONTAINS ((\$REGAD)+44)

```
001230 000000 $REG23: .WORD 0 ;;CONTAINS (($REGAD)+46)
000024 .REPT 24
001232 000000 $TMP0: .WORD 0 ;;USER DEFINED
001234 000000 $TMP1: .WORD 0 ;;USER DEFINED
001236 000000 $TMP2: .WORD 0 ;;USER DEFINED
001240 000000 $TMP3: .WORD 0 ;;USER DEFINED
001242 000000 $TMP4: .WORD 0 ;;USER DEFINED
001244 000000 $TMP5: .WORD 0 ;;USER DEFINED
001246 000000 $TMP6: .WORD 0 ;;USER DEFINED
001250 000000 $TMP7: .WORD 0 ;;USER DEFINED
001252 000000 $TMP10: .WORD 0 ;;USER DEFINED
001254 000000 $TMP11: .WORD 0 ;;USER DEFINED
001256 000000 $TMP12: .WORD 0 ;;USER DEFINED
001260 000000 $TMP13: .WORD 0 ;;USER DEFINED
001262 000000 $TMP14: .WORD 0 ;;USER DEFINED
001264 000000 $TMP15: .WORD 0 ;;USER DEFINED
001266 000000 $TMP16: .WORD 0 ;;USER DEFINED
001270 000000 $TMP17: .WORD 0 ;;USER DEFINED
001272 000000 $TMP20: .WORD 0 ;;USER DEFINED
001274 000000 $TMP21: .WORD 0 ;;USER DEFINED
001276 000000 $TMP22: .WORD 0 ;;USER DEFINED
001300 000000 $TMP23: .WORD 0 ;;USER DEFINED
001302 000000 $TIMES: 0 ;;MAX. NUMBER OF ITERATIONS
001304 000000 $ESCAPE: 0 ;;ESCAPE ON ERROR ADDRESS
001306 207 377 377 $BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL
001311 000
001312 077
001313 015
001314 012 000
$QUES: .ASCII /?/ ;;QUESTION MARK
$CRLF: .ASCII <15> ;;CARRIAGE RETURN
$LF: .ASCIZ <12> ;;LINE FEED
*****
.SBTTL APT MAILBOX-ETABLE
*****
.EVEN
001316 $MAIL: ;;APT MAILBOX
001316 000000 $MSGTY: .WORD AMSGTY ;;MESSAGE TYPE CODE
001320 000000 $FATAL: .WORD AFATAL ;;FATAL ERROR NUMBER
001322 000000 $TESTN: .WORD ATESTN ;;TEST NUMBER
001324 000000 $PASS: .WORD APASS ;;PASS COUNT
001326 000000 $DEVCT: .WORD ADEVCT ;;DEVICE COUNT
001330 000000 $UNIT: .WORD AUNIT ;;I/O UNIT NUMBER
001332 000000 $MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS
001334 000000 $MSGLG: .WORD AMSGLG ;;MESSAGE LENGTH
001336 $ETABLE: ;;APT ENVIRONMENT TABLE
001336 000 $ENV: .BYTE AENV ;;ENVIRONMENT BYTE
001337 000 $ENVM: .BYTE AENVM ;;ENVIRONMENT MODE BITS
001340 000000 $SWREG: .WORD ASWREG ;;APT SWITCH REGISTER
001342 000000 $USWR: .WORD AUSWR ;;USER SWITCHES
001344 000000 $CPUOP: .WORD ACPUOP ;;CPU TYPE,OPTIONS
:*
:* BITS 15-11=CPU TYPE
:* 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
:* 11/70=06,PDQ=07,Q=10
:*
:* BIT 10=REAL TIME CLOCK
:* BIT 9=FLOATING POINT PROCESSOR
:* BIT 8=MEMORY MANAGEMENT
001346 000 $MAMS1: .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE
001347 000 $MTYP1: .BYTE AMTYP1 ;;MEM. TYPE,BLK#1
:*
MEM.TYPE BYTE -- (HIGH BYTE)
```



```

          900 NSEC CORE=001
          300 NSEC BIPOLAR=002
          500 NSEC MOS=003
001350 000000 $MADR1: .WORD AMADR1 ;;HIGH ADDRESS,BLK#1
          MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
001352 000 $MAMS2: .BYTE AMAMS2 ;;HIGH ADDRESS,M.S. BYTE
001353 000 $MTYP2: .BYTE AMTYP2 ;;MEM.TYPE,BLK#2
001354 000000 $MADR2: .WORD AMADR2 ;;MEM.LAST ADDRESS,BLK#2
001356 000 $MAMS3: .BYTE AMAMS3 ;;HIGH ADDRESS,M.S.BYTE
001357 000 $MTYP3: .BYTE AMTYP3 ;;MEM.TYPE,BLK#3
001360 000000 $MADR3: .WORD AMADR3 ;;MEM.LAST ADDRESS,BLK#3
001362 000 $MAMS4: .BYTE AMAMS4 ;;HIGH ADDRESS,M.S.BYTE
001363 000 $MTYP4: .BYTE AMTYP4 ;;MEM.TYPE,BLK#4
001364 000000 $MADR4: .WORD AMADR4 ;;MEM.LAST ADDRESS,BLK#4
001366 000000 $VECT1: .WORD AVECT1 ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370 000000 $VECT2: .WORD AVECT2 ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372 000000 $BASE: .WORD ABASE ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374 000000 $DEVM: .WORD ADEVM ;;DEVICE MAP
001376 000000 $CDW1: .WORD ACDW1 ;;CONTROLLER DESCRIPTION WORD#1
001400 000000 $CDW2: .WORD ACDW2 ;;CONTROLLER DESCRIPTION WORD#2
001402 000000 $DDW0: .WORD ADDW0 ;;DEVICE DESCRIPTOR WORD#0
001404 000000 $DDW1: .WORD ADDW1 ;;DEVICE DESCRIPTOR WORD#1
001406 000000 $DDW2: .WORD ADDW2 ;;DEVICE DESCRIPTOR WORD#2
001410 000000 $DDW3: .WORD ADDW3 ;;DEVICE DESCRIPTOR WORD#3
001412 000000 $DDW4: .WORD ADDW4 ;;DEVICE DESCRIPTOR WORD#4
001414 000000 $DDW5: .WORD ADDW5 ;;DEVICE DESCRIPTOR WORD#5
001416 000000 $DDW6: .WORD ADDW6 ;;DEVICE DESCRIPTOR WORD#6
001420 000000 $DDW7: .WORD ADDW7 ;;DEVICE DESCRIPTOR WORD#7
001422 000000 $DDW8: .WORD ADDW8 ;;DEVICE DESCRIPTOR WORD#8
001424 000000 $DDW9: .WORD ADDW9 ;;DEVICE DESCRIPTOR WORD#9
001426 000000 $DDW10: .WORD ADDW10 ;;DEVICE DESCRIPTOR WORD#10
001430 000000 $DDW11: .WORD ADDW11 ;;DEVICE DESCRIPTOR WORD#11
001432 000000 $DDW12: .WORD ADDW12 ;;DEVICE DESCRIPTOR WORD#12
001434 000000 $DDW13: .WORD ADDW13 ;;DEVICE DESCRIPTOR WORD#13
001436 000000 $DDW14: .WORD ADDW14 ;;DEVICE DESCRIPTOR WORD#14
001440 000000 $DDW15: .WORD ADDW15 ;;DEVICE DESCRIPTOR WORD#15
001442 SETEND:

```

.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 \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
 ;* EM ;;POINTS TO THE ERROR MESSAGE
 ;* DH ;;POINTS TO THE DATA HEADER
 ;* DT ;;POINTS TO THE DATA
 ;* DF ;;POINTS TO THE DATA FORMAT

1091	001442	000443			\$ERRTB:	
1093	001442	046654	073610	076056	.REPT	MNUMBER
	001450	075421			:ITEM 1	.WORD EM1,DH1,DT1,DF1
	001452	046713	073663	076076	:ITEM 2	.WORD EM2,DH2,DT2,DF2
	001460	075430			:ITEM 3	.WORD EM3,DH3,DT3,DF3
	001462	046746	073753	076120	:ITEM 4	.WORD EM4,DH4,DT4,DF4
	001470	075430			:ITEM 5	.WORD EM5,DH5,DT5,DF5
	001472	047001	074043	076142	:ITEM 6	.WORD EM6,DH6,DT6,DF6
	001500	075430			:ITEM 7	.WORD EM7,DH7,DT7,DF7
	001502	047041	074132	076164	:ITEM 10	.WORD EM10,DH10,DT10,DF10
	001510	075440			:ITEM 11	.WORD EM11,DH11,DT11,DF11
	001512	047063	074132	076212	:ITEM 12	.WORD EM12,DH12,DT12,DF12
	001520	075452			:ITEM 13	.WORD EM13,DH13,DT13,DF13
	001522	047167	074043	076142	:ITEM 14	.WORD EM14,DH14,DT14,DF14
	001530	075430			:ITEM 15	.WORD EM15,DH15,DT15,DF15
	001532	047230	074132	076164	:ITEM 16	.WORD EM16,DH16,DT16,DF16
	001540	075440			:ITEM 17	.WORD EM17,DH17,DT17,DF17
	001542	047253	074043	076142		
	001550	075430				
	001552	047314	074132	076164		
	001560	075456				
	001562	047337	074173	076212		
	001570	075452				
	001572	047337	074173	076212		
	001600	075452				
	001602	047373	074132	076164		
	001610	075456				
	001612	047414	074233	076224		
	001620	075430				
	001622	047443	074173	076212		
	001630	075452				

001632	047501	074043	076224	:ITEM 20	.WORD	EM20,DH20,DT20,DF20
001640	075430					
001642	047542	074132	076164	:ITEM 21	.WORD	EM21,DH21,DT21,DF21
001650	075456					
001652	047542	074132	076164	:ITEM 22	.WORD	EM22,DH22,DT22,DF22
001660	075456					
001662	047565	074173	076212	:ITEM 23	.WORD	EM23,DH23,DT23,DF23
001670	075452					
001672	047624	074043	076224	:ITEM 24	.WORD	EM24,DH24,DT24,DF24
001700	075430					
001702	047666	074132	076164	:ITEM 25	.WORD	EM25,DH25,DT25,DF25
001710	075456					
001712	047712	074173	076212	:ITEM 26	.WORD	EM26,DH26,DT26,DF26
001720	075452					
001722	047751	074043	076224	:ITEM 27	.WORD	EM27,DH27,DT27,DF27
001730	075430					
001732	050013	074132	076164	:ITEM 30	.WORD	EM30,DH30,DT30,DF30
001740	075456					
001742	050037	074173	076212	:ITEM 31	.WORD	EM31,DH31,DT31,DF31
001750	075452					
001752	050075	074043	076224	:ITEM 32	.WORD	EM32,DH32,DT32,DF32
001760	075430					
001762	050136	074132	076164	:ITEM 33	.WORD	EM33,DH33,DT33,DF33
001770	075456					
001772	050161	074173	076212	:ITEM 34	.WORD	EM34,DH34,DT34,DF34
002000	075452					
002002	050220	074043	076224	:ITEM 35	.WORD	EM35,DH35,DT35,DF35
002010	075430					
002012	050262	074132	076164	:ITEM 36	.WORD	EM36,DH36,DT36,DF36
002020	075456					
002022	050306	074322	076246	:ITEM 37	.WORD	EM37,DH37,DT37,DF37
002030	075470					
002032	050332	074322	076246	:ITEM 40	.WORD	EM40,DH40,DT40,DF40
002040	075470					
002042	050360	074412	076312	:ITEM 41	.WORD	EM41,DH41,DT41,DF41
002050	075511					
002052	050406	074322	076246	:ITEM 42	.WORD	EM42,DH42,DT42,DF42
002060	075470					

002062	050465	074322	076246	:ITEM 43	.WORD	EM43,DH43,DT43,DF43
002070	075470					
002072	050571	074322	076246	:ITEM 44	.WORD	EM44,DH44,DT44,DF44
002100	075470					
002102	050671	074322	076246	:ITEM 45	.WORD	EM45,DH45,DT45,DF45
002110	075470					
002112	050747	074322	076246	:ITEM 46	.WORD	EM46,DH46,DT46,DF46
002120	075470					
002122	051053	074322	076246	:ITEM 47	.WORD	EM47,DH47,DT47,DF47
002130	075470					
002132	051153	074322	076246	:ITEM 50	.WORD	EM50,DH50,DT50,DF50
002140	075470					
002142	051267	074322	076246	:ITEM 51	.WORD	EM51,DH51,DT51,DF51
002150	075470					
002152	051313	074322	076246	:ITEM 52	.WORD	EM52,DH52,DT52,DF52
002160	075470					
002162	051337	074412	076312	:ITEM 53	.WORD	EM53,DH53,DT53,DF53
002170	075470					
002172	051363	074322	076246	:ITEM 54	.WORD	EM54,DH54,DT54,DF54
002200	075470					
002202	051442	074322	076246	:ITEM 55	.WORD	EM55,DH55,DT55,DF55
002210	075470					
002212	051570	074322	076246	:ITEM 56	.WORD	EM56,DH56,DT56,DF56
002220	075470					
002222	051672	074322	076246	:ITEM 57	.WORD	EM57,DH57,DT57,DF57
002230	075470					
002232	052002	074322	076246	:ITEM 60	.WORD	EM60,DH60,DT60,DF60
002240	075470					
002242	052112	074322	076246	:ITEM 61	.WORD	EM61,DH61,DT61,DF61
002250	075470					
002252	052214	073663	076224	:ITEM 62	.WORD	EM62,DH62,DT62,DF62
002260	075430					
002262	052320	073753	076224	:ITEM 63	.WORD	EM63,DH63,DT63,DF63
002270	075430					
002272	052346	074132	076164	:ITEM 64	.WORD	EM64,DH64,DT64,DF64
002300	075440					
002302	052422	073663	076224	:ITEM 65	.WORD	EM65,DH65,DT65,DF65
002310	075430					

002312	052445	074043	076142	:ITEM 66	.WORD	EM66,DH66,DT66,DF66
002320	075430					
002322	052504	073663	076142	:ITEM 67	.WORD	EM67,DH67,DT67,DF67
002330	075430					
002332	052605	073753	076142	:ITEM 70	.WORD	EM70,DH70,DT70,DF70
002340	075430					
002342	052676	074132	076356	:ITEM 71	.WORD	EM71,DH71,DT71,DF71
002350	075532					
002352	052715	073663	076142	:ITEM 72	.WORD	EM72,DH72,DT72,DF72
002360	075430					
002362	052776	074132	076412	:ITEM 73	.WORD	EM73,DH73,DT73,DF73
002370	075532					
002372	053017	074043	076142	:ITEM 74	.WORD	EM74,DH74,DT74,DF74
002400	075430					
002402	053041	073663	076076	:ITEM 75	.WORD	EM75,DH75,DT75,DF75
002410	075430					
002412	053064	074173	076212	:ITEM 76	.WORD	EM76,DH76,DT76,DF76
002420	075452					
002422	053125	074132	076412	:ITEM 77	.WORD	EM77,DH77,DT77,DF77
002430	075532					
002432	053147	074043	076142	:ITEM 100	.WORD	EM100,DH100,DT100,DF100
002440	075430					
002442	053172	073663	076076	:ITEM 101	.WORD	EM101,DH101,DT101,DF101
002450	075430					
002452	053216	074173	076212	:ITEM 102	.WORD	EM102,DH102,DT102,DF102
002460	075452					
002462	053257	074132	076412	:ITEM 103	.WORD	EM103,DH103,DT103,DF103
002470	075532					
002472	053301	074043	076142	:ITEM 104	.WORD	EM104,DH104,DT104,DF104
002500	075430					
002502	053324	073663	076076	:ITEM 105	.WORD	EM105,DH105,DT105,DF105
002510	075430					
002512	053350	074173	076212	:ITEM 106	.WORD	EM106,DH106,DT106,DF106
002520	075452					
002522	052736	074173	076212	:ITEM 107	.WORD	EM107,DH107,DT107,DF107
002530	075452					
002532	053412	074132	076412	:ITEM 110	.WORD	EM110,DH110,DT110,DF110
002540	075532					

002542	053435	074043	076142	:ITEM 111	.WORD	EM111,DH111,DT111,DF111
002550	075430					
002552	053461	073663	076076	:ITEM 112	.WORD	EM112,DH112,DT112,DF112
002560	075430					
002562	053506	074173	076212	:ITEM 113	.WORD	EM113,DH113,DT113,DF113
002570	075452					
002572	053550	074132	076412	:ITEM 114	.WORD	EM114,DH114,DT114,DF114
002600	075532					
002602	053573	074043	076142	:ITEM 115	.WORD	EM115,DH115,DT115,DF115
002610	075430					
002612	053617	073663	076076	:ITEM 116	.WORD	EM116,DH116,DT116,DF116
002620	075430					
002622	053644	074173	076212	:ITEM 117	.WORD	EM117,DH117,DT117,DF117
002630	075452					
002632	053705	074132	076412	:ITEM 120	.WORD	EM120,DH120,DT120,DF120
002640	075532					
002642	053727	074043	076142	:ITEM 121	.WORD	EM121,DH121,DT121,DF121
002650	075430					
002652	053752	073663	076076	:ITEM 122	.WORD	EM122,DH122,DT122,DF122
002660	075430					
002662	053776	074173	076212	:ITEM 123	.WORD	EM123,DH123,DT123,DF123
002670	075452					
002672	054040	074132	076412	:ITEM 124	.WORD	EM124,DH124,DT124,DF124
002700	075532					
002702	054063	074043	076142	:ITEM 125	.WORD	EM125,DH125,DT125,DF125
002710	075430					
002712	054107	073663	076076	:ITEM 126	.WORD	EM126,DH126,DT126,DF126
002720	075430					
002722	054134	074173	076212	:ITEM 127	.WORD	EM127,DH127,DT127,DF127
002730	075452					
002732	054176	074132	076412	:ITEM 130	.WORD	EM130,DH130,DT130,DF130
002740	075532					
002742	054221	073663	076076	:ITEM 131	.WORD	EM131,DH131,DT131,DF131
002750	075430					
002752	054246	074173	076212	:ITEM 132	.WORD	EM132,DH132,DT132,DF132
002760	075452					
002762	054311	074132	076412	:ITEM 133	.WORD	EM133,DH133,DT133,DF133
002770	075532					

002772	054335	073663	076076	:ITEM 134	.WORD	EM134,DH134,DT134,DF134
003000	075430					
003002	054363	074132	076164	:ITEM 135	.WORD	EM135,DH135,DT135,DF135
003010	075456					
003012	054436	074132	076164	:ITEM 136	.WORD	EM136,DH136,DT136,DF136
003020	075456					
003022	054455	073663	076224	:ITEM 137	.WORD	EM137,DH137,DT137,DF137
003030	075430					
003032	054476	074132	076164	:ITEM 140	.WORD	EM140,DH140,DT140,DF140
003040	075456					
003042	054517	074043	076142	:ITEM 141	.WORD	EM141,DH141,DT141,DF141
003050	075430					
003052	054566	073663	076142	:ITEM 142	.WORD	EM142,DH142,DT142,DF142
003060	075430					
003062	054611	074132	076164	:ITEM 143	.WORD	EM143,DH143,DT143,DF143
003070	075456					
003072	054633	074043	076142	:ITEM 144	.WORD	EM144,DH144,DT144,DF144
003100	075430					
003102	054703	073663	076142	:ITEM 145	.WORD	EM145,DH145,DT145,DF145
003110	075430					
003112	054727	074132	076164	:ITEM 146	.WORD	EM146,DH146,DT146,DF146
003120	075456					
003122	054751	074043	076142	:ITEM 147	.WORD	EM147,DH147,DT147,DF147
003130	075430					
003132	055021	073663	076142	:ITEM 150	.WORD	EM150,DH150,DT150,DF150
003140	075430					
003142	055045	074132	076164	:ITEM 151	.WORD	EM151,DH151,DT151,DF151
003150	075456					
003152	055070	074043	076142	:ITEM 152	.WORD	EM152,DH152,DT152,DF152
003160	075430					
003162	055141	073663	076142	:ITEM 153	.WORD	EM153,DH153,DT153,DF153
003170	075430					
003172	055166	074132	076164	:ITEM 154	.WORD	EM154,DH154,DT154,DF154
003200	075456					
003202	055211	074043	076142	:ITEM 155	.WORD	EM155,DH155,DT155,DF155
003210	075430					
003212	055262	073663	076142	:ITEM 156	.WORD	EM156,DH156,DT156,DF156
003220	075430					

003222	055307	074132	076164	:ITEM 157	.WORD	EM157,DH157,DT157,DF157
003230	075456					
003232	055331	074043	076142	:ITEM 160	.WORD	EM160,DH160,DT160,DF160
003240	075430					
003242	055423	073663	076142	:ITEM 161	.WORD	EM161,DH161,DT161,DF161
003250	075430					
003252	055447	074132	076164	:ITEM 162	.WORD	EM162,DH162,DT162,DF162
003260	075456					
003262	055472	073663	076142	:ITEM 163	.WORD	EM163,DH163,DT163,DF163
003270	075430					
003272	055517	074233	076142	:ITEM 164	.WORD	EM164,DH164,DT164,DF164
003300	075430					
003302	056315	074322	076246	:ITEM 165	.WORD	EM165,DH165,DT165,DF165
003310	075470					
003312	056336	074322	076246	:ITEM 166	.WORD	EM166,DH166,DT166,DF166
003320	075470					
003322	056357	074322	076246	:ITEM 167	.WORD	EM167,DH167,DT167,DF167
003330	075470					
003332	056400	074322	076246	:ITEM 170	.WORD	EM170,DH170,DT170,DF170
003340	075470					
003342	056423	074322	076246	:ITEM 171	.WORD	EM171,DH171,DT171,DF171
003350	075470					
003352	056446	074322	076246	:ITEM 172	.WORD	EM172,DH172,DT172,DF172
003360	075470					
003362	056471	074412	076312	:ITEM 173	.WORD	EM173,DH173,DT173,DF173
003370	075511					
003372	056514	074412	076312	:ITEM 174	.WORD	EM174,DH174,DT174,DF174
003400	075511					
003402	056537	074412	076312	:ITEM 175	.WORD	EM175,DH175,DT175,DF175
003410	075511					
003412	052630	073663	076142	:ITEM 176	.WORD	EM176,DH176,DT176,DF176
003420	075430					
003422	052653	073753	076142	:ITEM 177	.WORD	EM177,DH177,DT177,DF177
003430	075430					
003432	056562	074322	076246	:ITEM 200	.WORD	EM200,DH200,DT200,DF200
003440	075470					
003442	056637	074322	076246	:ITEM 201	.WORD	EM201,DH201,DT201,DF201
003450	075470					

003452	056740	074322	076246	:ITEM 202	.WORD	EM202,DH202,DT202,DF202
003460	075470					
003462	057041	074322	076246	:ITEM 203	.WORD	EM203,DH203,DT203,DF203
003470	075470					
003472	057221	074322	076246	:ITEM 204	.WORD	EM204,DH204,DT204,DF204
003500	075470					
003502	057276	074322	076246	:ITEM 205	.WORD	EM205,DH205,DT205,DF205
003510	075470					
003512	057375	074322	076246	:ITEM 206	.WORD	EM206,DH206,DT206,DF206
003520	075470					
003522	057476	074322	076246	:ITEM 207	.WORD	EM207,DH207,DT207,DF207
003530	075470					
003532	057575	074322	076246	:ITEM 210	.WORD	EM210,DH210,DT210,DF210
003540	075470					
003542	057674	074322	076246	:ITEM 211	.WORD	EM211,DH211,DT211,DF211
003550	075470					
003552	060002	074322	076246	:ITEM 212	.WORD	EM212,DH212,DT212,DF212
003560	075470					
003562	060103	074322	076246	:ITEM 213	.WORD	EM213,DH213,DT213,DF213
003570	075470					
003572	060230	074322	076246	:ITEM 214	.WORD	EM214,DH214,DT214,DF214
003600	075470					
003602	055573	074233	076142	:ITEM 215	.WORD	EM215,DH215,DT215,DF215
003610	075430					
003612	055724	074132	076164	:ITEM 216	.WORD	EM216,DH216,DT216,DF216
003620	075456					
003622	055746	074043	076142	:ITEM 217	.WORD	EM217,DH217,DT217,DF217
003630	075430					
003632	056016	073663	076142	:ITEM 220	.WORD	EM220,DH220,DT220,DF220
003640	075430					
003642	056042	074233	076142	:ITEM 221	.WORD	EM221,DH221,DT221,DF221
003650	075430					
003652	056174	074132	076164	:ITEM 222	.WORD	EM222,DH222,DT222,DF222
003660	075456					
003662	056217	074043	076142	:ITEM 223	.WORD	EM223,DH223,DT223,DF223
003670	075430					
003672	056270	073663	076142	:ITEM 224	.WORD	EM224,DH224,DT224,DF224
003700	075430					

003702	060355	074043	076142	:ITEM 225	.WORD	EM225,DH225,DT225,DF225
003710	075547					
003712	060400	073663	076142	:ITEM 226	.WORD	EM226,DH226,DT226,DF226
003720	075547					
003722	060424	074507	076212	:ITEM 227	.WORD	EM227,DH227,DT227,DF227
003730	075557					
003732	060454	074043	076142	:ITEM 230	.WORD	EM230,DH230,DT230,DF230
003740	075547					
003742	060500	073663	076142	:ITEM 231	.WORD	EM231,DH231,DT231,DF231
003750	075547					
003752	060525	074507	076212	:ITEM 232	.WORD	EM232,DH232,DT232,DF232
003760	075557					
003762	060556	074043	076142	:ITEM 233	.WORD	EM233,DH233,DT233,DF233
003770	075547					
003772	060602	073663	076142	:ITEM 234	.WORD	EM234,DH234,DT234,DF234
004000	075547					
004002	060627	074507	076212	:ITEM 235	.WORD	EM235,DH235,DT235,DF235
004010	075557					
004012	060660	074043	076142	:ITEM 236	.WORD	EM236,DH236,DT236,DF236
004020	075547					
004022	060705	073663	076142	:ITEM 237	.WORD	EM237,DH237,DT237,DF237
004030	075547					
004032	060733	074507	076212	:ITEM 240	.WORD	EM240,DH240,DT240,DF240
004040	075557					
004042	060765	074043	076142	:ITEM 241	.WORD	EM241,DH241,DT241,DF241
004050	075547					
004052	061012	073663	076142	:ITEM 242	.WORD	EM242,DH242,DT242,DF242
004060	075547					
004062	061040	074507	076212	:ITEM 243	.WORD	EM243,DH243,DT243,DF243
004070	075557					
004072	061072	074043	076142	:ITEM 244	.WORD	EM244,DH244,DT244,DF244
004100	075547					
004102	061116	073663	076142	:ITEM 245	.WORD	EM245,DH245,DT245,DF245
004110	075547					
004112	061143	074233	076142	:ITEM 246	.WORD	EM246,DH246,DT246,DF246
004120	075547					
004122	061174	074507	076212	:ITEM 247	.WORD	EM247,DH247,DT247,DF247
004130	075557					

004132	061225	074043	076142	:ITEM 250	.WORD	EM250,DH250,DT250,DF250
004140	075547					
004142	061252	073663	076142	:ITEM 251	.WORD	EM251,DH251,DT251,DF251
004150	075547					
004152	061300	074233	076142	:ITEM 252	.WORD	EM252,DH252,DT252,DF252
004160	075547					
004162	061332	074507	076212	:ITEM 253	.WORD	EM253,DH253,DT253,DF253
004170	075557					
004172	061364	074233	076142	:ITEM 254	.WORD	EM254,DH254,DT254,DF254
004200	075547					
004202	061420	074507	076212	:ITEM 255	.WORD	EM255,DH255,DT255,DF255
004210	075557					
004212	061454	074043	076142	:ITEM 256	.WORD	EM256,DH256,DT256,DF256
004220	075547					
004222	061502	073663	076142	:ITEM 257	.WORD	EM257,DH257,DT257,DF257
004230	075547					
004232	061531	074322	076246	:ITEM 260	.WORD	EM260,DH260,DT260,DF260
004240	075563					
004242	061566	074322	076246	:ITEM 261	.WORD	EM261,DH261,DT261,DF261
004250	075563					
004252	061625	074322	076246	:ITEM 262	.WORD	EM262,DH262,DT262,DF262
004260	075563					
004262	061725	074322	076246	:ITEM 263	.WORD	EM263,DH263,DT263,DF263
004270	075563					
004272	061753	074322	076246	:ITEM 264	.WORD	EM264,DH264,DT264,DF264
004300	075563					
004302	062050	074322	076246	:ITEM 265	.WORD	EM265,DH265,DT265,DF265
004310	075563					
004312	062141	074322	076246	:ITEM 266	.WORD	EM266,DH266,DT266,DF266
004320	075563					
004322	062254	074322	076246	:ITEM 267	.WORD	EM267,DH267,DT267,DF267
004330	075563					
004332	062351	074322	076246	:ITEM 270	.WORD	EM270,DH270,DT270,DF270
004340	075563					
004342	062412	074322	076246	:ITEM 271	.WORD	EM271,DH271,DT271,DF271
004350	075563					
004352	062460	074322	076246	:ITEM 272	.WORD	EM272,DH272,DT272,DF272
004360	075563					

004362	062551	074322	076246	:ITEM 273	.WORD	EM273,DH273,DT273,DF273
004370	075604					
004372	062606	074322	076246	:ITEM 274	.WORD	EM274,DH274,DT274,DF274
004400	075604					
004402	062645	074322	076246	:ITEM 275	.WORD	EM275,DH275,DT275,DF275
004410	075604					
004412	062745	074322	076246	:ITEM 276	.WORD	EM276,DH276,DT276,DF276
004420	075604					
004422	063042	074322	076246	:ITEM 277	.WORD	EM277,DH277,DT277,DF277
004430	075604					
004432	063116	074322	076246	:ITEM 300	.WORD	EM300,DH300,DT300,DF300
004440	075604					
004442	063213	074322	076446	:ITEM 301	.WORD	EM301,DH301,DT301,DF301
004450	075625					
004452	063237	074322	076446	:ITEM 302	.WORD	EM302,DH302,DT302,DF302
004460	075625					
004462	063265	074412	076520	:ITEM 303	.WORD	EM303,DH303,DT303,DF303
004470	075651					
004472	063313	074322	076446	:ITEM 304	.WORD	EM304,DH304,DT304,DF304
004500	075625					
004502	063402	074322	076446	:ITEM 305	.WORD	EM305,DH305,DT305,DF305
004510	075625					
004512	063505	074322	076446	:ITEM 306	.WORD	EM306,DH306,DT306,DF306
004520	075625					
004522	063672	074322	076446	:ITEM 307	.WORD	EM307,DH307,DT307,DF307
004530	075625					
004532	063774	074322	076446	:ITEM 310	.WORD	EM310,DH310,DT310,DF310
004540	075625					
004542	064077	074322	076446	:ITEM 311	.WORD	EM311,DH311,DT311,DF311
004550	075625					
004552	064200	074322	076446	:ITEM 312	.WORD	EM312,DH312,DT312,DF312
004560	075625					
004562	064302	074322	076446	:ITEM 313	.WORD	EM313,DH313,DT313,DF313
004570	075625					
004572	064403	074322	076446	:ITEM 314	.WORD	EM314,DH314,DT314,DF314
004600	075625					
004602	064504	074322	076446	:ITEM 315	.WORD	EM315,DH315,DT315,DF315
004610	075625					

004612	064605	074322	076446	:ITEM 316	.WORD	EM316,DH316,DT316,DF316
004620	075625					
004622	064706	074322	076446	:ITEM 317	.WORD	EM317,DH317,DT317,DF317
004630	075625					
004632	065007	074322	076446	:ITEM 320	.WORD	EM320,DH320,DT320,DF320
004640	075625					
004642	065110	074322	076446	:ITEM 321	.WORD	EM321,DH321,DT321,DF321
004650	075625					
004652	065211	074322	076572	:ITEM 322	.WORD	EM322,DH322,DT322,DF322
004660	075675					
004662	065246	074322	076572	:ITEM 323	.WORD	EM323,DH323,DT323,DF323
004670	075675					
004672	065305	074412	076636	:ITEM 324	.WORD	EM324,DH324,DT324,DF324
004700	075716					
004702	065344	074322	076572	:ITEM 325	.WORD	EM325,DH325,DT325,DF325
004710	075675					
004712	065344	074322	076572	:ITEM 326	.WORD	EM326,DH326,DT326,DF326
004720	075675					
004722	065505	074322	076572	:ITEM 327	.WORD	EM327,DH327,DT327,DF327
004730	075675					
004732	065607	074322	076572	:ITEM 330	.WORD	EM330,DH330,DT330,DF330
004740	075675					
004742	065712	074322	076572	:ITEM 331	.WORD	EM331,DH331,DT331,DF331
004750	075675					
004752	067166	074322	076572	:ITEM 332	.WORD	EM332,DH332,DT332,DF332
004760	075675					
004762	065246	074322	076572	:ITEM 333	.WORD	EM333,DH333,DT333,DF333
004770	075675					
004772	066015	074322	076572	:ITEM 334	.WORD	EM334,DH334,DT334,DF334
005000	075675					
005002	066111	074322	076572	:ITEM 335	.WORD	EM335,DH335,DT335,DF335
005010	075675					
005012	066213	074322	076572	:ITEM 336	.WORD	EM336,DH336,DT336,DF336
005020	075675					
005022	066267	074322	076572	:ITEM 337	.WORD	EM337,DH337,DT337,DF337
005030	075675					
005032	066371	074322	076572	:ITEM 340	.WORD	EM340,DH340,DT340,DF340
005040	075675					

005042	066473	074322	076572	:ITEM 341	.WORD	EM341,DH341,DT341,DF341
005050	075675					
005052	066577	074322	076572	:ITEM 342	.WORD	EM342,DH342,DT342,DF342
005060	075675					
005062	066701	074322	076572	:ITEM 343	.WORD	EM343,DH343,DT343,DF343
005070	075675					
005072	067003	074322	076572	:ITEM 344	.WORD	EM344,DH344,DT344,DF344
005100	075675					
005102	067260	074322	076572	:ITEM 345	.WORD	EM345,DH345,DT345,DF345
005110	075675					
005112	067360	074322	076572	:ITEM 346	.WORD	EM346,DH346,DT346,DF346
005120	075675					
005122	067456	074322	076572	:ITEM 347	.WORD	EM347,DH347,DT347,DF347
005130	075737					
005132	067502	074322	076572	:ITEM 350	.WORD	EM350,DH350,DT350,DF350
005140	075737					
005142	067530	074173	076212	:ITEM 351	.WORD	EM351,DH351,DT351,DF351
005150	075557					
005152	067634	074322	076572	:ITEM 352	.WORD	EM352,DH352,DT352,DF352
005160	075737					
005162	067740	074322	076572	:ITEM 353	.WORD	EM353,DH353,DT353,DF353
005170	075737					
005172	070044	074322	076572	:ITEM 354	.WORD	EM354,DH354,DT354,DF354
005200	075737					
005202	070150	074322	076572	:ITEM 355	.WORD	EM355,DH355,DT355,DF355
005210	075737					
005212	070254	074043	076076	:ITEM 356	.WORD	EM356,DH356,DT356,DF356
005220	075547					
005222	070352	074547	076120	:ITEM 357	.WORD	EM357,DH357,DT357,DF357
005230	075547					
005232	070450	074173	076212	:ITEM 360	.WORD	EM360,DH360,DT360,DF360
005240	075557					
005242	072700	073663	076446	:ITEM 361	.WORD	EM361,DH361,DT361,DF361
005250	075547					
005252	072723	074637	076702	:ITEM 362	.WORD	EM362,DH362,DT362,DF362
005260	075760					
005262	073033	074702	076720	:ITEM 363	.WORD	EM363,DH363,DT363,DF363
005270	075766					

005272	073101	074763	076740	:ITEM 364	.WORD	EM364,DH364,DT364,DF364
005300	075452					
005302	073201	075050	076212	:ITEM 365	.WORD	EM365,DH365,DT365,DF365
005310	075452					
005312	073264	075106	076752	:ITEM 366	.WORD	EM366,DH366,DT366,DF366
005320	075775					
005322	073347	075161	077020	:ITEM 367	.WORD	EM367,DH367,DT367,DF367
005330	076017					
005332	073432	075247	077042	:ITEM 370	.WORD	EM370,DH370,DT370,DF370
005340	076027					
005342	000000	000000	000000	:ITEM 371	.WORD	EM371,DH371,DT371,DF371
005350	000000					
005352	000000	000000	000000	:ITEM 372	.WORD	EM372,DH372,DT372,DF372
005360	000000					
005362	000000	000000	000000	:ITEM 373	.WORD	EM373,DH373,DT373,DF373
005370	000000					
005372	000000	000000	000000	:ITEM 374	.WORD	EM374,DH374,DT374,DF374
005400	000000					
005402	000000	000000	000000	:ITEM 375	.WORD	EM375,DH375,DT375,DF375
005410	000000					
005412	000000	000000	000000	:ITEM 376	.WORD	EM376,DH376,DT376,DF376
005420	000000					
005422	000000	000000	000000	:ITEM 377	.WORD	EM377,DH377,DT377,DF377
005430	000000					
005432	000000	000000	000000	:ITEM 400	.WORD	EM400,DH400,DT400,DF400
005440	000000					
005442	070543	074043	076142	:ITEM 401	.WORD	EM401,DH401,DT401,DF401
005450	075547					
005452	070566	073663	076142	:ITEM 402	.WORD	EM402,DH402,DT402,DF402
005460	075547					
005462	070610	074173	076212	:ITEM 403	.WORD	EM403,DH403,DT403,DF403
005470	075557					
005472	070742	074507	076212	:ITEM 404	.WORD	EM404,DH404,DT404,DF404
005500	075557					
005502	070772	074043	076142	:ITEM 405	.WORD	EM405,DH405,DT405,DF405
005510	075547					
005512	071016	073663	076142	:ITEM 406	.WORD	EM406,DH406,DT406,DF406
005520	075547					

005522	071041	074173	076212	:ITEM 407	.WORD	EM407,DH407,DT407,DF407
005530	075557					
005532	071174	074507	076212	:ITEM 410	.WORD	EM410,DH410,DT410,DF410
005540	075557					
005542	071225	074043	076142	:ITEM 411	.WORD	EM411,DH411,DT411,DF411
005550	075547					
005552	071251	073663	076142	:ITEM 412	.WORD	EM412,DH412,DT412,DF412
005560	075547					
005562	071274	074173	076212	:ITEM 413	.WORD	EM413,DH413,DT413,DF413
005570	075557					
005572	071427	074507	076212	:ITEM 414	.WORD	EM414,DH414,DT414,DF414
005600	075557					
005602	071460	074043	076142	:ITEM 415	.WORD	EM415,DH415,DT415,DF415
005610	075547					
005612	071505	073663	076142	:ITEM 416	.WORD	EM416,DH416,DT416,DF416
005620	075547					
005622	071531	074173	076212	:ITEM 417	.WORD	EM417,DH417,DT417,DF417
005630	075557					
005632	071577	074507	076212	:ITEM 420	.WORD	EM420,DH420,DT420,DF420
005640	075557					
005642	071631	074043	076142	:ITEM 421	.WORD	EM421,DH421,DT421,DF421
005650	075547					
005652	071656	073663	076142	:ITEM 422	.WORD	EM422,DH422,DT422,DF422
005660	075547					
005662	071702	074173	076212	:ITEM 423	.WORD	EM423,DH423,DT423,DF423
005670	075557					
005672	071750	074507	076212	:ITEM 424	.WORD	EM424,DH424,DT424,DF424
005700	075557					
005702	072002	074043	076142	:ITEM 425	.WORD	EM425,DH425,DT425,DF425
005710	075547					
005712	072026	073663	076142	:ITEM 426	.WORD	EM426,DH426,DT426,DF426
005720	075547					
005722	072051	074173	076212	:ITEM 427	.WORD	EM427,DH427,DT427,DF427
005730	075557					
005732	072204	074507	076212	:ITEM 430	.WORD	EM430,DH430,DT430,DF430
005740	075557					
005742	072235	074173	076212	:ITEM 431	.WORD	EM431,DH431,DT431,DF431
005750	075557					


```

005752 072310 074043 076142 ;ITEM 432
005760 075547           :      .WORD  EM432,DH432,DT432,DF432

005762 072335 073663 076142 ;ITEM 433
005770 075547           :      .WORD  EM433,DH433,DT433,DF433

005772 072361 074173 076212 ;ITEM 434
006000 075557           :      .WORD  EM434,DH434,DT434,DF434

006002 072515 074507 076212 ;ITEM 435
006010 075557           :      .WORD  EM435,DH435,DT435,DF435

006012 072547 074173 076212 ;ITEM 436
006020 075557           :      .WORD  EM436,DH436,DT436,DF436

006022 072624 074043 076142 ;ITEM 437
006030 075547           :      .WORD  EM437,DH437,DT437,DF437

006032 072652 074043 076142 ;ITEM 440
006040 075547           :      .WORD  EM440,DH440,DT440,DF440

006042 073466 075312 077106 ;ITEM 441
006050 076050           :      .WORD  EM441,DH441,DT441,DF441

006052 073522 075360 077124 ;ITEM 442
006060 076050           :      .WORD  EM442,DH442,DT442,DF442

006062 073554 075360 077124 ;ITEM 443
006070 076050           :      .WORD  EM443,DH443,DT443,DF443
  
```

1094
1095
1096

```

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

1097

```

.SBTTL APT PARAMETER BLOCK
:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
      .$X=           ;;SAVE CURRENT LOCATION
      =24           ;;SET POWER FAIL TO POINT TO START OF PROGRAM
      200           ;;FOR APT START UP
      =44           ;;POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR       ;;POINT TO APT HEADER BLOCK
      =.$X         ;;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MEADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
  
```

```

006072
000046 000046
042776
000052 000052
000000
006072

000024 006072
000200
000044 000044
006072
006072

006072 000000
006074 001316
  
```

006076 000010
006100 000040
006102 000000
006104 000052
1098
1099
1100 006106

\$TSTM: .WORD 10 ;;RUN TIM OF LONGEST TEST
\$PASTM: .WORD 40 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
\$UNITM: .WORD 0 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
\$ETEND-\$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

START:
 .SBTTL INITIALIZE THE COMMON TAGS
 ;;CLEAR THE COMMON TAGS (\$CMTAG) AREA
 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
 CLR (R6)+ ;;CLEAR MEMORY LOCATION
 CMP #SWR,R6 ;;DONE?
 BNE -6 ;;LOOP BACK IF NO
 MOV #STACK,SP ;;SETUP THE STACK POINTER
 ;;INITIALIZE A FEW VECTORS
 MOV #SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
 MOV #340,@#IOTVEC+2 ;;LEVEL 7
 MOV #ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
 MOV #340,@#EMTVEC+2 ;;LEVEL 7
 MOV #STRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
 MOV #340,@#TRAPVEC+2;LEVEL 7
 MOV #SPWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
 MOV #340,@#PWRVEC+2 ;;LEVEL 7
 MOV \$ENDCT,\$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
 CLR \$TIMES ;;INITIALIZE NUMBER OF ITERATIONS
 CLR \$ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
 MOVB #1,\$ERMAX ;;ALLOW ONE ERROR PER TEST
 ;;INITIALIZE THE 'T-BIT' TRAP VECTOR. THEN LOAD LOCATION '\$RTRN', IN
 ;;THE 'END-OF-PASS' (\$EOP) ROUTINE, WITH A 'RTI' OR 'RTT'.
 MOV #RTRN,@#TBITVEC ;;SET 'T' BIT VECTOR TO \$RTRN
 MOV #340,@#TBITVEC+2 ;;LEVEL 7
 MOV #RTI,\$RTRN ;;SET \$RTRN TO A RTI
 MOV #65\$,@#RESVEC ;;TRY TO DO A RTT
 CLR -(SP) ;;DUMMY PS
 MOV #64\$,-(SP) ;;AND PC
 RTT ;;TRY THE RTT
 64\$: MOV #RTT,\$RTRN ;;RTT IS LEGAL--SET \$RTRN TO A RTT
 BR 66\$
 65\$: ADD #10,SP ;;RTT ILLEGAL--CLEAN OFF THE STACK
 66\$: MOV #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
 CLR \$TBIT ;;CLEAR 'T' BIT SWITCH
 MOV #,\$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
 MOV #,\$LPERR ;;SETUP THE ERROR LOOP ADDRESS
 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
 ;;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
 MOV @#ERRVEC,-(SP) ;;SAVE ERROR VECTOR
 MOV #67\$,@#ERRVEC ;;SET UP ERROR VECTOR
 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
 BNE 69\$;;BRANCH IF NO TIMEOUT TRAP OCCURRED
 ;;AND THE HARDWARE SWR IS NOT = -1
 BR 68\$;;BRANCH IF NO TIMEOUT
 67\$: MOV #68\$,(SP) ;;SET UP FOR TRAP RETURN
 RTI
 68\$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
 MOV #DISPREG,DISPLAY

006106 012706 001100
006112 005026
006114 022706 001140
006120 001374
006122 012706 001100

006126 012737 043056 000020
006134 012737 000340 000022
006142 012737 043336 000030
006150 012737 000340 000032
006156 012737 045354 000034
006164 012737 000340 000036
006172 012737 045440 000024
006200 012737 000340 000026
006206 016767 034406 034376
006214 005067 173062
006220 005067 173060
006224 112767 000001 172663

006232 012737 043042 000014
006240 012737 000340 000016
006246 012767 042356 034566
006254 012737 006302 000010
006262 005046
006264 012746 006272
006270 000006
006272 012767 000006 034542 64\$:
006300 000402
006302 062706 000010 65\$:
006306 012737 000012 000010 66\$:
006314 005067 034530
006320 012767 006320 172560
006326 012767 006326 172554

006334 013746 000004
006340 012737 006374 000004
006346 012767 177570 172564
006354 012767 177570 172560
006362 022777 177777 172550
006370 001012

006372 000403
006374 012716 006402 67\$:
006400 000002
006402 012767 000176 172530 68\$:
006410 012767 000174 172524

```
006416 012637 000004      69$:  MOV    (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
006422 005067 172676          CLR    $PASS             ;;CLEAR PASS COUNT
006426 132767 000200 172703    BITB   #APTSIZE,$ENVM    ;;TEST USER SIZE UNDER APT
006434 001403          BEQ    70$              ;;YES,USE NON-APT SWITCH
006436 012767 001340 172474    MOV    #$$SWREG,$SWR    ;;NO,USE APT SWITCH REGISTER
006444
1101  .SBTTL  TYPE PROGRAM NAME
      ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
006444 005227 177777          INC    #-1              ;;FIRST TIME?
006450 001047          BNE    71$              ;;BRANCH IF NO
006452 022737 042776 000042    CMP    #$$ENDAD,@#42    ;;ACT-11?
006460 001443          BEQ    71$              ;;BRANCH IF YES
006462 104401 006530          TYPE   ,72$           ;;TYPE ASCIZ STRING
      .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
006466 005737 000042          TST    @#42             ;;ARE WE RUNNING UNDER XXDP/ACT?
006472 001012          BNE    73$              ;;BRANCH IF YES
006474 126727 172636 000001    CMPB   $ENV,#1          ;;ARE WE RUNNING UNDER APT?
006502 001406          BEQ    73$              ;;BRANCH IF YES
006504 026727 172430 000176    CMP    $SWR,#$SWREG     ;;SOFTWARE SWITCH REG SELECTED?
006512 001005          BNE    74$              ;;BRANCH IF NO
006514 104405          GT$WR                    ;;GET SOFT-SWR SETTINGS
006516 000403          BR    74$
006520 112767 000001 172406    73$:  MOV$B  #1,$AUTOB        ;;SET AUTO-MODE INDICATOR
006526 000420          74$:  BR    71$              ;;GET OVER THE ASCIZ
006570 006570          ;;72$:  .ASCIZ <CRLF>*CKFPCBO FP11F FLTG PNT PRT C*<CRLF>
      71$:
1102
1103 006570          LOOP:
1104
1105
1106
1107
1113
1114
      ;;*****
      ;;*TEST 1          STF WITH ILLEGAL ACCUMULATOR TEST
      ;;*
      ;;*THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.
      ;;*
      ;;*****
1115 006570 000004          TST1:  SCOPE
1116 006572          0001:
006572 104413          LPERR                    ;;SET UP THE LOOP ON ERROR ADDRESS.
006574 005000          CLR    R0              ;;SET THE FPS.
1118 006576 170100          LDFPS  R0
1119
1120 006600 012737 006636 000244    MOV    #000T,@#FPVECT  ;;SET UP FOR FP TRAPS.
1121 006606 012737 006614 001236    MOV    #1$,@#$TMP2
1122
1123 006614 174007          1$:  STF    AC0,AC7          ;;THIS TEST INSTRUCTION SHOULD
1124                                     ;;CAUSE A TRAP.
1125
1126          ;;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
1127 006616          0002:
1128 006616 170200          STFPS  R0              ;;GET FPS.
1129 006620 010037 001240          MOV    R0,@#$TMP3
```

```

1130 006624 170300          STST  R0          ;GET FEC.
1131 006626 010037 001242 3$:   MOV   R0,@#$TMP4
1132 006632 104001          ERROR +1          ;STF WITH ILLEGAL ACCUMULATOR, MODE
1133                                     ;0, DIDN'T TRAP. ST 765 TO ST 537.
1134 006634 000434          BR    OOODONE
1135
1136                                     ;TRAP TO OOOT, HERE, WHEN THE EXPECTED ERROR OCCURS.
1137 006636 011600  OOOT:  MOV   (SP),R0        ;MAKE SURE THE ERROR OCCURRED
1138 006640 022700 006616  CMP   #0002,R0       ;AT THE CORRECT ADDRESS.
1139 006644 001402          BEQ   0003          ;BRANCH IF TRAP ADDRESS CORRECT.
1140 006646 000137 046254  JMP   @#FPSPUR      ;IF INCORRECT GO REPORT SPURIOUS
1141                                     ;FP TRAP.
1142
1143 006652 170204 0003:  STFPS R4          ;GET FPS.
1144 006654 170305          STST  R5          ;GET FEC.
1145 006656 010437 001240  MOV   R4,@#$TMP3    ;SAVE DATA INCASE OF ERROR.
1146 006662 010537 001242  MOV   R5,@#$TMP4
1147 006666 012702 100000  MOV   #100000,R2    ;EXPECTED FPS
1148 006672 012703 000002  MOV   #2,R3         ;EXPECTED FEC
1149 006676 010237 001244  MOV   R2,@#$TMP5
1150 006702 010337 001246  MOV   R3,@#$TMP6
1151 006706 022626          CMP   (SP)+,(SP)+   ;RESET THE STACK.
1152
1153 006710 020204          CMP   R2,R4         ;WAS FPS CORRECT?
1154 006712 001402          BEQ   0004          ;BRANCH IF YES.
1155                                     ;OTHERWISE REPORT FPS INCORRECTLY
1156 006714 104002 1$:   ERROR +2          ;SET AFTER USE OF ILLEGAL ACC.
1157 006716 000403          BR    OOODONE
1158
1159 006720 020305 0004:  CMP   R3,R5         ;WAS THE FEC CORRECT?
1160 006722 001401          BEQ   OOODONE      ;BRANCH IF CORRECT.
1161                                     ;OTHERWISE REPORT INCORRECT FEC
1162 006724 104003 1$:   ERROR +3          ;AFTER USE OF ILLEGAL ACC.
1163
1164 006726          OOODONE:
1164 006726 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?).

```

1165
 1166
 1167
 1173
 1174

```

:*****
:*TEST 2          FDST MODE 1, FLOATING MODE, TEST
:*
:*THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.
:*
:*****

```

```

1175 006730 000004  TST2:  SCOPE
1176 006732          PPP1:
1176 006732 104413  LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1177
1178 006734 012700 177777  MOV   #-1,R0        ;SET UP A BACKGROUND PATTERN IN THE
1179 006740 012701 007070  MOV   #PPPBF0,R1    ;INPUT BUFFER.

```

```

1180 006744 012702 000014
1181 006750 010021
1182 006752 077202
1183
1184 006754 012700 000200
1185 006760 170100
1186 006762 012700 007120
1187 006766 172410
1188
1189 006770 012700 007104
1190 006774 005002
1191 006776 170102
1192 007000 012737 007012 001236
1193 007006 010037 001240
1194
1195 007012 174010
1196
1197 007014 022700 007104
1198 007020 001404
1199
1200 007022 010037 001242
1201 007026 104004
1202 007030 000456
1203
1204 007032 012700 007104
1205 007036 012701 007120
1206 007042 022021
1207 007044 001031
1208 007046 022011
1209 007050 001027
1210 007052 022720 177777
1211 007056 001034
1212 007060 022710 177777
1213 007064 001031
1214 007066 000437
1215
1216 007070 177777 177777 177777
1217 007076 177777 177777 177777
1218 007104 177777 177777 177777
1219 007112 177777 177777 177777
1220 007120 123456 023456
1221 007124 034567 045671
1222
1223
1224 007130 012737 007120 001242
1225 007136 012737 007104 001240
1226 007144 104005
1227 007146 000407
1228
1229
1230 007150 012737 007120 001242
1231 007156 012737 007104 001240
1232 007164 104006
1233
1234 007166
    
```

PPP2: MOV #14,R2
 MOV R0,(R1)+
 SOB R2,PPP2
 ;SET FD MODE.
 MOV #200,R0
 LDFPS R0
 MOV #PPPTP1,R0 ;PUT TEST DATA INTO ACO.
 LDD (R0),AC0
 MOV #PPPBF1,R0 ;FDST ADDRESS.
 CLR R2 ;CLEAR THE FPS.
 LDFPS R2
 MOV #PPP3,@#\$TMP2
 MOV R0,@#\$TMP3
 PPP3: STF AC0,(R0) ;TEST INSTRUCTION.
 CMP #PPPBF1,R0 ;WAS R0 MODIFIED DURING EXECUTION?
 BEQ PPP4 ;BRANCH IF R0 NOT MODIFIED, CORRECT.
 MOV R0,@#\$TMP4 ;OTHERWISE REPORT ERROR, R0 MODIFIED.
 1\$: ERROR +4
 BR PPPDONE ;GO TO NEXT TEST.
 PPP4: MOV #PPPBF1,R0 ;CHECK THE DATA IN THE OUTPUT BUFFER.
 MOV #PPPTP1,R1
 CMP (R0)+,(R1)+
 BNE PPP10 ;BRANCH IF INCORRECT.
 CMP (R0)+,(R1)
 BNE PPP10 ;BRANCH IF INCORRECT.
 CMP #-1,(R0)+ ;WAS FLOATING MODE USED?
 BNE PPP15 ;BRANCH IF NOT.
 CMP #-1,(R0)
 BNE PPP15
 BR PPPDONE ;GO TO NEXT TEST.
 PPPBF0: .WORD -1,-1,-1,-1,-1,-1
 PPPBF1: .WORD -1,-1,-1,-1,-1,-1
 PPPTP1: .WORD 123456,23456
 .WORD 34567,45671
 ;REPORT DATA IN OUT PUT BUFFER INCORRECT.
 PPP10: MOV #PPPTP1,@#\$TMP4
 MOV #PPPBF1,@#\$TMP3
 1\$: ERROR +5 ;BAD DATA.
 BR PPPDONE
 ;REPORT FLOATING MODE NOT USED, BUT FD FAILED.
 PPP15: MOV #PPPTP1,@#\$TMP4
 MOV #PPPBF1,@#\$TMP3
 1\$: ERROR +6 ;ST 707 TO 245 INTO 244 (BUT FD).
 PPPDONE:

```

007166 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?).

1235
1236
1237
1243
1244
:*****
:*TEST 3 FDST MODE 2 TEST
:*
:*THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.
:*
:*****
TST3: SCOPE
;FIRST TEST STF.
QQQ1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1245 007170 000004
1246
1247 007172 104413
1248
1249 007174 012700 177777 MOV #-1,R0 ;SET UP THE OUTPUT BUFFER.
1250 007200 012701 007332 MOV #QQQBF0,R1
1251 007204 012702 000014 MOV #14,R2
1252 007210 010021 QQQ2: MOV R0,(R1)+
1253 007212 077202 SOB R2,QQQ2
1254
1255 007214 012700 000200 MOV #200,R0 ;SET FD MODE.
1256 007220 170100 LDFPS R0
1257 007222 012700 007362 MOV #QQQTP1,R0 ;SETUP ACO.
1258 007226 172410 LDD (R0),AC0
1259
1260 007230 012700 007346 MOV #QQQBF1,R0 ;FDST ADDRESS.
1261 007234 005002 CLR R2
1262 007236 170102 LDFPS R2 ;SET FPS.
1263 007240 012737 007246 001236 MOV #QQQ3,@#$TMP2
1264
1265 007246 174020 QQQ3: STF ACO,(R0)+ ;TEST INSTRUCTION.
1266
1267 007250 022700 007352 CMP #QQQBF1+4,R0 ;WAS R0 INCREMENTED BY 4 PROPERLY?
1268
1269 007254 001407 BEQ QQQ4 ;BRANCH IF R0 CORRECT.
1270 007256 010037 001242 MOV R0,@#$TMP4 ;REPORT R0 INCORRECT AFTER FDST MODE 2.
1271 007262 012737 007352 001240 MOV #QQQBF1+4,@#$TMP3
1272 007270 104007 1$: ERROR +7 ;BAD CONSTANT USED OR DIDN'T GO 527 TO 642
1273 007272 000526 BR QQQDONE
1274 007274 012700 007346 QQQ4: MOV #QQQBF1,R0 ;WAS THE OUTPUT DATA CORRECT?
1275 007300 012701 007362 MOV #QQQTP1,R1
1276 007304 022021 CMP (R0)+,(R1)+
1277 007306 001031 BNE QQQ10 ;BRANCH IF INCORRECT.
1278 007310 022021 CMP (R0)+,(R1)+
1279 007312 001027 BNE QQQ10 ;BRANCH IF INCORRECT.
1280 007314 022027 177777 CMP (R0)+,#-1 ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
1281 007320 001024 BNE QQQ10 ;BRANCH IF INCORRECT.
1282 007322 022027 177777 CMP (R0)+,#-1
1283 007326 001021 BNE QQQ10 ;BRANCH IF INCORRECT.
1284 007330 000430 BR QQQ20

```

```

1285 007332 177777 177777 177777 QQQBF0: .WORD -1,-1,-1,-1,-1,-1
      007340 177777 177777 177777
1286 007346 177777 177777 177777 QQQBF1: .WORD -1,-1,-1,-1,-1,-1
      007354 177777 177777 177777
1287 007362 076543 QQQTP1: 76543
1288 007364 065432      65432
1289 007366 054321      54321
1290 007370 043210      43210
1291                                     ;REPORT OUTPUT DATA INCORRECT:
1292 007372 012737 007362 001240 QQQ10: MOV #QQQTP1,@#$TMP3
1293 007400 012737 007346 001242 MOV #QQQBF1,@#$TMP4
1294 007406 104010 1$: ERROR +10 ;BAD DATA
1295 007410 000457 BR QQQDONE
1296
1297                                     ;NOW TEST STD MODE 2.
1298
1299 007412 QQQ20:
      007412 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1300 007414 012700 007332 MOV #QQQBF0,R0 ;SET UP DEFAULT INPUT DATA BUFFER.
1301 007420 010001 MOV R0,R1
1302 007422 012702 000014 MOV #14,R2
1303 007426 010021 QQQ22: MOV R0,(R1)+
1304 007430 077202 SOB R2,QQQ22
1305 007432 012700 000200 MOV #200,R0 ;ENTER FLOATING DOUBLE MODE.
1306 007436 170100 LDFPS R0
1307 007440 012700 007362 MOV #QQQTP1,R0 ;LOAD ACO.
1308 007444 172410 LDD (R0),ACO
1309 007446 012700 007346 MOV #QQQBF1,R0 ;SET DESTINATION ADDRESS.
1310 007452 012737 007460 001236 MOV #QQQ23,@#$TMP2
1311 007460 174020 QQQ23: STD ACO,(R0)+ ;TEST INSTRUCTION.
1312 007462 022700 007356 CMP #QQQBF1+10,R0 ;WAS R0 INCREMENTED BY 10 CORRECTLY?
1313 007466 001407 BEQ QQQ24 ;BRANCH IF CORRECT.
1314 007470 010037 001242 MOV R0,@#$TMP4 ;REPORT R0 INCORRECTLY INCREMENTED.
1315 007474 012737 007356 001240 MOV #QQQBF1+10,@#$TMP3
1316 007502 104011 1$: ERROR +11 ;DO NOT INCREM BY 10 BAD CONSTANT
1317 007504 000421 BR QQQDONE
1318 007506 012700 007346 QQQ24: MOV #QQQBF1,R0 ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
1319 007512 012701 007362 MOV #QQQTP1,R1
1320 007516 012702 000004 MOV #4,R2
1321 007522 022021 1$: CMP (R0)+,(R1)+
1322 007524 001002 BNE QQQ25 ;BRANCH IF INCORRECT.
1323 007526 077203 SOB R2,1$
1324 007530 000407 BR QQQDONE
1325                                     ;REPORT DATA INCORRECT.
1326 007532 012737 007362 001240 QQQ25: MOV #QQQTP1,@#$TMP3
1327 007540 012737 007346 001242 MOV #QQQBF1,@#$TMP4
1328 007546 104012 1$: ERROR +12 ;BAD DATA
1329 007550 QQQDONE:
      007550 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?).

```

1330
1336

```

:*****
:*TEST 4 FDST MODE 2, WITH GR7, TEST
:*

```

```

; *THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.
; *
; *****
TST4:  SCOPE
1337 007552 000004
1338 007554
1339 007554 104413
1339 007556 012700 007634
1340 007562 012701 007702
1341 007566 012702 000004
1342 007572 012021
1343 007574 077202
1344 007576 012700 000200
1345 007602 170100
1346 007604 012700 007712
1347 007610 172410
1348 007612 012737 007732 000004
1349 007620 012737 007632 001236
1350 007626 005001
1351 007630 005004
1352
1353
1354
1355
1356
1357
1358 007632 174027
1359 007634 005201
1360 007636 005201
1361 007640 005201
1362 007642 005201
1363 007644 012700 007722
1364 007650 012702 007634
1365 007654 012703 000004
1366 007660 022022
1367 007662 001051
1368 007664 077303
1369 007666 005704
1370 007670 001056
1371 007672 022701 000003
1372 007676 001053
1373 007700 000474
1374
1375 007702 005201
1376 007704 005201
1377 007706 005201
1378 007710 005201
1379
1380 007712 005204
1381 007714 005204
1382 007716 005204
1383 007720 005204
1384
1385 007722 005204
1386 007724 005201
1387 007726 005201
1388 007730 005201

; *****
RRR1:  LPERR                ;SET UP THE LOOP ON ERROR ADDRESS.
        MOV #RRR3,R0        ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
        MOV #RRRTP1,R1
        MOV #4,R2
1$:     MOV (R0)+,(R1)+
        SOB R2,1$
        MOV #200,R0        ;ENTER FLOATING DOUBLE MODE.
        LDFPS R0
        MOV #RRRTP2,R0     ;SET UP ACO.
        LDD (R0),AC0
        MOV #RRR10,@#ERRVECT ;SET UP FOR AN ODD ADDRESS.
        MOV #RRR2,@#$TMP2
        CLR R1
        CLR R4

;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
RRR2:  STD ACO,(R7)+        ;TEST INSTRUCTION.
RRR3:  INC R1                ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4.
        INC R1
        INC R1
        INC R1
1363:  MOV #RRREXP,R0        ;SEE IF THE DATA WAS OUTPUT CORRECTLY.
1364:  MOV #RRR3,R2
RRR4:  MOV #4,R3
        CMP (R0)+,(R2)+
        BNE RRR25           ;BRANCH IF INCORRECT.
        SOB R3,RRR4
        TST R4              ;MAKE SURE R4 IS 0.
        BNE RRR15          ;BRANCH IF R4 IS INCORRECT.
        CMP #3,R1          ;SEE IF R1 IS CORRECT.
        BNE RRR15          ;BRANCH IF R1 IS INCORRECT.
        BR RRRDONE

;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3.
RRRTP1: INC R1
        INC R1
        INC R1
        INC R1

;THIS IS THE DATA PUT IN ACO BEFORE EXECUTION OF THE STD.
RRRTP2: INC R4
        INC R4
        INC R4
        INC R4

;THIS IS THE EXPECTED DATA AT RRR3 AFTER EXECUTION OF THE STD.
RRREXP: INC R4
        INC R1
        INC R1
        INC R1

```



```

1389 ;IF A FAILURE IN THE FDST FLOWS RESULTS IN AN ODD ADDRESS TRAP THROUGH
1390 :4 TO HERE:
1391 007732 011602 RRR10: MOV (SP),R2 ;SEE IF THE TRAP WAS BECAUSE OF AN ODD ADDRESS.
1392 007734 032702 000001 BIT #1,R2
1393 007740 001005 BNE RRR11 ;BRANCH IF YES.
1394 007742 020227 007636 CMP R2,#RRR3+2 ;SEE IF THE TRAP OCCURRED AT THE TEST INSTRUCTION.
1395 007746 001412 BEQ RRR12 ;BRANCH IF YES.
1396 007750 000137 046310 JMP @#CPSPUR ;OTHERWISE REPORT A SPURIOUS TRAP THROUGH VECTOR 4.
1397 ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
1398 007754 010237 001236 RRR11: MOV R2,@#$TMP2
1399 007760 012737 007636 001240 MOV #RRR3+2,@#$TMP3
1400 007766 022626 CMP (SP)+,(SP)+
1401 007770 104013 1$: ERROR +13 ;BAD CONSTANT #2 + PC ODD ADDR.
1402 007772 000437 BR RRRDONE
1403 007774 010237 001236 RRR12: MOV R2,@#$TMP2
1404 010000 022626 CMP (SP)+,(SP)+
1405 010002 104014 1$: ERROR +14 ;ODD ADDRESS TRAP
1406 010004 000432 BR RRRDONE ;WRONG MODE USED.
1407
1408 ;REPORT DATA INCORRECT:
1409 010006 012737 007634 001240 RRR25: MOV #RRR3,@#$TMP3
1410 010014 012737 007722 001242 MOV #RRR3+2,@#$TMP4
1411 010022 104015 1$: ERROR +15 ;BAD DATA BUT GR7 FAIL
1412 010024 000422 BR RRRDONE
1413
1414 ;REPORT PC INCORRECT MODIFIED DURING THE EXECUTION OF FDST IMMEDIATE
1415 ;MODE. THE PC SHOULD HAVE BEEN INCREMENTED BY 2 BUT IT WASN'T.
1416 ;USE R1 AND R4 TO COMPUTE THE ACTUAL ACTION THAT WAS TAKEN ON THE PC.
1417 010026 012737 007636 001240 RRR15: MOV #RRR3+2,@#$TMP3
1418 010034 005704 TST R4 ;IS R4 CLEAR.
1419 010036 001404 BEQ 1$
1420 010040 012737 007634 001242 MOV #RRR3,@#$TMP4
1421 010046 000410 BR 2$
1422 010050 012702 007636 1$: MOV #RRR3+2,R2
1423 010054 062701 177775 ADD #-3,R1
1424 010060 006301 ASL R1
1425 010062 160102 SUB R1,R2
1426 010064 010237 001242 MOV R2,@#$TMP4
1427 010070 2$:
1428 010070 104016 3$: ERROR +16 ;BAD CONSTANT PC+
1429 010072 RRRDONE:
1429 010072 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?).

```

```

1430
1436
;*****
;*TEST 5 FDST MODE 4 TEST
;*
;THIS IS A TEST OF STD WITH FDST MODE 4.
;*****
1437 010074 000004 TST5: SCOPE
1438 010076 SSS1:
1438 010076 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.

```

```

1439 010100 012700 177777          MOV    #-1,R0          ;SET UP THE OUTPUT BUFFER.
1440 010104 012701 010234          MOV    #SSSBF0,R1
1441 010110 012702 000010          MOV    #10,R2
1442 010114 010021          1$:   MOV    R0,(R1)+
1443 010116 077202          SOB    R2,1$
1444 010120 012700 000200          MOV    #200,R0        ;ENTER FLOATING DOUBLE MODE.
1445 010124 170100          LDFPS R0
1446 010126 012700 010254          MOV    #SSSTP1,R0     ;SET UP ACO.
1447 010132 172410          LDD   (R0),AC0
1448 010134 012737 010274 000004  MOV    #SSS10,@#ERRVECT ;SET UP FOR A TRAP TO 4.
1449 010142 012737 010154 001236  MOV    #SSS2,@#$TMP2
1450 010150 012700 010244          MOV    #SSSA1,R0      ;SET UP THE DESTINATION ADDRESS.
1451
1452 010154 174040          SSS2: STD    AC0,-(R0)    ;TEST INSTRUCTION.
1453 010156 005201          INC    R1
1454 010160 020027 010234          CMP    R0,#SSSBF0     ;SEE IF R0 WAS DECREMENTED PROPERLY.
1455 010164 001060          BNE    SSS15          ;BRANCH IF R0 IS INCORRECT.
1456 010166 012700 010234          MOV    #SSSBF0,R0     ;WAS THE OUTPUT DATA CORRECT?
1457 010172 012701 010254          MOV    #SSSTP1,R1
1458 010176 012702 000004          MOV    #4,R2
1459 010202 022021          1$:   CMP    (R0)+,(R1)+
1460 010204 001057          BNE    SSS20          ;BRANCH IF INCORRECT.
1461 010206 077203          SOB    R2,1$
1462 010210 012700 177777          MOV    #-1,R0          ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
1463 010214 012701 010244          MOV    #SSSA1,R1
1464 010220 012702 000004          MOV    #4,R2
1465 010224 020021          2$:   CMP    R0,(R1)+
1466 010226 001056          BNE    SSS25          ;BRANCH IF INCORRECT.
1467 010230 077203          SOB    R2,2$
1468 010232 000463          BR     SSSDONE
1469
1470          ;THIS IS THE OUTPUT DATA BUFFER.
1471 010234 177777          SSSBF0: -1
1472 010236 177777          -1
1473 010240 177777          -1
1474 010242 177777          -1
1475 010244 177777          SSSA1: -1
1476 010246 177777          -1
1477 010250 177777          -1
1478 010252 177777          -1
1479
1480          ;THIS IS THE TEST DATA LOADED INTO ACO:
1481 010254 147250          SSSTP1: 147250
1482 010256 036147          36147
1483 010260 025036          25036
1484 010262 147250          147250
1485 010264 177777          SSSTP2: -1
1486 010266 177777          -1
1487 010270 177777          -1
1488 010272 177777          -1
1489
1490          ;IF AN ODD ADDRESS TRAP OCCURS COME HERE:
1491 010274 011600          SSS10: MOV    (SP),R0     ;SEE IF THE TRAP ACCURRED ON THE TEST INSTRUCTION.
1492 010276 020027 010156          CMP    R0,#SSS2+2
1493 010302 001405          BEQ   SSS11          ;BRANCH IF YES.
1494 010304 020027 010160          CMP    R0,#SSS2+4
1495 010310 001402          BEQ   SSS11          ;BRANCH IF YES.

```

```

1496 010312 000137 046310          JMP      @#CPSPUR      ;OTHERWISE GO REPORT A SPURIOUS TRAP THROUGH 4.
1497          ;REPORT FAILURE IN FDST FLOWS RESULTED IN AN ODD ADDRESS.
1498 010316 010037 001236          SSS11: MOV      RO,@#$TMP2
1499 010322 104017          2$:      ERROR    +17      ;FDST FORK X ODD AD RES.
1500 010324 000426          BR        SSSDONE
1501
1502          ;REPORT RO INCORRECTLY DECREMENTED.
1503 010326 010037 001242          SSS15: MOV      RO,@#$TMP4
1504 010332 012737 010234 001240  MOV      #SSSBF0,@#$TMP3
1505 010340 104020          1$:      ERROR    +20      ;RO NOT DECRE PROP
1506 010342 000417          BR        SSSDONE
1507
1508          ;REPORT OUTPUT DATA INCORRECT:
1509 010344 012737 010234 001240  SSS20: MOV      #SSSBF0,@#$TMP3
1510 010352 012737 010254 001242  MOV      #SSSTP1,@#$TMP4
1511 010360 104021          1$:      ERROR    +21      ;BAD DATA
1512 010362 000407          BR        SSSDONE
1513 010364 012737 010244 001242  SSS25: MOV      #SSSA1,@#$TMP4
1514 010372 012737 010264 001240  MOV      #SSSTP2,@#$TMP3
1515 010400 104022          1$:      ERROR    +22      ;DATA BAD OUTSIDE TARGET AREA
1516 010402 010402 104412          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?).

```

1517
1523

```

:*****
:*TEST 6      FDST MODE 3 TEST
:*
:*THIS IS A TEST OF FDST MODE 3 USING STD.
:*
:*****
TST6:  SCOPE

```

```

1524 010404 000004          TTT1:
1525 010406 010406 104413          LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1526 010410 012701 010526          MOV      #TTTBFO,R1      ;SET UP THE OUTPUT DATA BUFFER.
1527 010414 012700 177777          MOV      #-1,RO
1528 010420 012702 000012          MOV      #12,R2
1529 010424 010021          1$:      MOV      RO,(R1)+
1530 010426 077202          SOB      R2,1$
1531 010430 012737 010526 010542  MOV      #TTTBFO,@#TTTA2
1532 010436 012700 000200          MOV      #200,RO      ;ENTER DOUBLE FLOATING MODE.
1533 010442 170100          LDFPS    RO
1534 010444 012700 010552          MOV      #TTTTP1,RO      ;SET UP ACO.
1535 010450 172410          LDD      (RO),AC0
1536 010452 012737 010562 000004  MOV      #TTT10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
1537 010460 016737 000006 001236  MOV      TTT2,@#$TMP2
1538 010466 012700 010542          MOV      #TTTA2,RO      ;SET UP THE DESTINATION ADDRESS.
1539
1540 010472 174030          TTT2:  STD      ACO,@(RO)+      ;TEST INSTRUCTION.
1541
1542 010474 020027 010544          CMP      RO,#TTTA2+2      ;SEE IF RO WAS INCREMENTED CORRECTLY.
1543 010500 001046          BNE     TTT15      ;BRANCH IF INCORRECT.
1544 010502 012701 010526          MOV      #TTTBFO,R1      ;CHECK THE OUTPUT DATA BUFFER.
1545 010506 012702 010552          MOV      #TTTTP1,R2

```

1546 010512 012703 000004
1547 010516 022122
1548 010520 001045
1549 010522 077303
1550 010524 000452
1551
1552
1553 010526 177777
1554 010530 177777
1555 010532 177777
1556 010534 177777
1557 010536 177777
1558 010540 177777
1559 010542 010526
1560 010544 177777
1561 010546 177777
1562 010550 177777
1563 010552 101213
1564 010554 141516
1565 010556 071727
1566 010560 037475
1567
1568
1569 010562 011602
1570 010564 020227 010474
1571 010570 001405
1572 010572 020227 010476
1573 010576 001402
1574 010600 000137 046310
1575
1576
1577 010604 010237 001236
1578 010610 022626
1579 010612 104023
1580 010614 000416
1581
1582
1583 010616 010037 001242
1584 010622 012737 010544 001240
1585 010630 104024
1586 010632 000407
1587
1588
1589 010634 012737 010526 001240
1590 010642 012737 010552 001242
1591 010650 104025
1592 010652
010652 104412

```

TTT3:  MOV #4,R3
        CMP (R1)+,(R2)+
        BNE TTT20 ;BRANCH IF NOT CORRECT.
        SOB R3,TTT3
        BR TTTDONE

;THIS IS THHE OUTPUT DATA BUFFER:
TTTBFO: -1
        -1
        -1
        -1
        -1
TTTA1:  -1
TTTA2:  TTTBFO
TTTA3:  -1
        -1
        -1
TTTTP1: 101213
        141516
        71727
        37475

;TRAP THROUGH VECTOR 4 TO HERE.
TTT10: MOV (SP),R2 ;SEE IF THE TRAP ADDRESS IS THAT OF THE TEST INSTRUCTION.
        CMP R2,#TTT2+2
        BEQ TTT11 ;BRANCH IF YES.
        CMP R2,#TTT2+4
        BEQ TTT11 ;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.
TTT11: MOV R2,@#$TMP2
        CMP (SP)+,(SP)+
1$:    ERROR +23 ;BET FDST X ODD ADR
        BR TTTDONE

;REPORT R0 INCORRECT:
TTT15: MOV R0,@#$TMP4
        MOV #TTTA2+2,@#$TMP3
1$:    ERROR +24 ;R0 NOT INCREMENT PROPERLY
        BR TTTDONE

;REPORT INCORRECT OUTPUT DATA:
TTT20: MOV #TTTBFO,@#$TMP3
        MOV #TTTTP1,@#$TMP4
1$:    ERROR +25 ;BAD DATA
TTTDONE: 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?).

```

1593
1599

```

*****
;*TEST 7          FDST MODE 5 TEST
;*
;*THIS IS A TEST OF FDST MODE 5 USING STD.

```

```

: *
: *****
TST7: SCOPE
1600 010654 000004
1601 010656
1602 010656 104413
1603 010660 012701 010776
1604 010664 012700 177777
1605 010670 012702 000012
1606 010674 010021
1607 010676 077202
1608 010700 012737 010776 011010
1609 010706 012700 000200
1610 010712 170100
1611 010714 012700 011022
1612 010722 012737 011032 000004
1613 010730 016737 000006 001236
1614 010736 012700 011012
1615 010742 174050
1616 010744 020027 011010
1617 010750 001046
1618 010752 012701 010776
1619 010756 012702 011022
1620 010762 012703 000004
1621 010766 022122
1622 010770 001045
1623 010772 077303
1624 010774 000452
1625
1626
1627 010776 177777
1628 011000 177777
1629 011002 177777
1630 011004 177777
1631 011006 177777
1632 011010 010776

UUU1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #UUUBF0,R1 ;SET UP THE OUTPUT DATA BUFFER.
MOV #-1,R0
MOV #12,R2
1$: MOV R0,(R1)+
SOB R2,1$
MOV #UUUBF0,@#UUUA1
MOV #200,R0 ;ENTER DOUBLE FLOATING MODE.
LDFPS R0
MOV #UUUTP1,R0 ;SET UP ACO.
LDD (R0),ACO
MOV #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
MOV UUU2,@#$TMP2
MOV #UUUA2,R0 ;SET UP THE DESTINATION ADDRESS.
UUU2: STD ACO,@-(R0) ;TEST INSTRUCTION.
CMP R0,#UUUA2-2 ;WAS R0 DECRIMENTED PROPERLY?
BNE UUU15 ;BRANCH IF R0 IS INCORRECT.
MOV #UUUBF0,R1 ;WAS THE DATA OUTPUT CORRECTLY?
MOV #UUUTP1,R2
MOV #4,R3
UUU3: CMP (R1)+,(R2)+
BNE UUU20 ;BRANCH IF DATA IS INCORRECT.
SOB R3,UUU3
BR UUUDONE

;THIS IS THE OUTPUT DATA BUFFER
UUUBF0: -1
-1
-1
-1
-1
UUUA1: UUUBF0

```

1634 011012 177777
1635 011014 177777
1636 011016 177777
1637 011020 177777
1638 011022 020212
1639 011024 023242
1640 011026 026273
1641 011030 031323
1642
1643
1644 011032 011602
1645 011034 020227 010744
1646 011040 001405
1647 011042 020227 010746
1648 011046 001402
1649 011050 000137 046310
1650
1651 011054 010237 001236
1652 011060 022626
1653 011062 104026
1654 011064 000416
1655
1656
1657 011066 010037 001242
1658 011072 012737 011014 001240
1659 011100 104027
1660 011102 000407
1661
1662
1663 011104 012737 010776 001242
1664 011112 012737 011022 001240
1665 011120 104030

UUUA2: -1
UUUA3: -1
UUUTP1: 20212
23242
26273
031323

;IF A TRAP TO 4 OCCURS COME HERE.
UUU10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2,#UUU2+2
BEQ UUU11 ;BRANCH IF YES.
CMP R2,#UUU2+4
BEQ UUU11 ;BRANCH IF YES.
JMP @#CPSPUR ;OTHERWISE REPORT A SPURIOUS TRAP TO 4.
;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
UUU11: MOV R2,@#\$TMP2
CMP (SP)+,(SP)+
1\$: ERROR +26 ;BET FDST X ODD ADR
BR UUUDONE
;REPORT R0 INCORRECT.
UUU15: MOV R0,@#\$TMP4
MOV #UUUA2+2,@#\$TMP3
1\$: ERROR +27 ;R0 NOT INCREMENT PROPERLY
BR UUUDONE
;REPORT BAD DATA.
UUU20: MOV #UUUBFO,@#\$TMP4
MOV #UUUTP1,@#\$TMP3
1\$: ERROR +30 ;BAD DATA

1667 011122
011122 104412

UUUDONE:
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?).

1668

1676

```
*****  
*TEST 10      FDST MODE 6, INDEX MODE, TEST  
*  
*THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.  
*  
*****  
TST10: SCOPE
```

011124 000004


```

1678
1679 011126          VVV1:
      011126 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1680 011130 012700 000200          MOV #200,R0          ;ENTER DOUBLE FLOATING MODE.
1681 011134 170100          LDFPS R0
1682 011136 012701 011246          MOV #VVVBFO,R1          ;SET UP THE OUT PUT DATA BUFFER.
1683 011142 012700 177777          MOV #-1,R0
1684 011146 012702 000004          MOV #4,R2
1685 011152 010021          1$: MOV R0,(R1)+
1686 011154 077202          SOB R2,1$
1687 011156 012737 011266 000004          MOV #VVV10,@#ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR.
1688 011164 012700 011256          MOV #VVVTP1,R0          ;SET UP ACO.
1689 011170 172410          LDD (R0),AC0
1690 011172 012737 011210 001236          MOV #VVV2,@#STMP2
1691 011200 012700 003345          MOV #VVVBFO-5701,R0 ;SET UP THE DESTINATION ADDRESS.
1692 011204 012701 000001          MOV #1,R1
1693 011210 174060 005701          VVV2: STD AC0,5701(R0) ;TEST INSTRUCTION.
1694
1695 011214 020027 003345          CMP R0,#VVVBFO-5701 ;SEE IF R0 WAS MODIFIED.
1696 011220 001040          BNE VVV15 ;BRANCH IF INCORRECT.
1697 011222 012702 011246          MOV #VVVBFO,R2          ;WAS THE OUTPUT DATA CORRECT.
1698 011226 012703 011256          MOV #VVVTP1,R3
1699 011232 012704 000004          MOV #4,R4
1700 011236 022223          1$: CMP (R2)+,(R3)+
1701 011240 001037          BNE VVV20 ;BRANCH IF INCORRECT DATA.
1702 011242 077403          SOB R4,1$
1703 011244 000444          BR VVVDONE
1704 011246 177777          VVVBFO: -1
1705 011250 177777          -1
1706 011252 177777          -1
1707 011254 177777          -1
1708 011256 030313          VVVTP1: 30313
1709 011260 023334          23334
1710 011262 035363          35363
1711 011264 074041          74041
1712
1713          ;COME HERE AFTER A TRAP THROUGH VECTOR 4.
1714 011266 011602          VVV10: MOV (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
1715 011270 020227 011212          CMP R2,#VVV2+2
1716 011274 001405          BEQ VVV11 ;BRANCH IF YES.
1717 011276 020227 011214          CMP R2,#VVV2+4
1718 011302 001402          BEQ VVV11 ;BRANCH IF YES.
1719 011304 000137 046254          JMP @#FPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1720          ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
1721 011310 010237 001236          VVV11: MOV R2,@#STMP2
1722 011314 022626          CMP (SF)+,(SP)+
1723 011316 104031          1$: ERROR +31 ;FDST FORK X ODD ADD
1724 011320 000416          BR VVVDONE
1725
1726          ;REPORT R0 MODIFIED.
1727 011322 010037 001242          VVV15: MOV R0,@#STMP4
1728 011326 012737 003345 001240          MOV #VVVBFO-5701,@#STMP3
1729 011334 104032          1$: ERROR +32 ;R0 MODIFIED!
1730 011336 000407          BR VVVDONE
1731
1732          ;REPORT INCORRECT DATA.
1733 011340 012737 011246 001240          VVV20: MOV #VVVBFO,@#STMP3
  
```

1734 011346 012737 011256 001242
1735 011354 104033
1736 011356
011356 104412

```

1$:      MOV      #VVVTP1,@#$TMP4
ERROR    +33      ;BAD DATA
VVVDONE: 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?).

```

1737
1743

```

*****
*TEST 11      FDST MODE 7, INDEX DEFERRED MODE, TEST
*
*THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD.
*
*****
TST11:  SCOPE

```

1744 011360 000004
1745 011362
011362 104413
1746 011364 012700 000200
1747 011370 170100
1748 011372 012701 011510
1749 011376 012700 177777
1750 011402 012702 000004
1751 011406 010021
1752 011410 077202
1753 011412 012737 011540 000004
1754 011420 012700 011520
1755 011424 172410
1756 011426 012737 011452 001236
1757 011434 012700 003627
1758 011440 012701 000001
1759 011444 012737 011510 011530
1760 011452 174070 005701
1761
1762 011456 020027 003627
1763 011462 001044
1764 011464 012702 011510
1765 011470 012703 011520
1766 011474 012704 000004
1767 011500 022223
1768 011502 001043
1769 011504 077403
1770 011506 000450
1771 011510 177777
1772 011512 177777
1773 011514 177777
1774 011516 177777
1775 011520 041424
1776 011522 034445
1777 011524 046475
1778 011526 051525
1779 011530 177777
1780 011532 177777
1781 011534 177777
1782 011536 177777
1783

```

WWW1:
LPERR    ;SET UP THE LOOP ON ERROR ADDRESS.
MOV      #200,R0 ;ENTER DOUBLE FLOATING MODE.
LDFPS   R0
MOV      #WWWBF0,R1 ;SET UP THE OUTPUT DATA BUFFER.
MOV      #-1,R0
MOV      #4,R2
1$:      MOV      R0,(R1)+
SOB     R2,1$
MOV      #WWW10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
MOV      #WWWTP1,R0 ;SET UP ACO.
LDD     (R0),ACO
MOV      #WWW2,@#$TMP2
MOV      #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS.
MOV      #1,R1
WWW2:   MOV      #WWWBF0,@#WWWBF1
STD     ACO,@5701(R0) ;TEST INSTRUCTION.

CMP      R0,#WWWBF1-5701 ;IS R0 CORRECT?
BNE     WWW15 ;BRANCH IF INCORRECT.
MOV      #WWWBF0,R2 ;WAS THE DATA OUTPUT CORRECTLY?
MOV      #WWWTP1,R3
MOV      #4,R4
1$:      CMP      (R2)+,(R3)+
BNE     WWW20 ;BRANCH IF DATA IS INCORRECT.
SOB     R4,1$
BR      WWWDONE

WWWBF0: -1
        -1
        -1
        -1
WWWTP1: 41424
        34445
        46475
        051525
WWWBF1: -1
        -1
        -1
        -1

```

```

1784 ;TRAP THROUGH 4 TO HERE.
1785 011540 011602 WWW10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
1786 011542 020227 011454 CMP R2,#WWW2+2
1787 011546 001405 BEQ WWW11 ;BRANCH IF YES.
1788 011550 020227 011456 CMP R2,#WWW2+4
1789 011554 001402 BEQ WWW11 ;BRANCH IF YES.
1790 011556 000137 046254 JMP @#FPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1791 ;REPORT FAILURE OF FDST FORK RESULTED IN AN ODD ADDRESS TRAP TO 4.
1792 011562 010237 001236 WWW11: MOV R2,@#$TMP2
1793 011566 022626 CMP (SP)+,(SP)+
1794 011570 104034 1$: ERROR +34 ;FDST FORK X ODD ADD
1795 011572 000416 BR WWWDONE
1796
1797 ;REPORT R0 MODIFIED.
1798 011574 010037 001242 WWW15: MOV R0,@#$TMP4
1799 011600 012737 003607 001240 MOV #WWWBFO-5701,@#$TMP3
1800 011606 104035 1$: ERROR +35 ;R0 MODIFIED!
1801 011610 000407 BR WWWDONE
1802
1803 ;REPORT DATA INCORRECT
1804 011612 012737 011510 001240 WWW20: MOV #WWWBFO,@#$TMP3
1805 011620 012737 011520 001242 MOV #WWWTP1,@#$TMP4
1806 011626 104036 1$: ERROR +36 ;BAD DATA
1807 011630 WWWDONE:
011630 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?).
  
```

```

1808
1814
;*****
;TEST 12 STCFD TEST
;
;THIS IS A TEST OF THE STCFD INSTRUCTION.
;
;*****
  
```

```

011632 000004 TST12: SCOPE
1815
1816 ;AC=0
1817 011634 XXX1:
011634 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1818 011636 004767 000330 JSR PC,STCFDS
1819 011642 000000 1$: 0 ;AC
1820 011644 000000 0
1821 011646 000000 0
1822 011650 000000 0
1823 011652 000000 2$: 0 ;RES
1824 011654 000000 0
1825 011656 000000 0
1826 011660 000000 0
1827 011662 000000 3$: 0 ;ERROR RES.
1828 011664 000000 0
1829 011666 177777 -1
1830 011670 177777 -1
1831 011672 047000 4$: 47000 ;FPS BEFORE EXECUTION.
1832 011674 047004 47004 ;FPS AFTER EXECUTION.
1833 011676 177777 -1 ;FEC
  
```

```

1834 011700 147004          147004          ;ERROR FPS.
1835 011702 104042          5$: ERROR +42          ;FDL<---FDLXST 767
1836 011704 000401          BR 6$
1837 011706 104043          ERROR +43          ;BUT EZBT X ST560 TO 061 INTO 261
1838 011710          6$:
1839          ;
1840 011710          XXX2:
      011710 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1841 011712 004767 000254  JSR PC,STCFDS
1842 011716 017203          1$: 17203          ;AC
      011716 017203          142536
1843 011720 142536          142536
1844 011722 047506          47506
1845 011724 172031          172031
1846 011726 017203          2$: 17203          ;RES
      011726 017203          142536
1847 011730 142536          142536
1848 011732 000000          0
1849 011734 000000          0
1850 011736 017203          3$: 17203          ;ERROR RES.
      011736 017203          142536
1851 011740 142536          142536
1852 011742 047506          47506
1853 011744 172031          172031
1854 011746 040000          4$: 40000          ;FPS BEFORE EXECUTION.
      011746 040000          40000          ;FPS AFTER EXECUTION.
1855 011750 040000          -1          ;FEC
1856 011752 177777          -1          ;ERROR FPS.
1857 011754 177777          -1
1858 011756 104044          5$: ERROR +44          ;X11(1,0)<---0 X ST766
      011756 104044          BR 6$
1859 011760 000401          ERROR +40
1860 011762 104040          6$:
1861 011764          ;
1862          ;
1863 011764          XXX3:
      011764 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1864 011766 004767 000200  JSR PC,STCFDS
1865 011772 050717          1$: 50717          ;AC
      011772 050717          27374
1866 011774 027374          27374
1867 011776 075767          75767
1868 012000 077071          77071
1869 012002 050717          2$: 50717          ;RES
      012002 050717          27374
1870 012004 027374          27374
1871 012006 000000          0
1872 012010 000000          0
1873 012012 000000          3$: 0          ;ERROR RES.
      012012 000000          0
1874 012014 000000          0
1875 012016 000000          0
1876 012020 000000          0
1877 012022 047000          4$: 47000          ;FPS BEFORE EXECUTION.
      012022 047000          47000          ;FPS AFTER EXECUTION.
1878 012024 047000          -1          ;FEC
1879 012026 177777          -1          ;ERROR FPS.
1880 012030 174002          174002
1881 012032 104045          5$: ERROR +45          ;BUT OPIC X ST251
      012032 104045          BR 6$
1882 012034 000401          ERROR +46          ;BUT EZBT X ST421
1883 012036 104046          6$:
1884 012040          ;
1885          ;
1886 012040          XXX4:
      012040 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1887 012042 004767 000124  JSR PC,STCFDS

```

```

1888 012046 020212 1$: 20212 ;AC
1889 012050 032425 32425
1890 012052 026272 26272
1891 012054 002123 02123
1892 012056 020212 2$: 20212 ;RES
1893 012060 032425 32425
1894 012062 000000 0
1895 012064 000000 0
1896 012066 020212 3$: 20212 ;ERROR RES.
1897 012070 032425 32425
1898 012072 100000 100000
1899 012074 000000 0
1900 012076 040000 4$: 40000 ;FPS BEFORE EXECUTION.
1901 012100 040000 40000 ;FPS AFTER EXECUTION.
1902 012102 177777 -1 ;FEC
1903 012104 177777 -1 ;ERROR FPS.
1904 012106 104047 5$: ERROR +47 ;BUT FD IN ROUND X ST113
1905 012110 000401 BR 6$
1906 012112 104040 ERROR +40
1907 012114
1908
1909 012114 XXX5: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
012114 104413 JSR PC,STCFDS
1910 012116 004767 000050 121314 1$: 121314 ;AC
1911 012122 121314 151617
1912 012124 151617 101112
1913 012126 101112 131415
1914 012130 131415 2$: 121314 ;RES
1915 012132 121314 151617
1916 012134 151617 0
1917 012136 000000 0
1918 012140 000000 3$: 21314 ;ERROR RES.
1919 012142 021314 151617
1920 012144 151617 0
1921 012146 000000 0
1922 012150 000000 4$: 40000 ;FPS BEFORE EXECUTION.
1923 012152 040000 40010 ;FPS AFTER EXECUTION.
1924 012154 040010 -1 ;FEC
1925 012156 177777 -1 ;ERROR FPS.
1926 012160 177777 5$: ERROR +50 ;BUT ENBT X ST567 OR BAD SIGN ST460
1927 012162 104050 BR 6$
1928 012164 000401 ERROR +40
1929 012166 104040 BR XXXDONE
1930 012170 000535
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
  
```

```

;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
  
```

```

1944      :
1945      :
1946      :
1947      :
1948      :
1949      :
1950      :
1951      :
1952      :
1953      :
1954      :
1955      :
1956      :
1957      :
1958      :
1959      :
1960      :
1961      :
1962      :
1963      :
1964      :
1965      :
1966      :
1967 012172 012601          STCFDS: MOV      (SP)+,R1          ;PICK UP THE POINTER TO THE OPERANDS.
1968 012174 012700 000200  MOV      #200,R0          ;ENTER DOUBLE FLOATING MODE.
1969 012200 170100          LDFPS   R0
1970 012202 010100          MOV      R1,R0           ;LOAD ACO.
1971 012204 172410          LDD     (R0),ACO
1972 012206 012700 177777  MOV      #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
1973 012212 012702 012454  MOV      #STCFT,R2
1974 012216 012703 000004  MOV      #4,R3
1975 012222 010022          1$: MOV      R0,(R2)+
1976 012224 077302          SOB     R3,1$
1977 012226 016100 000030  MOV      30(R1),R0       ;LOAD THE FPS.
1978 012232 170100          LDFPS   R0
1979 012234 012737 012246 001236  MOV      #2$,@#$TMP2
1980 012242 012700 012454  MOV      #STCFT,R0       ;SET UP THE DESTINATION ADDRESS.
1981 012246 176010          2$: STCFD   ACO,(R0)       ;TEST INSTRUCTION.
1982
1983 012250 170204          STFPS   R4               ;GET THE FPS.
1984 012252 170305          STST   R5               ;GET THE FEC.
1985 012254 010102          MOV      R1,R2          ;SAVE THE DATA IN CASE OF ERROR.
1986 012256 010237 001240  MOV      R2,@#$TMP3
1987 012262 062702 000010  ADD     #10,R2
1988 012266 010237 001244  MOV      R2,@#$TMP5
1989 012272 012737 012454 001242  MOV      #STCFT,@#$TMP4
1990 012300 010437 001250  MOV      R4,@#$TMP7
1991 012304 016137 000032 001252  MOV      32(R1),@#$TMP10
1992
1993 012312 010102          MOV      R1,R2          ;CHECK THE RESULT.
1994 012314 062702 000010  ADD     #10,R2
1995 012320 012703 012454  MOV      #STCFT,R3
1996 012324 012700 000004  MOV      #4,R0
1997 012330 022223          3$: CMP     (R2)+,(R3)+
1998 012332 001014          BNE    15$              ;BRANCH IF INCORRECT.
1999 012334 077003          SOB     R0,3$
2000

```

```

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 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.

```

```

2001 012336 016102 000032      MOV      32(R1),R2
2002 012342 020204      CMP      R2,R4      ;IS THE FPS CORRECT?
2003 012344 001025      BNE      20$        ;BRANCH IF FPS INCORRECT.
2004 012346 005702      TST      R2        ;IF EXPECTED FPS IS NEGATIVE, THEN
2005 012350 100003      BPL      4$        ;GO AHEAD AND CHECK THE FEC.
2006 012352 026105 000036      CMP      36(R1),R5
2007 012356 001027      BNE      25$        ;BRANCH IF FEC IS INCORRECT.
2008 012360 000161 000046      4$:      JMP      46(R1)    ;RETURN.
2009
2010      ;RESULT INCORRECT:
2011 012364 010102      15$:     MOV      R1,R2      ;SEE IF ERROR WAS ANTICIPATED.
2012 012366 062702 000020      ADD      #20,R2
2013 012372 012703 012454      MOV      #STCFT,R3
2014 012376 012700 000004      MOV      #4,R0
2015 012402 022223      16$:     CMP      (R2)+,(R3)+
2016 012404 001003      BNE      17$        ;BRANCH IF NOT ANTICIPATED.
2017 012406 077003      SOB      R0,16$
2018 012410 000161 000040      JMP      40(R1)    ;IF ERROR WAS ANTICIPATED RETURN.
2019      ;OTHERWISE REPORT RESULT INCORRECT HERE.
2020 012414
2021 012414 104037      17$:
2021 012414 104037      18$:     ERROR  +37      ;DATA ERROR
  
```

```

2023 012416 000760 BR 4$
2024
2025 :FPS INCORRECT:
2026 012420 020461 000034 20$: CMP R4,34(R1) ;WAS THE ERROR ANTICIPATED.
2027 012424 001002 :BNE 21$ ;BRANCH IF NOT ANTICIPATED.
2028 012426 000161 000044 :JMP 44(R1) ;IF IT WAS ANTICIPATED RETURN.
2029
2030 :THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
2031 012432 21$:
2032 012432 104040 22$: ERROR +40 ;FPS X
2033 012434 000751 BR 4$
2034
2035 :REPORT FEC INCORRECT:
2036 012436 016137 000036 001256 25$: MOV 36(R1),@#$TMP12
2037 012444 010537 001254 :MOV R5,@#$TMP11
2038 012450 104041 26$: ERROR +41 ;FEC X
2039 012452 000742 :BR 4$
2040 012454 177777 177777 STCFT: -1,-1,-1,-1
2041 012464 177777
012464 104412 XXXDONE: 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?).

```

2042
2048

```

:*****
:*TEST 13 STCDF TEST
:*
:*THIS IS A TEST OF THE STCDF INSTRUCTION.
:*
:*****
TST13: SCOPE

```

```

012466 000004
2049
2050 :AC=0
2051 012470 YYY1:
012470 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2052 012472 004767 000330 JSR PC,STCDFS
2053 012476 000000 1$: 0 ;AC
2054 012500 000000 0
2055 012502 000000 0
2056 012504 000000 2$: 0 ;RES
2057 012506 000000 0
2058 012510 000000 0
2059 012512 177777 -1
2060 012514 177777 -1
2061 012516 000000 3$: 0 ;ERROR RES.
2062 012520 000000 0
2063 012522 000000 0
2064 012524 000000 0
2065 012526 047200 4$: 47200 ;FPS BEFORE EXECUTION.
2066 012530 047204 47204 ;FPS AFTER EXECUTION.
2067 012532 177777 -1 ;FEC
2068 012534 177777 -1 ;ERROR FPS.
2069 012536 104054 5$: ERROR +54 ;FDFL<---FDFL X ST767
2070 012540 000401 BR 6$
2071 012542 104052 ERROR +52 ;FPS INCORRECT.

```



```

2072 012544      6$:
2073      :
2074 012544      YYY2:
      012544 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2075 012546 004767 000254      JSR      PC,STCDF5
2076 012552 067574      1$:      67574      ;ACO
2077 012554 073727      73727
2078 012556 170777      170777
2079 012560 067574      67574
2080 012562 067574      2$:      67574      ;RES
2081 012564 073730      73730
2082 012566 177777      -1
2083 012570 177777      -1
2084 012572 067574      3$:      67574      ;ERROR RES.
2085 012574 073727      73727
2086 012576 177777      -1
2087 012600 177777      -1
2088 012602 040200      4$:      40200      ;FPS BEFORE EXECUTION.
2089 012604 040200      40200      ;FPS AFTER EXECUTION.
2090 012606 177777      -1      ;FEC
2091 012610 177777      -1      ;ERROR FPS.
2092 012612 104055      5$:      ERROR      +55      ;EITHER ROUND FAILED OR WENT TO 766 X1(1,0)----0 INTO 767
2093 012614 000401      BR      6$
2094 012616 104052      ERROR      +52
2095 012620      6$:
2096      :
2097 012620      YYY3:
      012620 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2098 012622 004767 000200      JSR      PC,STCDF5
2099 012626 077777      1$:      77777      ;ACO
2100 012630 177777      -1
2101 012632 100000      100000
2102 012634 000000      0
2103 012636 000000      2$:      0      ;RES
2104 012640 000000      0
2105 012642 177777      -1
2106 012644 177777      -1
2107 012646 077777      3$:      77777      ;ERROR RES.
2108 012650 177777      -1
2109 012652 177777      -1
2110 012654 177777      -1
2111 012656 040200      4$:      40200      ;FPS BEFORE EXECUTION.
2112 012660 040206      40206      ;FPS AFTER EXECUTION.
2113 012662 177777      -1      ;FEC
2114 012664 040204      40204      ;ERROR FPS.
2115 012666 104055      5$:      ERROR      +55
2116 012670 000401      BR      6$
2117 012672 104056      ERROR      +56      ;BUT EZBT X ST421 TO 062 INTO 262
2118 012674      6$:
2119      :
2120 012674      YYY4:
      012674 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2121 012676 004767 000124      JSR      PC,STCDF5
2122 012702 077777      1$:      77777      ;ACO
2123 012704 177777      -1
2124 012706 100000      100000
2125 012710 000000      0
  
```

```

2126 012712 000000 2$: 0 ;RES
2127 012714 000000 0
2128 012716 177777 -1
2129 012720 177777 -1
2130 012722 077777 3$: 77777 ;ERROR RES.
2131 012724 177777 -1
2132 012726 177777 -1
2133 012730 177777 -1
2134 012732 040200 4$: 40200 ;FPS BEFORE EXECUTION.
2135 012734 040206 ;FPS AFTER EXECUTION.
2136 012736 177777 -1 ;FEC
2137 012740 140206 ;ERROR FPS.
2138 012742 104055 5$: ERROR +55
2139 012744 000401 BR 6$
2140 012746 104057 ERROR +57 ;BUT FIV ST262 TO 123 INTO 103
2141 012750
2142
2143 012750 ;YYY5:
012750 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
012752 004767 JSR PC,STCDFS
2144 012756 177777 1$: 177777 ;ACO
2145 012760 177777 -1
2146 012762 100000 100000
2147 012764 000000 0
2148 012766 100000 2$: 100000 ;RES
2149 012770 000000 0
2150 012772 177777 -1
2151 012774 177777 -1
2152 012776 000000 3$: 0 ;ERROR RES.
2153 013000 000000 0
2154 013002 177777 -1
2155 013004 177777 -1
2156 013006 047200 4$: 47200 ;FPS BEFORE EXECUTION.
2157 013010 147216 ;FPS AFTER EXECUTION.
2158 013012 000010 10 ;FEC
2159 013014 047206 ;ERROR FPS.
2160 013016 104060 5$: ERROR +60 ;BUT FIV ST262 FAIL TO 103 INT 123
2161 013020 000401 BR 6$
2162 013022 104061 ERROR +61 ;BUT FLAG ST 147 X TO ST 361 INTO 365
2163 013024 000535 6$: BR YYYDONE
2164
2165 ;THIS SUBROUTINE, STCDFS, IS USED TO SET UP THE OPERANDS, EXECUTE
2166 ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
2167 ;TO IT IS MADE THUS:
2168 :
2169 : JSR PC,@#STCDFS
2170 : ACARG: .WORD X,X,X,X ;AC OPERAND
2171 : RES: .WORD X,X,X,X ;EXPECTED RESULT
2172 : ERRES: .WORD X,X,X,X ;ERROR RESULT
2173 : FPSB: .WORD X ;FPS BEFORE EXECUTION
2174 : FPSA: .WORD X ;FPS AFTER EXECUTION
2175 : FEC: .WORD X ;EXPECTED FEC
2176 : ERFPS: .WORD X ;ERROR FPS.
2177 : ERR1: ERROR +X ;DATA ERROR.
2178 : BR CONT
2179 : ERR2: ERROR +X ;FPS ERROR.
2180 : CONT: ;RETURN ADDRESS
2181 :

```

```

2182 ;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
2183 ;THE STCFD INSTRUCTION IS EXECUTED.
2184 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
2185 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
2186 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
2187 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
2188 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
2189 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
2190 ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
2191 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
2192 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
2193 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
2194 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
2195 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
2196
2197 013026 012601 STCFDS: MOV (SP)+,R1 ;PICK UP THE POINTER TO THE OPERANDS.
2198 013030 012700 000200 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE.
2199 013034 170100 LDFPS R0
2200 013036 010100 MOV R1,R0 ;LOAD ACO.
2201 013040 172410 LDD (R0),ACO
2202 013042 012700 177777 MOV #-1,R0 ;FILL THE OUTPUT BUFFER WITH -1'S.
2203 013046 012702 013310 MOV #STCDT,R2
2204 013052 012703 000004 MOV #4,R3
2205 013056 010022 1$: MOV R0,(R2)+
2206 013060 077302 SOB R3,1$
2207 013062 016100 000030 MOV 30(R1),R0 ;LOAD THE FPS.
2208 013066 170100 LDFPS R0
2209 013070 012737 013102 001236 MOV #2$,@#$TMP2
2210 013076 012700 013310 MOV #STCDT,R0 ;SET UP THE DESTINATION ADDRESS.
2211 013102 176010 2$: STCDF ACO,(R0) ;TEST INSTRUCTION.
2212
2213 013104 170204 STFPS R4 ;GET THE FPS.
2214 013106 170305 STST R5 ;GET THE FEC.
2215 013110 010102 MOV R1,R2 ;SAVE THE DATA IN CASE OF ERROR.
2216 013112 010237 001240 MOV R2,@#$TMP3
2217 013116 062702 000010 ADD #10,R2
2218 013122 010237 001244 MOV R2,@#$TMP5
2219 013126 012737 013310 001242 MOV #STCDT,@#$TMP4
2220 013134 010437 001250 MOV R4,@#$TMP7
2221 013140 016137 000032 001252 MOV 32(R1),@#$TMP10
2222
2223 013146 010102 MOV R1,R2 ;CHECK THE RESULT.
2224 013150 062702 000010 ADD #10,R2
2225 013154 012703 013310 MOV #STCDT,R3
2226 013160 012700 000004 MOV #4,R0
2227 013164 022223 3$: CMP (R2)+,(R3)+
2228 013166 001014 BNE 15$ ;BRANCH IF INCORRECT.
2229 013170 077003 SOB R0,3$
2230
2231 013172 016102 000032 MOV 32(R1),R2
2232 013176 020204 CMP R2,R4 ;IS THE FPS CORRECT?
2233 013200 001025 BNE 20$ ;BRANCH IF FPS INCORRECT.
2234 013202 005702 TST R2 ;IF EXPECTED FPS IS NEGATIVE, THEN
2235 013204 100003 BPL 4$ ;GO AHEAD AND CHECK THE FEC.
2236 013206 026105 000034 CMP 34(R1),R5
2237 013212 001027 BNE 25$ ;BRANCH IF FEC IS INCORRECT.
2238 013214 000161 000046 4$: JMP 46(R1) ;RETURN.

```

```

2239
2240      ;RESULT INCORRECT:
2241 013220 010102      15$:  MOV    R1,R2      ;SEE IF ERROR WAS ANTICIPATED.
2242 013222 062702 000020      ADD    #20,R2
2243 013226 012703 013310      MOV    #STCDT,R3
2244 013232 012700 000004      MOV    #4,R0
2245 013236 022223      16$:  CMP    (R2)+,(R3)+
2246 013240 001003      BNE   17$      ;BRANCH IF NOT ANTICIPATED.
2247 013242 077003      SOB   R0,16$
2248 013244 000161 000040      JMP   40(R1)   ;IF ERROR WAS ANTICIPATED RETURN.
2249      ;OTHERWISE REPORT RESULT INCORRECT HERE.
2250 013250      17$:
2251 013250 104051      18$:  ERROR  +51      ;DATA ERROR
2252 013252 000760      BR    4$
2253
2254      ;FPS INCORRECT:
2255 013254 020461 000034      20$:  CMP    R4,34(R1)      ;WAS THE ERROR ANTICIPATED.
2256 013260 001002      BNE   21$      ;BRANCH IF NOT ANTICIPATED.
2257 013262 000161 000044      JMP   44(R1)   ;IF IT WAS ANTICIPATED RETURN.
2258
2259      ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
2260 013266      21$:
2261 013266 104052      22$:  ERROR  +52      ;FPS X
2262 013270 000751      BR    4$
2263
2264      ;REPORT FEC INCORRECT:
2265 013272 016137 000036 001256      25$:  MOV    36(R1),@#$TMP12
2266 013300 010537 001254      MOV    R5,@#$TMP11
2267 013304 104053      26$:  ERROR  +53      ;FEC X
2268 013306 000742      BR    4$
2269 013310 177777 177777 177777      STCDT: -1,-1,-1,-1
2270 013320      YYYDONE:
2270 013320 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?).
2276      ;*****
                ;*TEST 14      STCFD WITH ILLEGAL ACCUMULATOR TEST
                ;*
                ;*THIS TEST STCFD WITH ILLEGAL AC 6.
                ;*
                ;*****
                TST14: SCOPE
2277
2278 013324      ZZZ1:
2278 013324 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2279 013326 012700 040000      MOV    #40000,R0      ;DISSABLE INTERRUPTS.
2280 013332 170100      LDFPS   R0
2281 013334 012737 013342 001236      MOV    #ZZZ2,@#$TMP2
2282 013342 176006      ZZZ2:  STCFD   AC0,AC6      ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.
2283
2284 013344 170204      STFPS   R4      ;GET FPS.
2285 013346 170305      STST   R5      ;GET FEC.
2286 013350 020427 140000      CMP    R4,#140000      ;IS FPS CORRECT?
2287 013354 001004      BNE   ZZZ10      ;BRANCH IF INCORRECT FPS.

```

```

2288 013356 022705 000002          CMP      #2,R5          ;IS FEC CORRECT?
2289 013362 001010          BNE      ZZZ15         ;BRANCH IF INCORRECT.
2290 013364 000415          BR       ZZZDONE
2291
2292          ;REPORT FPS INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
2293 013366 010437 001242          ZZZ10:  MOV      R4,@#$TMP4
2294 013372 012737 140000 001240      MOV      #140000,@#$TMP3
2295 013400 104062          1$:      ERROR    +62          ;BUT FDST ST767 X TO 567 INTO 577
2296 013402 000406          BR       ZZZDONE
2297
2298          ;REPORT FEC INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
2299 013404 010537 001242          ZZZ15:  MOV      R5,@#$TMP4
2300 013410 012737 000002 001240      MOV      #2,@#$TMP3
2301 013416 104063          1$:      ERROR    +63          ;FEC<---2 ST577 X
2302 013420          ZZZDONE: 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?).

```

2303
2309

```

:*****
:*TEST 15      CLRD TEST
:*
:*THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.
:*
:*****

```

```

2310 013422 000004          TST15:  SCOPE
                                AAB1:
                                LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
                                MOV      #AABTP1,R0      ;SET UP OUTPUT BUFFER
2311 013424 104413          MOV      #AABBF0,R1
2312 013426 012700 013612      MOV      #4,R2
2313 013432 012701 013602      MOV      #4,R2
2314 013436 012702 000004          1$:      MOV      (R0)+,(R1)+
2315 013442 012021          SOB      R2,1$
2316 013444 077202          MOV      #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS.
2317 013446 012700 013602      MOV      #213,R1          ;SET UP FPS.
2318 013452 012701 000213      LDFPS   R1
2319 013456 170101          MOV      #2$,@#$TMP2
2320 013460 012737 013466 001236      2$:      CLRD      (R0)          ;TEST INSTRUCTION.
2321 013466 170410
2322 013470 170205          STFPS   R5          ;GET FPS.
2323 013472 012702 000004          MOV      #4,R2          ;SEE IF RESULT CLEAR, 0.
2324 013476 012701 013602      MOV      #AABBF0,R1
2325 013502 005721          3$:      TST      (R1)+
2326 013504 001010          BNE      AAB2          ;BRANCH IF RESULT INCORRECT, NOT 0.
2327 013506 077203          SOB      R2,3$
2328 013510 022705 000204          CMP      #204,R5        ;SEE IF FPS IS CORRECT.
2329 013514 001014          BNE      AAB3          ;BRANCH IF INCORRECT.
2330 013516 020027 013602      CMP      R0,#AABBF0     ;SEE IF R0 IS CORRECT.
2331 013522 001020          BNE      AAB4          ;BRANCH IF R0 IS INCORRECT.
2332 013524 000442          BR       AABDONE
2333
2334          ;RESULT NOT 0, REPORT ERROR.
2335 013526 012737 013602 001240      AAB2:  MOV      #AABBF0,@#$TMP3
2336 013534 012737 013622 001242      MOV      #AABTP2,@#$TMP4
2337 013542 104064          1$:      ERROR    +64          ;BAD DATA = 0 X 11+ZERO ST770 X

```

```

2338 013544 000432 BR AABDONE
2339
2340 ;REPORT FPS INCORRECT:
2341 013546 010437 001242 AAB3: MOV R4,@#$TMP4
2342 013552 012737 000204 001240 MOV #204,@#$TMP3
2343 013560 104065 1$: ERROR +65 ;BAD FPS
2344 013562 000423 BR AABDONE
2345
2346 ;REPORT R0 INCORRECT.
2347 013564 010037 001242 AAB4: MOV R0,@#$TMP4
2348 013570 012737 013602 001240 MOV #AABBF0,@#$TMP3
2349 013576 104066 1$: ERROR +66
2350 013600 000414 BR AABDONE
2351
2352 ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
2353 013602 073475 AABBF0: 73475
2354 013604 067707 67707
2355 013606 127347 127347
2356 013610 056770 56770
2357 ;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
2358 013612 073475 AABTP1: 73475
2359 013614 067707 67707
2360 013616 127347 127347
2361 013620 056770 56770
2362 ;THIS IS THE EXPECTED DATA, RESULT:
2363 013622 000000 AABTP2: 0
2364 013624 000000 0
2365 013626 000000 0
2366 013630 000000 0
2367 013632 AABDONE: RSETUP
013632 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?).
  
```

```

2368
2374
;*****
;*TEST 16 CLR D WITH ILLEGAL ACCUMULATOR TEST
;*
;*THIS IS A TEST OF CLR D WITH ILLEGAL AC7.
;*
;*****
  
```

```

2375 013634 000004 TST16: SCOPE
013636 CCB1:
013636 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2376 013640 012700 040200 MOV #40200,R0 ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
2377 013644 170100 LDFPS R0
2378 013646 012737 013654 001236 CCB2: MOV #CCB2,@#$TMP2
2379 013654 170407 CCB2: CLR D AC7 ;TEST INSTRUCTION.
2380
2381 013656 170204 STFPS R4 ;GET FPS.
2382 013660 170305 STST R5 ;GET FEC.
2383 013662 020427 140200 CMP R4,#140200 ;IS THE FPS CORRECT?
2384 013666 001004 BNE CCB10 ;BRANCH IF FPS IS INCORRECT.
2385 013670 022705 000002 CMP #2,R5 ;IS THE FEC CORRECT?
2386 013674 001010 BNE CCB15 ;BRANCH IF FEC IS INCORRECT.
2387 013676 000415 BR CCBDONE
  
```

```

2388
2389
2390 013700 010437 001242
2391 013704 012737 140200 001240
2392 013712 104067
2393 013714 000406
2394
2395
2396 013716 010537 001242
2397 013722 012737 000002 001240
2398 013730 104070
2399 013732
013732 104412

```

```

:REPORT INCORRECT FPS:
CCB10: MOV R4,@#$TMP4
MOV #140200,@#$TMP3
1$: ERROR +67 ;BUT FDST ST 700X TO 607 INTO 677
BR CCBDONE

```

```

2395
2396 013716 010537 001242
2397 013722 012737 000002 001240
2398 013730 104070
2399 013732
013732 104412

```

```

:REPORT INCORRECT FEC:
CCB15: MOV R5,@#$TMP4
MOV #2,@#$TMP3
1$: 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?).

```

```

2408
2409
2410

```

```

;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
:*****
:*TEST 17 SEE ABOVE COMMENT FOR TEST TITLE
:*
:*THIS IS A TEST OF THE SPECIAL
:*DEST FLOWS USING THE NEG D INST
:*WITH MODE ZERO AND ILLEGAL
:*AC7.
:*
:*****

```

```

013734 000004
2411
2412 013736
013736 104413
2413 013740 012700 040200
2414 013744 170100
2415 013746 012737 013754 001236
2416
2417 013754 170707
2418
2419 013756 170204
2420 013760 170305
2421
2422 013762 022704 140200
2423 013766 001004
2424 013770 022705 000002
2425 013774 001010
2426 013776 000415
2427
2428
2429 014000 012737 140200 001240
2430 014006 010437 001242
2431 014012 104176
2432 014014 000406
2433
2434
2435 014016 012737 000002 001240
2436 014024 010537 001242
2437 014030 104177

```

```

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,@#$TMP2
VVB2: NEG D 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

```

```

2428
2429 014000 012737 140200 001240
2430 014006 010437 001242
2431 014012 104176
2432 014014 000406
2433
2434
2435 014016 012737 000002 001240
2436 014024 010537 001242
2437 014030 104177

```

```

:REPORT INCORRECT FPS:
VVB10: MOV #140200,@#$TMP3
MOV R4,@#$TMP4
1$: ERROR +176 ;FPS BAD
BR VVBDONE

```

```

2434
2435 014016 012737 000002 001240
2436 014024 010537 001242
2437 014030 104177

```

```

:REPORT FEC INCORRECT:
VVB15: MOV #2,@#$TMP3
MOV R5,@#$TMP4
1$: ERROR +177 ;FEC BAD

```

2438
 2439 014032 104412

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?).

2440
 2448
 2449

;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 0 TEST

::*****
 ;*TEST 20 SEE ABOVE COMMENT FOR TEST TITLE

::*
 ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
 ;*SOURCE FLOWS. THE NEGD INSTRUCTION
 ;*IS USED TO TEST MODE 0

::*****
 TST20: SCOPE

2450 014034 000004
 2451 014036
 2452 014036 104413
 2453 014040 012700 000200
 2454 014044 170100
 2455 014046 012700 014210
 2456 014052 172410
 2457 014054 005000
 2458 014056 170100
 2459 014060 012700 014220
 2460 014064 172410
 2461 014066 012700 000201
 2462 014072 170100
 2463 014074 012737 014102 001236
 2464
 2465 014102 170700
 2466
 2467 014104 170205
 2468 014106 012700 000200
 2469 014112 170100
 2470 014114 012700 014230
 2471 014120 174010
 2472
 2473 014122 012701 000004
 2474 014126 005720
 2475 014130 001005
 2476 014132 077103
 2477 014134 022705 000204
 2478 014140 001014
 2479 014142 000442
 2480
 2481
 2482 014144 012737 014220 001242
 2483 014152 012737 014240 001240
 2484 014160 012737 014230 001244
 2485 014166 104071
 2486 014170 000427
 2487

DDB1:

LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
 MOV #200,R0 ;SET FD MODE.
 LDFPS R0
 MOV #DDBTP1,R0 ;SET UP ACO.
 LDD (R0),ACO ;SET ACO = 0
 CLR R0 ;CLEAR THE FPS.
 LDFPS R0
 MOV #DDBTP2,R0 ;LOAD ACO TO BE A FLOATING 0.
 LDF (R0),ACO ;SET ACO=ZERO
 ;FLOAT
 MOV #201,R0 ;SET FD MODE.
 LDFPS R0
 MOV #DDB2,@#\$TMP2

DDB2:

NEGD ACO ;TEST INSTRUCTION.
 STFPS R5 ;GET FPS.
 MOV #200,R0 ;SET FD MODE.
 LDFPS R0
 MOV #DDBBF0,R0 ;GET THE RESULT OUT OF ACO.
 STD ACO,(R0) ;SEE IF THE RESULT IS CORRECT.
 1\$: MOV #4,R1
 TST (R0)+
 BNE DDB5 ;BRANCH IF THE RESULT IS INCORRECT.
 SOB R1,1\$
 CMP #204,R5 ;IS THE FPS CORRECT?
 BNE DDB6 ;BRANCH IF THE FPS IS INCORRECT.
 BR DDBDONE

;RESULT INCORRECT, REPORT FAILURE:

DDB5: MOV #DDBTP2,@#\$TMP4 ;EXPECT D0
 MOV #DDBTP3,@#\$TMP3 ;PREV F0 IMPURE
 MOV #DDBBF0,@#\$TMP5 ;GOT
 1\$: ERROR +71
 BR DDBDONE


```

2488 ;REPORT FPS INCORRECT:
2489 014172 012737 000204 001240 DDB6: MOV #204,@#$TMP3
2490 014200 010537 001242 MOV R5,@#$TMP4
2491 014204 104072 1$: ERROR +72
2492 014206 000420 BR DDBDONE
2493
2494 ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
2495 014210 101112 DDBTP1: 101112
2496 014212 131415 131415
2497 014214 161710 161710
2498 014216 111213 111213
2499 014220 000000 DDBTP2: 0
2500 014222 000000 0
2501 014224 000000 0
2502 014226 000000 0
2503
2504 014230 177777 DDBBF0: -1
2505 014232 177777 -1
2506 014234 177777 -1
2507 014236 177777 -1
2508 014240 000000 DDBTP3: 0
2509 014242 000000 0
2510 014244 161710 161710
2511 014246 111213 111213
2512
2513 014250 DDBDONE: RSETUP
014250 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?).

```

```

2514
2515 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
2516

```

```

;*****
;*TEST 21 SEE ABOVE COMMENT FOR TEST TITLE
;*
;*THIS IS A TEST THE NEGF, ABSF AND TSTF
;*SOURCE FLOWS. THE NEGD INSTRUCTION
;*IS USED TO TEST MODE 1
;*
;*****
TST21: SCOPE

```

```

2517 014252 000004
2518 014254 EEB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
014254 104413 MOV #EEBTP1,R0 ;SET UP THE DATA BUFFER.
2519 014256 012700 014364 MOV #EEBBF1,R1
2520 014262 012701 014414 MOV #4,R2
2521 014266 012702 000004 1$: MOV (R0)+,(R1)+
2522 014272 012021 SOB R2,1$
2523 014274 077202 MOV #200,R0 ;SET FD MODE.
2524 014276 012700 000200 LDFPS R0
2525 014302 170100 MOV #EEBBF1,R0 ;SET UP THE OPERAND ADDRESS.
2526 014304 012700 014414 MOV #EEB2,@#$TMP2
2527 014310 012737 014324 001236 MOV #EEB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR.
2528 014316 012737 014424 000004 EEB2: NEGD (R0) ;TEST INSTRUCTION.
2529 014324 170710
2530

```

```

2531 014326 170205          STFPS  R5          ;GET FPS.
2532 014330 012701 014414  MOV    #EEBF1,R1   ;SEE IF RESULT IS CORRECT.
2533 014334 012702 000004          MOV    #4,R2
2534 014340 005721          1$:   TST    (R1)+
2535 014342 001046          BNE    EEB15       ;BRANCH IF NOT CORRECT.
2536 014344 077203          SOB    R2,1$
2537
2538 014346 020027 014414  CMP    R0,#EEBF1   ;IS R0 CORRECT?
2539 014352 001055          BNE    EEB20       ;BRANCH IF NOT CORRECT.
2540 014354 022705 000204  CMP    #204,R5     ;IS THE FPS CORRECT?
2541 014360 001061          BNE    EEB25       ;BRANCH IF NOT CORRECT.
2542 014362 000466          BR     EEBDONE
2543
2544          ;THESE ARE TEST DATA TABLES AND A BUFFER.
2545 014364 000177  EEBTP1: 177
2546 014366 167574          167574
2547 014370 137271          137271
2548 014372 107675          107675
2549 014374 000000  EEBTP2: 0
2550 014376 000000          0
2551 014400 000000          0
2552 014402 000000          0
2553 014404 177777  EEBBF0: -1
2554 014406 177777          -1
2555 014410 177777          -1
2556 014412 177777          -1
2557 014414 177777  EEBBF1: -1
2558 014416 177777          -1
2559 014420 177777          -1
2560 014422 177777          -1
2561
2562          ;IF A TRAP TO 4 OCCURS COME HERE:
2563 014424 011602  EEB10: MOV    (SP),R2   ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
2564 014426 020227 014326  CMP    R2,#EEB2+2
2565 014432 001405          BEQ    1$         ;BRANCH IF YES.
2566 014434 020227 014330  CMP    R2,#EEB2+4
2567 014440 001402          BEQ    1$         ;BRANCH IF YES.
2568 014442 000137 046310  JMP    @#CPSPUR   ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
2569          ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP TO 4.
2570 014446 022626 1$:   CMP    (SP)+(SP)+ ;RESET THE STACK.
2571 014450 010237 001236  MOV    R2,@#$TMP2
2572 014454 104107 2$:   ERROR +107      ;ODD ADRES
2573 014456 000430          BR     EEBDONE    ;BUT FDSTX IN ST 771
2574
2575          ;REPORT RESULT INCORRECT.
2576 014460 012737 014374 001242  EEB15: MOV    #EEBTP2,@#$TMP4
2577 014466 012737 014364 001240  MOV    #EEBTP1,@#$TMP3
2578 014474 012737 014414 001244  MOV    #EEBF1,@#$TMP5
2579 014502 104073 1$:   ERROR +73      ;BAD DATA X11*0 ST 312X
2580 014504 000415          BR     EEBDONE
2581
2582          ;R0 INCORRECT:
2583 014506 012737 014414 001240  EEB20: MOV    #EEBF1,@#$TMP3
2584 014514 010037 001242  MOV    R0,@#$TMP4
2585 014520 104074 1$:   ERROR +74      ;R0 BADX
2586 014522 000406          BR     EEBDONE
2587

```

```

2588 ;REPORT FPS INCORRECT:
2589 014524 010537 001240 EEB25: MOV R5,@#$TMP3
2590 014530 012737 000204 001244 MOV #204,@#$TMP5
2591 014536 104075 1$: ERROR +75 ;FPS X
2592
2593 014540 EEBDONE:
014540 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?).

```

2594 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
2595
2596

```

:*****
:*TEST 22 SEE ABOVE COMMENT FOR TEST TITLE
:*
:*THIS IS A TEST THE NEGF, ABSF AND TSTF
:*SOURCE FLOWS. THE ABSD INSTRUCTION
:*IS USED TO TEST MODE 2
:*
:*****

```

```

014542 000004 TST22: SCOPE
2597
2598 014544 FFB1:
014544 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2599 014546 012700 014654 MOV #FFBTP1,R0 ;SET UP THE DATA BUFFER.
2600 014552 012701 014704 MOV #FFBBF1,R1
2601 014556 012702 000004 MOV #4,R2
2602 014562 012021 1$: MOV (R0)+,(R1)+
2603 014564 077202 SOB R2,1$
2604 014566 012700 000200 MOV #200,R0 ;SET FD.
2605 014572 170100 LDFPS R0
2606 014574 012700 014704 MOV #FFBBF1,R0 ;SET UP THE OPERAND ADDRESS.
2607 014600 012737 014614 001236 MOV #FFB2,@#$TMP2
2608 014606 012737 014714 000004 MOV #FFB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2609
2610 014614 170620 FFB2: ABSD (R0)+ ;TEST INSTRUCTION.
2611
2612 014616 170205 STFPS R5 ;GET FPS.
2613 014620 012701 014704 MOV #FFBBF1,R1 ;CHECK RESULT.
2614 014624 012702 000004 MOV #4,R2
2615 014630 005721 1$: TST (R1)+
2616 014632 001046 BNE FFB15 ;BRANCH IF INCORRECT.
2617 014634 077203 SOB R2,1$
2618
2619 014636 020027 014714 CMP R0,#FFBBF1+10 ;IS R0 CORRECT?
2620 014642 001055 BNE FFB20 ;BRANCH IF INCORRECT.
2621 014644 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
2622 014650 001061 BNE FFB25 ;BRANCH IF INCORRECT.
2623 014652 000466 BR FFBDONE
2624
2625 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2626 014654 000177 FFBTP1: 177
2627 014656 167574 167574
2628 014660 137271 137271
2629 014662 107675 107675
2630 014664 000000 FFBTP2: 0

```

```

2631 014666 000000 0
2632 014670 000000 0
2633 014672 000000 0
2634 014674 177777 FFBBF0: -1
2635 014676 177777 -1
2636 014700 177777 -1
2637 014702 177777 -1
2638 014704 177777 FFBBF1: -1
2639 014706 177777 -1
2640 014710 177777 -1
2641 014712 177777 -1
2642
2643 ;IF A TRAP TO 4 OCCURS COME HERE.
2644 014714 011602 FFBI0: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2645 014716 020227 014616 CMP R2,#FFB2+2
2646 014722 001405 BEQ 1$ ;BRANCH IF YES.
2647 014724 020227 014620 CMP R2,#FFB2+4
2648 014730 001402 BEQ 1$ ;BRANCH IF YES.
2649 014732 000137 046310 JMP @#CPSUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2650 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2651 014736 022626 1$: CMP (SP)+,(SP)+
2652 014740 010237 001236 MOV R2,@#$TMP2
2653 014744 104076 2$: ERROR +76 ;ODD ADRES
2654 014746 000430 BR FFBDONE ;BUT FDSTX IN ST 771
2655
2656 ;REPORT RESULT INCORRECT:
2657 014750 012737 014664 001240 FFBI5: MOV #FFBTP2,@#$TMP3
2658 014756 012737 014654 001242 MOV #FFBTP1,@#$TMP4
2659 014764 012737 014704 001244 MOV #FFBBF1,@#$TMP5
2660 014772 104077 1$: ERROR +77 ;BAD DATA X11*0 ST 312X
2661 014774 000415 BR FFBDONE
2662
2663 ;REPORT R0 INCORRECT:
2664 014776 012737 014710 001240 FFBI20: MOV #FFBBF1+4,@#$TMP3
2665 015004 010037 001242 MOV R0,@#$TMP4
2666 015010 104100 1$: ERROR +100 ;R0 BADX
2667 015012 000406 BR FFBDONE
2668
2669 ;REPORT FPS INCORRECT:
2670 015014 010537 001240 FFBI25: MOV R5,@#$TMP3
2671 015020 012737 000204 001244 MOV #204,@#$TMP5
2672 015026 104101 1$: ERROR +101 ;FPS X
2673
2674 FFBDONE:
015030 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?).
2675 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
2676 ;*****
;*TEST 23 SEE ABOVE COMMENT FOR TEST TITLE
;*
;*THIS IS A TEST THE NEGF, ABSF AND TSTF
;*SOURCE FLOWS. THE ABSD INSTRUCTION
;*IS USED TO TEST MODE 4
;*
```

2677 015032 000004

.....
T23: SCOPE

2679	015034			GGB1:	LPERR			
	015034	104413			MOV	#GGBTP1,R0		;SET UP THE LOOP ON ERROR ADDRESS.
2680	015036	012700	015144		MOV	#GGBBF0,R1		;SET UP THE DATA BUFFER.
2681	015042	012701	015164		MOV	#4,R2		
2682	015046	012702	000004		MOV	(R0)+,(R1)+		
2683	015052	012021		1\$:				

```

2685 015054 077202          SOB      R2,1$
2686 015056 012700 000200  MOV      #200,R0          ;SET FD.
2687 015062 170100          LDFPS   R0
2688 015064 012700 015174  MOV      #GGBBF1,R0      ;SET UP THE OPERAND ADDRESS.
2689 015070 012737 015104 001236  MOV      #GGB2,@#$TMP2
2690 015076 012737 015204 000004  MOV      #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2691
2692 015104 170640          GGB2:   ABSD   -(R0)          ;TEST INSTRUCTION.
2693
2694 015106 170205          STFPS   R5                ;GET FPS.
2695 015110 012701 015164  MOV      #GGBBF0,R1      ;CHECK RESULT.
2696 015114 012702 000004  MOV      #4,R2
2697 015120 005721          1$:    TST    (R1)+
2698 015122 001046          BNE     GGB15            ;BRANCH IF INCORRECT.
2699 015124 077203          SOB     R2,1$
2700
2701 015126 020027 015164  CMP      R0,#GGBBF0      ;IS R0 CORRECT?
2702 015132 001055          BNE     GGB20            ;BRANCH IF INCORRECT.
2703 015134 022705 000204  CMP      #204,R5         ;IS THE FPS CORRECT?
2704 015140 001061          BNE     GGB25            ;BRANCH IF INCORRECT.
2705 015142 000466          BR      GGBDONE
2706
2707          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2708 015144 000177  GGBTP1: 177
2709 015146 117273          117273
2710 015150 147576          147576
2711 015152 177071          177071
2712 015154 000000  GGBTP2: 0
2713 015156 000000          0
2714 015160 000000          0
2715 015162 000000          0
2716 015164 177777  GGBBF0: -1
2717 015166 177777          -1
2718 015170 177777          -1
2719 015172 177777          -1
2720 015174 177777  GGBBF1: -1
2721 015176 177777          -1
2722 015200 177777          -1
2723 015202 177777          -1
2724
2725          ;IF A TRAP TO 4 OCCURS COME HERE.
2726 015204 011602  GGB10:  MOV    (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2727 015206 020227 015106  CMP      R2,#GGB2+2
2728 015212 001405          BEQ     1$              ;BRANCH IF YES.
2729 015214 020227 015110  CMP      R2,#GGB2+4
2730 015220 001402          BEQ     1$              ;BRANCH IF YES.
2731 015222 000137 046310  JMP      @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2732          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2733 015226 022626  1$:    CMP      (SP)+,(SP)+
2734 015230 010237 001236  MOV      R2,@#$TMP2
2735 015234 104102  2$:    ERROR  +102          ;ODD ADRES
2736 015236 000430          BR      GGBDONE        ;BUT FDSTX IN ST 771
2737
2738          ;REPORT RESULT INCORRECT:
2739 015240 012737 015154 001240  GGB15:  MOV      #GGBTP2,@#$TMP3
2740 015246 012737 015144 001242  MOV      #GGBTP1,@#$TMP4
2741 015254 012737 015164 001244  MOV      #GGBBF0,@#$TMP5

```

```

2742 015262 104103      1$:      ERROR      +103          ;BAD DATA X11*0 ST 312X
2743 015264 000415      BR          GGBDONE
2744
2745      ;REPORT R0 INCORRECT:
2746 015266 012737 015164 001240 GGB20: MOV      #GGBBF01,@#$TMP3
2747 015274 010037 001242      MOV      R0,@#$TMP4
2748 015300 104104      1$:      ERROR      +104          ;R0 BADX
2749 015302 000406      BR          GGBDONE
2750
2751      ;REPORT FPS INCORRECT:
2752 015304 010537 001240 GGB25: MOV      R5,@#$TMP3
2753 015310 012737 000204 001244      MOV      #204,@#$TMP5
2754 015316 104105      1$:      ERROR      +105          ;FPS X
2755
2756 015320      GGBDONE:
      015320 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?).
  
```

```

2757      ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
2758      ;*****
      ;*TEST 24      SEE ABOVE COMMENT FOR TEST TITLE
      ;*
      ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
      ;*SOURCE FLOWS. THE ABSD INSTRUCTION
      ;*IS USED TO TEST MODE 3
      ;*
      ;*****
      TST24: SCOPE
  
```

```

2759 015322 000004
2760 015324      HHB1:
      015324 104413      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
2761 015326 012700 015434      MOV      #HHBTP1,R0      ;SET UP THE DATA BUFFER.
2762 015332 012701 015464      MOV      #HHBBF0,R1
2763 015336 012702 000010      MOV      #10,R2
2764 015342 012021      1$:      MOV      (R0)+,(R1)+
2765 015344 077202      SOB      R2,1$
2766 015346 012700 000200      MOV      #200,R0          ;SET FD.
2767 015352 170100      LDFPS     R0
2768 015354 012700 015474      MOV      #HHBF1,R0        ;SET UP THE OPERAND ADDRESS.
2769 015360 012737 015374 001236      MOV      #HHB2,@#$TMP2
2770 015366 012737 015504 000004      MOV      #HHB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2771
2772 015374 170630      HHB2:  ABSD      @(R0)+      ;TEST INSTRUCTION.
2773
2774 015376 170205      STFPS     R5          ;GET FPS.
2775 015400 012701 015464      MOV      #HHBBF0,R1      ;CHECK RESULT.
2776 015404 012702 000004      MOV      #4,R2
2777 015410 005721      1$:      TST      (R1)+
2778 015412 001052      BNE      HHB15          ;BRANCH IF INCORRECT.
2779 015414 077203      SOB      R2,1$
2780 015416 020027 015476      CMP      R0,#HHBBF1+2    ;IS R0 CORRECT?
2781 015422 001061      BNE      HHB20          ;BRANCH IF INCORRECT.
2782 015424 022705 000204      CMP      #204,R5        ;IS THE FPS CORRECT?
2783 015430 001065      BNE      HHB25          ;BRANCH IF INCORRECT.
2784 015432 000472      BR          HHBDONE
  
```



```
2785
2786 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2787 015434 000177 HHBTP1: 177
2788 015436 147576 147576
2789 015440 177071 177071
2790 015442 107576 015464 177777 107576,HHBBF0,-1,-1,-1
    015450 177777 177777
2791 015454 000000 000000 000000 HHBTP2: 0,0,0,0
    015462 000000
2792 015464 177777 HHBBF0: -1
2793 015466 177777 -1
2794 015470 177777 -1
2795 015472 177777 -1
2796 015474 177777 HHBBF1: -1
2797 015476 177777 -1
2798 015500 177777 -1
2799 015502 177777 -1
2800
2801 ;IF A TRAP TO 4 OCCURS COME HERE.
2802 015504 011602 HHB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2803 015506 020227 015376 CMP R2,#HHB2+2
2804 015512 001405 BEQ 1$ ;BRANCH IF YES.
2805 015514 020227 015400 CMP R2,#HHB2+4
2806 015520 001402 BEQ 1$ ;BRANCH IF YES.
2807 015522 000137 046310 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2808 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2809 015526 022626 1$: CMP (SP)+,(SP)+
2810 015530 010237 001236 MOV R2,@#$TMP2
2811 015534 104106 2$: ERROR +106 ;ODD ADRES
2812 015536 000430 BR HHBDONE ;BUT FDSTX IN ST 771
2813
2814 ;REPORT RESULT INCORRECT:
2815 015540 012737 015454 001240 HHB15: MOV #HHBTP2,@#$TMP3
2816 015546 012737 015434 001242 MOV #HHBTP1,@#$TMP4
2817 015554 012737 015464 001244 MOV #HHBBF0,@#$TMP5
2818 015562 104110 1$: ERROR +110 ;BAD DATA X11*0 ST 3127
2819 015564 000415 BR HHBDONE
2820
2821 ;REPORT R0 INCORRECT:
2822 015566 012737 015476 001240 HHB20: MOV #HHBBF1+2,@#$TMP3
2823 015574 010037 001242 MOV R0,@#$TMP4
2824 015600 104111 1$: ERROR +111 ;R0 INCORRECT.
2825 015602 000406 BR HHBDONE
2826 ;REPORT FPS INCORRECT:
2827 015604 010537 001240 HHB25: MOV R5,@#$TMP3
2828 015610 012737 000204 001244 MOV #204,@#$TMP5
2829 015616 104112 1$: ERROR +112 ;FPSX
2830
2831 HHBDONE:
    015620 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?).
2832 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
2833 ;*****
;*TEST 25 SEE ABOVE COMMENT FOR TEST TITLE
```

```

: *
: * THIS IS A TEST THE NEGF, ABSF AND TSTF
: * SOURCE FLOWS. THE NEGD INSTRUCTION
: * IS USED TO TEST MODE 5
: *
: *****
TST25: SCOPE

2834 015622 000004
2835 015624
2836 015624 104413
2837 015626 012700 015734
2838 015632 012701 015764
2839 015636 012702 000010
2840 015642 012021
2841 015644 077202
2842 015646 012700 000200
2843 015652 170100
2844 015654 012700 015776
2845 015660 012737 015674 001236
2846 015666 012737 016004 000004
2847 015674 170750
2848
2849 015676 170205
2850 015700 012701 015764
2851 015704 012702 000004
2852 015710 005721
2853 015712 001052
2854 015714 077203
2855 015716 020027 015774
2856 015722 001061
2857 015724 022705 000204
2858 015730 001065
2859 015732 000472
2860
2861
2862 015734 000176
2863 015736 177074
2864 015740 127374
2865 015742 157677 015764 177777
2866 015750 177777 177777
2867 015754 000000
2868 015756 000000
2869 015760 000000
2870 015762 000000
2871 015764 177777
2872 015766 177777
2873 015770 177777
2874 015772 177777
2875 015774 177777
2876 015776 177777
2877 016000 177777
2878 016002 177777
2879
2880 016004 011602
2881 016006 020227 015676

: *****
: THESE ARE TEST DATA TABLES AND DATA BUFFER.
IIBTP1: 176
177074
127374
157677, IIBBF0, -1, -1, -1

IIBTP2: 0
0
0
0

IIBBF0: -1
-1
-1
-1

IIBBF1: -1
-1
-1
-1

: IF A TRAP TO 4 OCCURS COME HERE.
IIB10: MOV (SP), R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2, #IIB2+2

```

```

2882 016012 001405          BEQ      1$          ;BRANCH IF YES.
2883 016014 020227 015700    CMP      R2,#IIB2+4
2884 016020 001402          BEQ      1$          ;BRANCH IF YES.
2885 016022 000137 046310    JMP      @#CPSPUR    ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2886          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2887 016026 022626          1$:      CMP      (SP)+,(SP)+
2888 016030 010237 001236    MOV      R2,@#$TMP2
2889 016034 104113          2$:      ERROR    +113      ;ODD ADRES
2890 016036 000430          BR       IIBDONE     ;BUT FDSTX IN ST 771
2891
2892          ;REPORT RESULT INCORRECT:
2893 016040 012737 015754 001240 IIB15:  MOV      #IIBTP2,@#$TMP3
2894 016046 012737 015734 001242    MOV      #IIBTP1,@#$TMP4
2895 016054 012737 015764 001244    MOV      #IIBBF0,@#$TMP5
2896 016062 104114          1$:      ERROR    +114      ;BAD DATA X11*0 ST 3127
2897 016064 000415          BR       IIBDONE
2898
2899          ;REPORT R0 INCORRECT:
2900 016066 012737 015774 001240 IIB20:  MOV      #IIBBF1,@#$TMP3
2901 016074 010037 001242    MOV      R0,@#$TMP4
2902 016100 104115          1$:      ERROR    +115      ;R0 BADX
2903 016102 000406          BR       IIBDONE
2904          ;REPORT FPS INCORRECT:
2905 016104 010537 001240 001244 IIB25:  MOV      R5,@#$TMP3
2906 016110 012737 000204    MOV      #204,@#$TMP5
2907 016116 104116          1$:      ERROR    +116      ;FPSX
2908
2909 016120          IIBDONE:
        016120 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?).
  
```

2910
 2911
 2912

```

;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
:*****
:*TEST 26      SEE ABOVE COMMENT FOR TEST TITLE
:*
:*THIS IS A TEST THE NEGF, ABSF AND TSTF
:*SOURCE FLOWS. THE ABSD INSTRUCTION
:*IS USED TO TEST MODE 6
:*
:*****
  
```

```

2913 016122 000004          TST26:  SCOPE
2914 016124          JJB1:
        016124 104413          LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2915 016126 012700 016236    MOV      #JJBTP1,R0      ;SET UP THE DATA BUFFER.
2916 016132 012701 016260    MOV      #JJBFF0,R1
2917 016136 012702 000004    MOV      #4,R2
2918 016142 012021          1$:      MOV      (R0)+,(R1)+
2919 016144 077202          SOB      R2,1$
2920 016146 012700 000200    MOV      #200,R0        ;SET FD.
2921 016152 170100          LDFPS   R0
2922 016154 012700 016251    MOV      #JJBFF0-7,R0   ;SET UP THE OPERAND ADDRESS.
2923 016160 012737 016174 001236 MOV      #JJB2,@#$TMP2
2924 016166 012737 016300 000004 MOV      #JJB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
  
```

```

2925
2926 016174 170660 000007      JJB2:  ABSD      7(R0)          ;TEST INSTRUCTION.
2927
2928 016200 170205              STFPS      R5          ;GET FPS.
2929 016202 012701 016260      MOV        #JJBFF0,R1      ;CHECK RESULT.
2930 016206 012702 000004      MOV        #4,R2
2931 016212 005721              1$:      TST        (R1)+
2932 016214 001047              BNE        JJB15          ;BRANCH IF INCORRECT.
2933 016216 077203              SOB        R2,1$
2934 016220 020027 016251      CMP        R0,#JJBFF0-7    ;IS R0 CORRECT?
2935 016224 001043              BNE        JJB15          ;BRANCH IF INCORRECT.
2936 016226 022705 000204      CMP        #204,R5        ;IS THE FPS CORRECT?
2937 016232 001053              BNE        JJB20          ;BRANCH IF INCORRECT.
2938 016234 000467              BR         JJBDONE
2939
2940                          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2941 016236 000177      JJBTP1: 177
2942 016240 161524
2943 016242 131273
2944 016244 107174 000000      JJBTP2: 0
2945 016250 000000
2946 016252 000000
2947 016254 000000
2948 016256 000000
2949 016260 177777      JJBFF0: -1
2950 016262 177777
2951 016264 177777
2952 016266 177777
2953 016270 177777      JJBFF1: -1
2954 016272 177777
2955 016274 177777
2956 016276 177777
2957
2958                          ;IF A TRAP TO 4 OCCURS COME HERE.
2959 016300 011602      JJB10: MOV        (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2960 016302 020227 016176      CMP        R2,#JJB2+2
2961 016306 001405      BEQ        1$            ;BRANCH IF YES.
2962 016310 020227 016200      CMP        R2,#JJB2+4
2963 016314 001402      BEQ        1$            ;BRANCH IF YES.
2964 016316 000137 046310      JMP        @#CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2965                          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2966 016322 022626      1$:      CMP        (SP)+,(SP)+
2967 016324 010237 001236      MOV        R2,@#$TMP2
2968 016330 104117      2$:      ERROR      +117          ;ODD ADRES
2969 016332 000430      BR         JJBDONE      ;BUT FDSTX IN ST 771
2970
2971                          ;REPORT RESULT INCORRECT:
2972 016334 012737 016250 001240      JJB15: MOV        #JJBTP2,@#$TMP3
2973 016342 012737 016236 001242      MOV        #JJBTP1,@#$TMP4
2974 016350 012737 016260 001244      MOV        #JJBFF0,@#$TMP5
2975 016356 104120      1$:      ERROR      +120          ;BAD DATA X11*0 ST 3127
2976 016360 000415      BR         JJBDONE
2977
2978                          ;REPORT R0 INCORRECT:
2979 016362 012737 016251 001240      JJB20: MOV        #JJBFF0-7,@#$TMP3
2980 016370 010037 001242      MOV        R0,@#$TMP4
2981 016374 104124      1$:      ERROR      +124          ;R0 BADX
  
```

```

2982 016376 000406 BR JJB DONE
2983 ;REPORT FPS INCORRECT:
2984 016400 010537 001240 JJB25: MOV R5,@#$TMP3
2985 016404 012737 000204 001244 MOV #204,@#$TMP5
2986 016412 104122 1$: ERROR +122 ;FPSX
2987 016414 JJB DONE:
016414 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?).

```

```

2988 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
2989 :*****
:*TEST 27 SEE ABOVE COMMENT FOR TEST TITLE
:*
:*THIS IS A TEST THE NEGF, ABSF AND TSTF
:*SOURCE FLOWS. THE ABSD INSTRUCTION
:*IS USED TO TEST MODE 6
:*
:*****
TST27: SCOPE

```

```

2990 016416 000004
2991 016420 KKB1:
016420 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2992 016422 012700 016532 MOV #KKBTP1,R0 ;SET UP THE DATA BUFFER.
2993 016426 012701 016562 MOV #KKB BF0,R1
2994 016432 012702 000010 MOV #10,R2
2995 016436 012021 1$: MOV (R0)+,(R1)+
2996 016440 077202 SOB R2,1$
2997 016442 012700 000200 MOV #200,R0 ;SET FD.
2998 016446 170100 LDFPS R0
2999 016450 012700 016563 MOV #KKB BF1-7,R0 ;SET UP THE OPERAND ADDRESS.
3000 016454 012737 016470 001236 MOV #KKB2,@#$TMP2
3001 016462 012737 016602 000004 MOV #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3002
3003 016470 170770 000007 KKB2: NEGD @7(R0) ;TEST INSTRUCTION.
3004
3005 016474 170205 STFPS R5 ;GET FPS.
3006 016476 012701 016562 MOV #KKB BF0,R1 ;CHECK RESULT.
3007 016502 012702 000004 MOV #4,R2
3008 016506 005721 1$: TST (R1)+
3009 016510 001052 BNE KKB15 ;BRANCH IF INCORRECT.
3010 016512 077203 SOB R2,1$
3011 016514 020027 016563 CMP R0,#KKB BF1-7 ;IS R0 CORRECT?
3012 016520 001061 BNE KKB20 ;BRANCH IF INCORRECT.
3013 016522 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
3014 016526 001056 BNE KKB20 ;BRANCH IF INCORRECT.
3015 016530 000472 BR KKB DONE
3016
3017 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.

```

```

3018 016532 000177 KKBTP1: 177
3019 016534 167574 167574
3020 016536 137271 137271
3021 016540 107675 016562 177777 107675,KKB BF0,-1,-1,-1
016546 177777 177777
3022 016552 000000 KKBTP2: 0
3023 016554 000000 0

```

3024	016556	000000		0
3025	016560	000000		0
3026	016562	177777	KKBBF0:	-1
3027	016564	177777		-1
3028	016566	177777		-1
3029	016570	177777		-1
3030	016572	177777	KKBBF1:	-1
3031	016574	177777		-1
3032	016576	177777		-1

```

3034 016600 177777          -1
3035
3036          ;IF A TRAP TO 4 OCCURS COME HERE.
3037 016602 011602          KKB10: MOV      (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3038 016604 020227 016472          CMP      R2,#KKB2+2
3039 016610 001405          BEQ      1$          ;BRANCH IF YES.
3040 016612 020227 016474          CMP      R2,#KKB2+4
3041 016616 001402          BEQ      1$          ;BRANCH IF YES.
3042 016620 000137 046310          JMP      @#CPSPUR          ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3043          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3044 016624 022626          1$:      CMP      (SP)+,(SP)+
3045 016626 010237 001236          MOV      R2,@#$TMP2
3046 016632 104123          2$:      ERROR   +123          ;ODD ADRES
3047 016634 000430          BR       KKBDONE          ;BUT FDSTX IN ST 771
3048
3049          ;REPORT RESULT INCORRECT:
3050 016636 012737 016552 001240          KKB15: MOV      #KKBTP2,@#$TMP3
3051 016644 012737 016532 001242          MOV      #KKBTP1,@#$TMP4
3052 016652 012737 016562 001244          MOV      #KKBFBF0,@#$TMP5
3053 016660 104124          1$:      ERROR   +124          ;BAD DATA X11*0 ST 3127
3054 016662 000415          BR       KKBDONE
3055
3056          ;REPORT R0 INCORRECT:
3057 016664 012737 016563 001240          KKB20: MOV      #KKBFBF1-7,@#$TMP3
3058 016672 010037 001242          MOV      R0,@#$TMP4
3059 016676 104125          1$:      ERROR   +125          ;R0 BADX
3060 016700 000406          BR       KKBDONE
3061          ;REPORT FPS INCORRECT:
3062 016702 010537 001240          KKB25: MOV      R5,@#$TMP3
3063 016706 012737 000204 001244          MOV      #204,@#$TMP5
3064 016714 104126          1$:      ERROR   +126          ;FPSX
3065
3066 016716          KKBDONE:
3066 016716 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
          ;SEE IF THE USER HAS EXPRESSED
          ;THE DESIRE TO CHANGE THE SOFTWARE
          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
          ;THE USER TYPED CONTROL G?).
3067          ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 6, GR7
3068          ;*****
          ;*TEST 30          SEE ABOVE COMMENT FOR TEST TITLE
          ;*
          ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
          ;*SOURCE FLOWS. THE NEGD INSTRUCTION
          ;*IS USED TO TEST MODE 6
          ;*
          ;*****
3069 016720 000004          TST30: SCOPE
3069 016722          LLB1:
3069 016722 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3070 016724 012700 017022          MOV      #LLBTP1,R0          ;SET UP THE DATA BUFFER.
3071 016730 012701 017042          MOV      #LLBBF0,R1
3072 016734 012702 000004          MOV      #4,R2
3073 016740 012021          1$:      MOV      (R0)+,(R1)+
3074 016742 077202          SOB      R2,1$
3075 016744 012700 000200          MOV      #200,R0          ;SET FD.
3076 016750 170100          LDFPS      R0
    
```

```

3077 016752 012737 016766 001236      MOV    #LLB2,@#$TMP2
3078 016760 012737 017062 000004      MOV    #LLB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3079
3080 016766 170767 000050      LLB2:  NEG    LLBBF0      ;TEST INSTRUCTION.
3081
3082 016772 170205      STFPS  R5      ;GET FPS.
3083 016774 012701 017042      MOV    #LLBBF0,R1      ;CHECK RESULT.
3084 017000 012702 000004      MOV    #4,R2
3085 017004 005721      1$:    TST    (R1)+
3086 017006 001043      BNE   LLB15      ;BRANCH IF INCORRECT.
3087 017010 077203      SOB   R2,1$
3088 017012 022705 000204      CMP   #204,R5      ;IS THE FPS CORRECT?
3089 017016 001052      BNE   LLB25      ;BRANCH IF INCORRECT.
3090 017020 000457      BR    LLBDONE
3091
3092      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3093 017022 000127      LLBTP1: 127
3094 017024 137475
3095 017026 147372
3096 017030 117057
3097 017032 000000      LLBTP2: 0
3098 017034 000000
3099 017036 000000
3100 017040 000000
3101 017042 177777      LLBBF0: -1
3102 017044 177777
3103 017046 177777
3104 017050 177777
3105 017052 177777      LLBBF1: -1
3106 017054 177777
3107 017056 177777
3108 017060 177777
3109
3110      ;IF A TRAP TO 4 OCCURS COME HERE.
3111 017062 011602      LLB10: MOV    (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3112 017064 020227 016770      CMP    R2,#LLB2+2
3113 017070 001405      BEQ   1$      ;BRANCH IF YES.
3114 017072 020227 016772      CMP    R2,#LLB2+4
3115 017076 001402      BEQ   1$      ;BRANCH IF YES.
3116 017100 000137 046310      JMP   @#CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3117      ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3118 017104 022626      1$:    CMP    (SP)+,(SP)+
3119 017106 010237 001236      MOV    R2,@#$TMP2
3120 017112 104127      2$:    ERROR  +127      ;ODD ADRES
3121 017114 000421      BR    LLBDONE      ;BUT FDSTX IN ST 771
3122
3123      ;REPORT RESULT INCORRECT:
3124 017116 012737 017032 001240      LLB15: MOV    #LLBTP2,@#$TMP3
3125 017124 012737 017022 001242      MOV    #LLBTP1,@#$TMP4
3126 017132 012737 017042 001244      MOV    #LLBBF0,@#$TMP5
3127 017140 104130      1$:    ERROR  +130      ;BAD DATA X11*0 ST 3127
3128 017142 000406      BR    LLBDONE
3129      ;REPORT FPS INCORRECT:
3130 017144 010537 001240      LLB25: MOV    R5,@#$TMP3
3131 017150 012737 000204 001244      MOV    #204,@#$TMP5
3132 017156 104131      1$:    ERROR  +131      ;FPSX
3133

```



```

3134 017160 LLBDONE:
      017160 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?).

```

```

3135 ;TEST TITLE:NEGF, ABSF AND TSTF SOURCE MODE 7, GR7
3136 :*****
:*TEST 31 SEE ABOVE COMMENT FOR TEST TITLE
:*
:*THIS IS A TEST THE NEGF, ABSF AND TSTF
:*SOURCE FLOWS. THE ABSD INSTRUCTION
:*IS USED TO TEST MODE 7
:*
:*****

```

```

      017162 000004 TST31: SCOPE
3137
3138 017164 MMB1:
      017164 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3139 017166 012700 017264 MOV #MMBTP1,R0 ;SET UP THE DATA BUFFER.
3140 017172 012701 017314 MOV #MMBBF0,R1
3141 017176 012702 000010 MOV #10,R2
3142 017202 012021 1$: MOV (R0)+,(R1)+
3143 017204 077202 SOB R2,1$
3144 017206 012700 000200 MOV #200,R0 ;SET FD.
3145 017212 170100 LDFPS R0
3146 017214 012737 017230 001236 MOV #MMB2,@#$TMP2
3147 017222 012737 017334 000004 MOV #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3148
3149 017230 170677 000070 MMB2: ABSD @MMBBF1 ;TEST INSTRUCTION.
3150
3151 017234 170205 STFPS R5 ;GET FPS.
3152 017236 012701 017314 MOV #MMBBF0,R1 ;CHECK RESULT.
3153 017242 012702 000004 MOV #4,R2
3154 017246 005721 1$: TST (R1)+
3155 017250 001047 BNE MMB15 ;BRANCH IF INCORRECT.
3156 017252 077203 SOB R2,1$
3157 017254 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
3158 017260 001056 BNE MMB25 ;BRANCH IF INCORRECT.
3159 017262 000463 BR MMBDONE
3160

```

;THESE ARE TEST DATA TABLES AND DATA BUFFER.

```

3161 MMBTP1: 137
3162 017264 000137 045607
3163 017266 045607 101230
3164 017270 101230 45607,MMBBF0,-1,-1,-1
3165 017272 045607 017314 177777
      017300 177777 177777
3166 017304 000000 MMBTP2: 0
3167 017306 000000 0
3168 017310 000000 0
3169 017312 000000 0
3170 017314 177777 MMBBF0: -1
3171 017316 177777 -1
3172 017320 177777 -1
3173 017322 177777 -1
3174 017324 177777 MMBBF1: -1
3175 017326 177777 -1

```

```

3176 017330 177777          -1
3177 017332 177777          -1
3178
3179          ;IF A TRAP TO 4 OCCURS COME HERE.
3180 017334 011602          MMB10: MOV (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
3181 017336 020227 017232    CMP R2,#MMB2+2
3182 017342 001405          BEQ 1$          ;BRANCH IF YES.
3183 017344 020227 017234    CMP R2,#MMB2+4
3184 017350 001402          BEQ 1$          ;BRANCH IF YES.
3185 017352 000137 046310    JMP @#CPSPUR    ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3186          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3187 017356 022626          1$: CMP (SP)+,(SP)+
3188 017360 010237 001236    MOV R2,@#$TMP2
3189 017364 104132          2$: ERROR +132          ;ODD ADRES
3190 017366 000421          BR MMBDONE      ;BUT FDSTX IN ST 771
3191
3192          ;REPORT RESULT INCORRECT:
3193 017370 012737 017304 001240 MMB15: MOV #MMBTP2,@#$TMP3
3194 017376 012737 017264 001242    MOV #MMBTP1,@#$TMP4
3195 017404 012737 017314 001244    MOV #MMBBFO,@#$TMP5
3196 017412 104133          1$: ERROR +133          ;BAD DATA X11*0 ST 3127
3197 017414 000406          BR MMBDONE
3198          ;REPORT FPS INCORRECT:
3199 017416 010537 001240 MMB25: MOV R5,@#$TMP3
3200 017422 012737 000204 001244    MOV #204,@#$TMP5
3201 017430 104134          1$: ERROR +134          ;FPSX
3202
3203 017432          MMBDONE:
3204 017432 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?).
3210          ;*****
          ;*TEST 32          SPECIAL DEST, MODE 0, TEST
          ;*
          ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
          ;*MODE 0 USING THE NEGD INSTR.
          ;*
          ;*****
          TST32: SCOPE
3211 017434 000004
3212 017436          NNB1:
3213 017436 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3214 017440 012700 000200    MOV #200,R0    ;SET FD.
3215 017444 170100          LDFPS R0
3216 017446 012700 017534    MOV #NNBTP1,R0          ;SET UP ACO.
3217 017452 172410          LDD (R0),ACO
3218 017454 012737 017462 001236    MOV #NNB2,@#$TMP2
3219 017462 170700          NNB2: NEGD ACO          ;TEST INSTRUCTION.
3220
3221 017464 170205          STFPS R5          ;GET FPS.
3222 017466 012700 000200    MOV #200,R0    ;SET FD.
3223 017472 170100          LDFPS R0
3224 017474 012700 017554    MOV #NNBBFO,R0          ;GET THE RESULT.
3225 017500 174010          STD ACO,(R0)

```

```
3226 017502 012700 017554      MOV      #NNBBF0,R0      ;IS THE RESULT CORRECT?
3227 017506 012701 017544      MOV      #NNBTP2,R1
3228 017512 012702 000004      MOV      #4,R2
3229 017516 022021      1$:      CMP      (R0)+,(R1)+
3230 017520 001021      BNE      NNB10      ;BRANCH IF INCORRECT.
3231 017522 077203      SOB      R2,1$
3232 017524 022705 000210      CMP      #210,R5      ;IS THE FPS CORRECT?
3233 017530 001033      BNE      NNB15      ;BRANCH IF INCORRECT.
3234 017532 000440      BR       NNB DONE
3235
3236      ;THESE ARE DATA TABLES AND A DATA BUFFER.
3237 017534 013572      NNBTP1: 013572
3238 017536 046013      46013
3239 017540 057246      57246
3240 017542 013570      013570
3241 017544 113572      NNBTP2: 113572
3242 017546 046013      46013
3243 017550 057246      57246
3244 017552 013570      013570
3245 017554 000000      NNBBF0: 0
3246 017556 000000      0
3247 017560 000000      0
3248 017562 000000      0
3249
3250      ;REPORT RESULT INCORRECT:
3251 017564 012737 017554 001240      NNB10: MOV      #NNBBF0,@#$TMP3
3252 017572 012737 017544 001242      MOV      #NNBTP2,@#$TMP4
3253 017600 023737 017534 017554      CMP      @#NNBTP1,@#NNBBF0
3254 017606 001002      BNE      NNB11
3255 017610 104135      1$:      ERROR   +135      ;E10*200X ST 336
3256 017612 000410      BR       NNB DONE
3257
3258      ;REPORT RESULT INCORRECT:
3259 017614      NNB11:
3260 017614 104136      1$:      ERROR   +136      ;BAD DATA NEGf
3261 017616 000406      BR       NNB DONE
3262
3263      ;REPORT FPS INCORRECT:
3264 017620 010537 001242      NNB15: MOV      R5,@#$TMP4
3265 017624 012737 000210 001240      MOV      #210,@#$TMP3
3266 017632 104137      1$:      ERROR   +137      ;FPSX
3267
3268 017634      NNB DONE:
3269 017634 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?).
;*****
;*TEST 33      SPECIAL DEST, MODE 1, TEST
;*
;*THIS IS A TEST OF THE NEGf ABSF AND TSTf DESTINATION FLOWS
;*MODE 1 USING THE NEGf INSTR.
;*
;*****
3270 017636 000004      TST33: SCOPE
```

```

3271 017640          00B1: LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      017640 104413      MOV          #00BTP1,R1      ;SET UP THE DATA BUFFER.
3272 017642 012701 017752      MOV          #00BTP2,R0
3273 017646 012700 017762      MOV          #4,R2
3274 017652 012702 000004      MOV          (R0)+,(R1)+
3275 017656 012021      1$:      MOV          R2,1$
3276 017660 077202      SOB          R2,1$
3277 017662 012700 017752      MOV          #00BTP1,R0
3278 017666 042710 100000      BIC          #100000,(R0)      ;MAKE OPERAND POSITIVE.
3279 017672 012737 017706 001236      MOV          #00B2,@#$TMP2
3280 017700 012701 000200      MOV          #200,R1      ;SET FD.
3281 017704 170101      LDFPS       R1
3282
3283 017706 170710      00B2: NEGD          (R0)          ;TEST INSTRUCTION.
3284 017710 170205      STFPS       R5          ;GET FPS.
3285 017712 012701 017752      MOV          #00BTP1,R1      ;IS THE RESULT CORRECT.
3286 017716 012702 017762      MOV          #00BTP2,R2
3287 017722 012703 000004      MOV          #4,R3
3288 017726 022122      1$:      CMP          (R1)+,(R2)+
3289 017730 001020      BNE          00B10          ;BRANCH IF INCORRECT.
3290 017732 077303      SOB          R3,1$
3291 017734 022700 017752      CMP          #00BTP1,R0      ;IS R0 CORRECT.
3292 017740 001024      BNE          00B15          ;BRANCH IF INCORRECT.
3293 017742 022705 000210      CMP          #210,R5        ;IS THE FPS CORRECT?
3294 017746 001030      BNE          00B20          ;BRANCH IF INCORRECT.
3295 017750 000435      BR          00BDONE
3296
3297      ;THESE ARE DATA TABLES AND A DATA BUFFER.
3298 017752 023245      00BTP1: 023245
3299 017754 026720      26720
3300 017756 122324      122324
3301 017760 052672      52672
3302 017762 123245      00BTP2: 123245
3303 017764 026720      26720
3304 017766 122324      122324
3305 017770 052672      52672
3306
3307      ;REPORT RESULT INCORRECT:
3308 017772 012737 017752 001240      00B10: MOV          #00BTP1,@#$TMP3
3309 020000 012737 017762 001242      MOV          #00BTP2,@#$TMP4
3310 020006 104140      1$:      ERROR          +140      ;BAD DATA
3311 020010 000415      BR          00BDONE
3312
3313      ;REPORT R0 INCORRECT:
3314 020012 012737 017752 001240      00B15: MOV          #00BTP1,@#$TMP3
3315 020020 010037 001242      MOV          R0,@#$TMP4
3316 020024 104141      1$:      ERROR          +141      ;SPEC DESTX
3317 020026 000406      BR          00BDONE      ;ROX
3318
3319      ;REPORT FPS INCORRECT:
3320 020030 012737 000210 001240      00B20: MOV          #210,@#$TMP3
3321 020036 010537 001242      MOV          R5,@#$TMP4
3322 020042 104142      1$:      ERROR          +142
3323
3324 020044          00BDONE:
      020044 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?).

3325

:*****
:*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.
:*
:*****

3326 020046 000004
020050
020050 104413
3327
3328 020052 012701 020162
3329 020056 012700 020172
3330 020062 012702 000004
3331 020066 012021
3332 020070 077202
3333 020072 012700 020162
3334 020076 042710 100000
3335 020102 012737 020116 001236
3336 020110 012701 000200
3337 020114 170101
3338
3339 020116 170720
3340
3341 020120 170205
3342 020122 012701 020162
3343 020126 012702 020172
3344 020132 012703 000004
3345 020136 022122
3346 020140 001020
3347 020142 077303
3348 020144 022700 020172
3349 020150 001024
3350 020152 022705 000210
3351 020156 001030
3352 020160 000435
3353

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
1\$: MOV (R0)+,(R1)+
SOB R2,1\$
MOV #PPBTP1,R0
BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
MOV #PPB2,@#\$TMP2
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
1\$: 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

3354
3355 020162 023245
3356 020164 026720
3357 020166 122324
3358 020170 052672
3359 020172 123245
3360 020174 026720
3361 020176 122324
3362 020200 052672
3363
3364
3365 020202 012737 020162 001240
3366 020210 012737 020172 001242
3367 020216 104143
3368 020220 000415
3369
3370

;THESE ARE DATA TABLES AND A DATA BUFFER.
PPBTP1: 023245
26720
122324
52672
PPBTP2: 123245
26720
122324
52672
;REPORT RESULT INCORRECT:
PPB10: MOV #PPBTP1,@#\$TMP3
MOV #PPBTP2,@#\$TMP4
1\$: ERROR +143 ;BAD DATA
BR PPBDONE
;REPORT R0 INCORRECT:

```

3371 020222 012737 020172 001240 PPB15: MOV #PPBTP1+10,@#$TMP3
3372 020230 010037 001242 MOV R0,@#$TMP4
3373 020234 104144 1$: ERROR +144 ;SPEC DESTX ROX
3374 020236 000406 BR PPBDONE
3375
3376 ;REPORT FPS INCORRECT:
3377 020240 012737 000210 001240 PPB20: MOV #210,@#$TMP3
3378 020246 010537 001242 MOV R5,@#$TMP4
3379 020252 104145 1$: ERROR +145
3380
3381 020254 PPBDONE:
    020254 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?).
    
```

3382 ::*****

```

;*TEST 35 SPECIAL DEST, MODE 4, TEST
;*
;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
;*MODE 4 USING THE NEGD INSTR.
;*
    
```

```

3383 020256 000004 TST35: SCOPE
    020260 QQB1:
    020260 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3384 020262 012701 020374 MOV #QQBTP1,R1 ;SET UP THE DATA BUFFER.
3385 020266 012700 020414 MOV #QQBTP2,R0
3386 020272 012702 000004 MOV #4,R2
3387 020276 012021 1$: MOV (R0)+,(R1)+
3388 020300 077202 SOB R2,1$
3389 020302 012700 020404 MOV #QQBTP1+10,R0
3390 020306 042760 100000 177770 BIC #100000,-10(R0) ;MAKE OPERAND POSITIVE.
3391 020314 012737 020330 001236 MOV #QQB2,@#$TMP2
3392 020322 012701 000200 MOV #200,R1 ;SET FD.
3393 020326 170101 LDFPS R1
3394
3395 020330 170740 QQB2: NEGD -(R0) ;TEST INSTRUCTION.
3396
3397 020332 170205 STFPS R5 ;GET FPS.
3398 020334 012701 020374 MOV #QQBTP1,R1 ;IS THE RESULT CORRECT.
3399 020340 012702 020414 MOV #QQBTP2,R2
3400 020344 012703 000004 MOV #4,R3
3401 020350 022122 1$: CMP (R1)+,(R2)+
3402 020352 001024 BNE QQB10 ;BRANCH IF INCORRECT.
3403 020354 077303 SOB R3,1$
3404 020356 022700 020374 CMP #QQBTP1,R0 ;IS R0 CORRECT.
3405 020362 001030 BNE QQB15 ;BRANCH IF INCORRECT.
3406 020364 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
3407 020370 001034 BNE QQB20 ;BRANCH IF INCORRECT.
3408 020372 000441 BR QQBDONE
3409
3410 ;THESE ARE DATA TABLES AND A DATA BUFFER.
3411 020374 023245 QQBTP1: 023245
3412 020376 026720 26720
3413 020400 122324 122324
3414 020402 052672 52672
    
```

```
3415 020404 177777 177777 177777 .WORD -1,-1,-1,-1
      020412 177777
3416 020414 123245 QQBTP2: 123245
3417 020416 026720 26720
3418 020420 122324 122324
3419 020422 052672 52672
3420
3421 ;REPORT RESULT INCORRECT:
3422 020424 012737 020374 001240 QQB10: MOV #QQBTP1,@#$TMP3
3423 020432 012737 020414 001242 QQB10: MOV #QQBTP2,@#$TMP4
3424 020440 104146 1$: ERROR +146 ;BAD DATA
3425 020442 000415 BR QQBDONE
3426
3427 ;REPORT R0 INCORRECT:
3428 020444 012737 020374 001240 QQB15: MOV #QQBTP1,@#$TMP3
3429 020452 010037 001242 QQB15: MOV R0,@#$TMP4
3430 020456 104147 1$: ERROR +147 ;SPEC DESTX ROX
3431 020460 000406 BR QQBDONE
3432
3433 ;REPORT FPS INCORRECT:
3434
3435 020462 012737 000210 001240 QQB20: MOV #210,@#$TMP3
3436 020470 010537 001242 QQB20: MOV R5,@#$TMP4
3437 020474 104150 1$: ERROR +150
3438
3439 QQBDONE: 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?).

3440
3441 ;*****
;*TEST 36 SPECIAL DEST, MODE 3, TEST
;*
;*THIS IS A TEST OF THE NEGJ ABSF AND TSTF DESTINATION FLOWS
;*MODE 3 USING THE NEGJ INSTR.
;*
;*****
TST36: SCOPE
3442 020500 000004
3443 020502 RRB1:
      020502 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3444 020504 012701 020622 MOV #RRBTP1,R1 ;SET UP THE DATA BUFFER.
3445 020510 012700 020632 MOV #RRBTP2,R0
3446 020514 012702 000004 MOV #4,R2
3447 020520 012021 1$: MOV (R0)+,(R1)+
3448 020522 077202 SOB R2,1$
3449 020524 012700 020642 MOV #RRBTP3,R0
3450 020530 012710 020622 MOV #RRBTP1,(R0)
3451 020534 042737 100000 020622 BIC #100000,@#RRBTP1 ;MAKE THE OPERAND POSITIVE.
3452 020542 012737 020556 001236 MOV #RRB2,@#$TMP2
3453 020550 012701 000200 MOV #200,R1 ;SET FD.
3454 020554 170101 LDFPS R1
3455
3456 020556 170730 RRB2: NEGJ @ (R0)+ ;TEST INSTRUCTION.
3457
```

```

3458 020560 170205          STFPS R5          ;GET FPS.
3459 020562 012701 020622  MOV #RRBTP1,R1    ;IS THE RESULT CORRECT.
3460 020566 012702 020632  MOV #RRBTP2,R2
3461 020572 012703 000004  MOV #4,R3
3462 020576 022122          1$: CMP (R1)+,(R2)+
3463 020600 001021          BNE RRB10         ;BRANCH IF INCORRECT.
3464 020602 077303          SOB R3,1$
3465 020604 022700 020644  CMP #RRBTP3+2,R0 ;IS R0 CORRECT.
3466 020610 001025          BNE RRB15         ;BRANCH IF INCORRECT.
3467 020612 022705 000210  CMP #210,R5      ;IS THE FPS CORRECT?
3468 020616 001031          BNE RRB20         ;BRANCH IF INCORRECT.
3469 020620 000436          BR RRBDONE
3470
3471          ;THESE ARE DATA TABLES AND A DATA BUFFER.
3472 020622 023245  RRBTP1: 023245
3473 020624 026720          26720
3474 020626 122324          122324
3475 020630 052672          52672
3476 020632 123245  RRBTP2: 123245
3477 020634 026720          26720
3478 020636 123324          123324
3479 020640 052672          52672
3480 020642 020622  RRBTP3: RRBTP1
3481
3482          ;REPORT RESULT INCORRECT:
3483 020644 012737 020622 001240 RRB10: MOV #RRBTP1,@#$TMP3
3484 020652 012737 020632 001242  MOV #RRBTP2,@#$TMP4
3485 020660 104150          1$: ERROR +150    ;BAD DATA
3486 020662 000415          BR RRBDONE
3487
3488          ;REPORT R0 INCORRECT:
3489 020664 012737 020644 001240 RRB15: MOV #RRBTP3+2,@#$TMP3
3490 020672 010037 001242  MOV R0,@#$TMP4
3491 020676 104152          1$: ERROR +152    ;SPEC DESTX ROX
3492 020700 000406          BR RRBDONE
3493
3494          ;REPORT FPS INCORRECT:
3495 020702 012737 000210 001240 RRB20: MOV #210,@#$TMP3
3496 020710 010537 001242  MOV R5,@#$TMP4
3497 020714 104153          1$: ERROR +153
3498
3499          RRBDONE:
          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?).

```

3500
3501

```

:*****
:*TEST 37          SPECIAL DEST, MODE 5, TEST
:*
:*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
:*MODE 5 USING THE NEGD INSTR.
:*
:*****
TST37: SCOPE
SSB1:

```

3502 020720 000004
020722


```

3503 020722 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3504 020724 012701 021044  MOV      #SSBTP1,R1 ;SET UP THE DATA BUFFER.
3505 020730 012700 021054  MOV      #SSBTP2,R0
3506 020734 012702 000004  MOV      #4,R2
3507 020740 012021          1$: MOV      (R0)+,(R1)+
3508 020742 077202          SOB      R2,1$
3509 020744 012700 021066  MOV      #SSBTP3+2,R0
3510 020750 012760 021044 177776  MOV      #SSBTP1,-2(R0)
3511 020756 042737 100000 021044  BIC      #100000,@#SSBTP1 ;MAKE THE OPERAND POSITIVE.
3512 020764 012737 021000 001236  MOV      #SSB2,@#TMP2
3513 020772 012701 000200  MOV      #200,R1 ;SET FD.
3514 020776 170101          LDFPS   R1
3515 021000 170750          SSB2:  NEG   @-(R0) ;TEST INSTRUCTION.
3516
3517 021002 170205          STFPS   R5 ;GET FPS.
3518 021004 012701 021044  MOV      #SSBTP1,R1 ;IS THE RESULT CORRECT.
3519 021010 012702 021054  MOV      #SSBTP2,R2
3520 021014 012703 000004  MOV      #4,R3
3521 021020 022122          1$:  CMP      (R1)+,(R2)+
3522 021022 001021          BNE     SSB10 ;BRANCH IF INCORRECT.
3523 021024 077303          SOB     R3,1$
3524 021026 022700 021064  CMP      #SSBTP3,R0 ;IS R0 CORRECT.
3525 021032 001025          BNE     SSB15 ;BRANCH IF INCORRECT.
3526 021034 022705 000210  CMP      #210,R5 ;IS THE FPS CORRECT?
3527 021040 001031          BNE     SSB20 ;BRANCH IF INCORRECT.
3528 021042 000436          BR      SSBDONE
3529
3530 ;THESE ARE DATA TABLES AND A DATA BUFFER.
3531 021044 023245          SSBTP1: 023245
3532 021046 026720          26720
3533 021050 122324          122324
3534 021052 052672          52672
3535 021054 123245          SSBTP2: 123245
3536 021056 026270          26270
3537 021060 122324          122324
3538 021062 052672          52672
3539 021064 021044          SSBTP3: SSBTP1
3540
3541 ;REPORT RESULT INCORRECT:
3542 021066 012737 021044 001240 SSB10: MOV      #SSBTP1,@#TMP3
3543 021074 012737 021054 001242  MOV      #SSBTP2,@#TMP4
3544 021102 104154          1$:  ERROR   +154 ;BAD DATA
3545 021104 000415          BR      SSBDONE
3546
3547 ;REPORT R0 INCORRECT:
3548 021106 012737 021064 001240 SSB15: MOV      #SSBTP3,@#TMP3
3549 021114 010037 001242  MOV      R0,@#TMP4
3550 021120 104155          1$:  ERROR   +155 ;SPEC DESTX ROX
3551 021122 000406          BR      SSBDONE
3552
3553 ;REPORT FPS INCORRECT:
3554 021124 012737 000210 001240 SSB20: MOV      #210,@#TMP3
3555 021132 010537 001242  MOV      R5,@#TMP4
3556 021136 104156          1$:  ERROR   +156
3557
3558 021140          SSBDONE:

```

021140 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?).

3559

;*TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST
;*
;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
;*MODE 2 USING THE NEGF INSTR.
;*

021142 000004
3560 021144
021144 104413
3561 021146 012701 021256
3562 021152 012700 021266
3563 021156 012702 000004
3564 021162 012021
3565 021164 077202
3566 021166 012700 021256
3567 021172 042710 100000
3568 021176 012737 021212 001236
3569 021204 012701 000000
3570 021210 170101
3571

TST40: SCOPE
TTB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #TTBTP1,R1 ;SET UP THE DATA BUFFER.
MOV #TTBTP2,R0
MOV #4,R2
1\$: MOV (R0)+,(R1)+
SOB R2,1\$
MOV #TTBTP1,R0
BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
MOV #TTB2,@#\$TMP2
MOV #000,R1 ;SET FD.
LDFPS R1

3572 021212 170720
3573
3574 021214 170205
3575 021216 012701 021256
3576 021222 012702 021266
3577 021226 012703 000004
3578 021232 022122
3579 021234 001020
3580 021236 077303
3581 021240 022700 021262
3582 021244 001024
3583 021246 022705 000010
3584 021252 001030
3585 021254 000435
3586

TTB2: NEGF (R0)+ ;TEST INSTRUCTION.
STFPS R5 ;GET FPS.
MOV #TTBTP1,R1 ;IS THE RESULT CORRECT.
MOV #TTBTP2,R2
MOV #4,R3
1\$: CMP (R1)+,(R2)+
BNE TTB10 ;BRANCH IF INCORRECT.
SOB R3,1\$
CMP #TTBTP1+4,R0 ;IS R0 CORRECT.
BNE TTB15 ;BRANCH IF INCORRECT.
CMP #010,R5 ;IS THE FPS CORRECT?
BNE TTB20 ;BRANCH IF INCORRECT.
BR TTB DONE

3587
3588 021256 023245
3589 021260 026720
3590 021262 122324
3591 021264 052672
3592 021266 123245
3593 021270 026720
3594 021272 122324
3595 021274 052672
3596

;THESE ARE DATA TABLES AND A DATA BUFFER.
TTBTP1: 023245
26720
122324
52672
TTBTP2: 123245
26720
122324
52672

3597
3598 021276 012737 021256 001240
3599 021304 012737 021266 001242
3600 021312 104150
3601 021314 000415
3602

;REPORT RESULT INCORRECT:
TTB10: MOV #TTBTP1,@#\$TMP3
MOV #TTBTP2,@#\$TMP4
1\$: ERROR +150 ;BAD DATA
BR TTB DONE

```

3603 ;REPORT R0 INCORRECT:
3604 021316 012737 021262 001240 TTB15: MOV #TTBTP1+4,@#$TMP3
3605 021324 010037 001242 MOV R0,@#$TMP4
3606 021330 104160 1$: ERROR +160 ;SPEC DESTX R0X
3607 021332 000406 BR TTBDONE
3608
3609 ;REPORT FPS INCORRECT:
3610 021334 012737 000010 001240 TTB20: MOV #010,@#$TMP3
3611 021342 010537 001242 MOV R5,@#$TMP4
3612 021346 104161 1$: ERROR +161
3613
3614 021350 TTBDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
    021350 104412 ;SEE IF THE USER HAS EXPRESSED
    ;THE DESIRE TO CHANGE THE SOFTWARE
    ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
    ;THE USER TYPED CONTROL G?).
3615 ;TEST TITLE:SPECIAL DEST, MODE2, GR7 (IMMEDIATE)
3616 ;*****
    ;*TEST 41 SEE ABOVE COMMENT FOR TEST TITLE
    ;*
    ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
    ;*MODE 2(IMMEDIATE) USING THE NEGD INSTR.
    ;*
    ;*****
3617 021352 000004 TST41: SCOPE
    021354 UUB1:
    021354 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3618 021356 012700 021502 MOV #UUBTP2,R0
3619 021362 012701 021430 MOV #UUBTP1,R1 ;SET UP THE DATA BUFFER.
3620 021366 012702 000004 MOV #4,R2
3621 021372 012021 1$: MOV (R0)+,(R1)+
3622 021374 077202 SOB R2,1$
3623 021376 012700 021430 MOV #UUBTP1,R0
3624 021402 042737 100000 021430 BIC #100000,@#UUBTP1 ;MAKE THE OPERAND POSITIVE.
3625 021410 012737 021426 001236 MOV #UUB2,@#$TMP2
3626 021416 012701 000200 MOV #200,R1 ;SET FD.
3627 021422 170101 LDFPS R1
3628 021424 005001 CLR R1
3629
3630 021426 170727 UUB2: NEGD (R7)+ ;TEST INSTRUCTION.
3631 021430 005201 005201 005201 UUBTP1: 5201,5201,5201,5201
    021436 005201
3632 ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
3633 021440 170205 STFPS R5 ;GET FPS.
3634 021442 012703 021430 MOV #UUBTP1,R3 ;IS THE RESULT CORRECT.
3635 021446 012702 021502 MOV #UUBTP2,R2
3636 021452 012704 000004 MOV #4,R4
3637 021456 022322 1$: CMP (R3)+,(R2)+
3638 021460 001014 BNE UUB10 ;BRANCH IF INCORRECT.
3639 021462 077403 SOB R4,1$
3640 021464 022701 000003 CMP #3,R1 ;WAS R1 INCREMENTED CORRECTLY.
3641 021470 001027 BNE UUB15 ;BRANCH IF INCORRECT.
3642 021472 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
3643 021476 001015 BNE UUB20 ;BRANCH IF INCORRECT.
3644 021500 000436 BR UUBDONE
3645
    
```

```

3646 ;THESE ARE DATA TABLE.
3647 021502 105201 UUBTP2: 105201
3648 021504 005201 5201
3649 021506 005201 5201
3650 021510 005201 5201
3651
3652 ;REPORT RESULT INCORRECT:
3653 021512 012737 021430 001240 UUB10: MOV #UUBTP1,@#$TMP3
3654 021520 012737 021502 001242 MOV #UUBTP2,@#$TMP4
3655 021526 104162 1$: ERROR +162 ;BAD DATA
3656 021530 000422 BR UUBDONE
3657
3658 ;REPORT FPS INCORRECT:
3659 021532 012737 000210 001240 UUB20: MOV #210,@#$TMP3
3660 021540 010537 001242 MOV R5,@#$TMP4
3661 021544 104163 1$: ERROR +163 ;FPS
3662 021546 000413 BR UUBDONE
3663
3664 ;REPORT PC INCORRECTLY INCREMENTED DURING EXECUTION.
3665 021550 162701 000003 UUB15: SUB #3,R1
3666 021554 006301 ASL R1
3667 021556 012702 021432 MOV #UUBTP1+2,R2
3668 021562 010237 001240 MOV R2,@#$TMP3
3669 021566 160102 SUB R1,R2
3670 021570 010237 001242 MOV R2,@#$TMP4
3671 021574 104164 1$: ERROR +164 ;PC BAD CONSTAND B GR7X
3672
3673 021576 UUBDONE:
3674 021576 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?).

*****
;*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.
;
;*****
3675 021600 000004 TST42: SCOPE
3676 021602 104413 XXB1:
3677 021604 012701 021726 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3678 021610 012700 021736 MOV #XXBTP1,R1 ;SET UP THE DATA BUFFER.
3679 021614 012702 000004 MOV #XXBTP2,R0
3680 021622 077202 1$: MOV #4,R2
3681 021624 012700 014525 MOV (R0)+,(R1)+
3682 021630 042737 100000 021726 SOB R2,1$
3683 021636 012737 021654 001236 MOV #XXBTP1-5201,R0
3684 021644 012701 000200 BIC #100000,@#XXBTP1;MAKE OPERAND POSITIVE.
3685 021650 170101 MOV #XXB2,@#$TMP2
3686 3687 021652 005001 MOV #200,R1 ;SET FD.
3688 021654 170760 005201 LDFPS R1
XXB2: CLR R1
NEGD 5201(R0) ;TEST INSTRUCTION.
  
```

3690
3691 021660 170205
3692 021662 005701

STFPS R5
TST R1

;GET FPS.

```

3694 021664 001030      BNE      XXB25      ;WAS THE PC CORRECT AFTER EXECUTION?
3695 021666 012701 021726  MOV      #XXBTP1,R1      ;IS THE RESULT CORRECT.
3696 021672 012702 021736  MOV      #XXBTP2,R2
3697 021676 012703 000004  MOV      #4,R3
3698 021702 022122      1$:      CMP      (R1)+,(R2)+
3699 021704 001030      BNE      XXB10      ;BRANCH IF INCORRECT.
3700 021706 077303      SOB      R3,1$
3701 021710 022700 014525  CMP      #XXBTP1-5201,R0 ;IS R0 CORRECT.
3702 021714 001034      BNE      XXB15      ;BRANCH IF INCORRECT.
3703 021716 022705 000210  CMP      #210,R5      ;IS THE FPS CORRECT?
3704 021722 001040      BNE      XXB20      ;BRANCH IF INCORRECT.
3705 021724 000445      BR       XXBDONE

```

```

3706
3707      ;THESE ARE DATA TABLES AND A DATA BUFFER.
3708 021726 023245      XXBTP1: 023245
3709 021730 026720      26720
3710 021732 122324      122324
3711 021734 052672      52672
3712 021736 123245      XXBTP2: 123245
3713 021740 026720      26720
3714 021742 122324      122324
3715 021744 052672      52672

```

```

3716
3717
3718      ;REPORT PC INCORRECT AFTER EXECUTION.
3719 021746 012737 021656 001242  XXB25:  MOV      #XXB2+2,@#$TMP4
3720 021754 012737 021660 001240  MOV      #XXB2+4,@#$TMP3
3721 021762 104215      1$:      ERROR    +215      ;PC NOT INCREMENTED BY 2.
3722 021764 000425      BR       XXBDONE

```

```

3723
3724      ;REPORT RESULT INCORRECT:
3725 021766 012737 021726 001240  XXB10:  MOV      #XXBTP1,@#$TMP3
3726 021774 012737 021736 001242  MOV      #XXBTP2,@#$TMP4
3727 022002 104216      1$:      ERROR    +216      ;BAD DATA
3728 022004 000415      BR       XXBDONE

```

```

3729
3730      ;REPORT R0 INCORRECT:
3731 022006 012737 014525 001240  XXB15:  MOV      #XXBTP1-5201,@#$TMP3
3732 022014 010037 001242  MOV      R0,@#$TMP4
3733 022020 104217      1$:      ERROR    +217      ;SPEC DESTX ROX
3734 022022 000406      BR       XXBDONE

```

```

3735
3736      ;REPORT FPS INCORRECT:
3737
3738 022024 012737 000210 001240  XXB20:  MOV      #210,@#$TMP3
3739 022032 010537 001242  MOV      R5,@#$TMP4
3740 022036 104220      1$:      ERROR    +220

```

```

3741
3742 022040      XXBDONE:
3743 022040 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?).

```

3743
3744

```

;*****
;*TEST 43      SPECIAL DEST, MODE 7, TEST

```

```

022042 000004
3745
3746 022044
022044 104413
3747 022046 012701 022176
3748 022052 012700 022206
3749 022056 012702 000004
3750 022062 012021
3751 022064 077202
3752 022066 012700 015015
3753 022072 012760 022176 005201
3754 022100 042737 100000 022176
3755 022106 012737 022124 001236
3756 022114 012701 000200
3757 022120 170101
3758
3759 022122 005001
3760 022124 170770 005201
3761
3762 022130 170205
3763 022132 005701
3764 022134 001031
3765 022136 012701 022176
3766 022142 012702 022206
3767 022146 012703 000004
3768 022152 022122
3769 022154 001031
3770 022156 077303
3771 022160 022700 015015
3772 022164 001035
3773 022166 022705 000210
3774 022172 001041
3775 022174 000446
3776
3777
3778 022176 023245
3779 022200 026720
3780 022202 122324
3781 022204 052672
3782 022206 123245
3783 022210 026720
3784 022212 123324
3785 022214 052672
3786 022216 022176
3787
3788
3789 022220 016737 177702 001242
3790 022226 016737 177676 001240
3791 022234 104221
3792 022236 000425
3793
3794
  
```

```

:*
:*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
:*MODE 7 USING THE NEGD 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
1$: MOV (R0)+,(R1)+
SOB R2,1$
MOV #YYBTP3-5201,R0
MOV #YYBTP1,5201(R0)
BIC #100000,@#YYBTP1 ;MAKE THE OPERAND POSITIVE.
MOV #YYB2,@#$TMP2
MOV #200,R1 ;SET FD.
LDFPS R1
YYB2: CLR R1
NEGD @5201(R0) ;TEST INSTRUCTION.
STFPS R5 ;GET FPS.
TST R1 ;WAS THE PC CORRECT AFTER EXECUTION?
BNE YYB25
MOV #YYBTP1,R1 ;IS THE RESULT CORRECT.
MOV #YYBTP2,R2
MOV #4,R3
1$: CMP (R1)+,(R2)+
BNE YYB10 ;BRANCH IF INCORRECT.
SOB R3,1$
CMP #YYBTP3-5201,R0 ;IS R0 CORRECT.
BNE YYB15 ;BRANCH IF INCORRECT.
CMP #210,R5 ;IS THE FPS CORRECT?
BNE YYB20 ;BRANCH IF INCORRECT.
BR YYBDONE
;THESE ARE DATA TABLES AND A DATA BUFFER.
YYBTP1: 023245
26720
122324
52672
YYBTP2: 123245
26720
123324
52672
YYBTP3: YYBTP1
;REPORT PC INCORRECT AFTER EXECUTION.
YYB25: MOV YYB2+2,@#$TMP4
MOV YYB2+4,@#$TMP3
1$: ERROR +221 ;PC NOT INCREMENTED BY 2.
BR YYBDONE
;REPORT RESULT INCORRECT:
  
```

```

3795 022240 012737 022176 001240 YYB10: MOV #YYBTP1,@#$TMP3
3796 022246 012737 022206 001242   MOV #YYBTP2,@#$TMP4
3797 022254 104222   1$: ERROR +222 ;BAD DATA
3798 022256 000415   BR YYBDONE
3799
3800   ;REPORT RO INCORRECT:
3801 022260 012737 015015 001240 YYB15: MOV #YYBTP3-5201,@#$TMP3
3802 022266 010037 001242   MOV RO,@#$TMP4
3803 022272 104223   1$: ERROR +223 ;SPEC DESTX ROX
3804 022274 000406   BR YYBDONE
3805
3806   ;REPORT FPS INCORRECT:
3807 022276 012737 000210 001240 YYB20: MOV #210,@#$TMP3
3808 022304 010537 001242   MOV R5,@#$TMP4
3809 022310 104224   1$: ERROR +224
3810
3811 022312   YYBDONE:
      022312 104412   RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
                                     ;SEE IF THE USER HAS EXPRESSED
                                     ;THE DESIRE TO CHANGE THE SOFTWARE
                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                     ;THE USER TYPED CONTROL G?).
3817
      ;*****
      ;*TEST 44 NEG, ABSD AND TSTD TEST
      ;*
      ;*THIS IS A TEST OF THE NEG, ABSD AND TSTD INSTRUCTIONS.
      ;*
      ;*****
      TST44: SCOPE
      ;TEST NEG, ABSD WITH POS NONZERO OPERAND
      WWB1:
      LPERR PC,NATSUB ;SET UP THE LOOP ON ERROR ADDRESS.
      1$: 0 ;FLAG=NEG.
      2$: 16341 ;OPERAND.
          55772
          21133
          55447
      3$: 116341 ;RESULT.
          55772
          21133
          55447
      4$: 16341 ;ERROR RES.
          55772
          21133
          55447
      5$: 207 ;FPS BEFORE EXECUTION.
          210 ;FPS AFTER EXECUTION.
          200 ;ERROR FPS.
          -1 ;FEC
      6$: ERROR +200 ;E10<---E10*200X ST 336
          BR 7$
          ERROR +201 ;BUT ENBT ST 336X WENT TO 053 INTO 453
      7$:
      ;TEST NEG, ABSD WITH NEG OPERAND.
      WWB2:
      LPERR ;SET UP THE LOOP ON ERROR ADDRESS.

```



```

3844 022376 004767 000556      JSR      PC,NATSUB
3845 022402 000000      1$:      0          ;FLAG=NEGD.
3846 022404 152525      2$:      152525     ;OPERAND.
3847 022406 053545          53545
3848 022410 055565          55565
3849 022412 057505          57505
3850 022414 052525      3$:      52525     ;RESULT.
3851 022416 053545          53545
3852 022420 055565          55565
3853 022422 057505          57505
3854 022424 152525      4$:      152525     ;ERROR RES.
3855 022426 053545          53545
3856 022430 055565          55565
3857 022432 057505          57505
3858 022434 000217      5$:      217          ;FPS BEFORE EXECUTION.
3859 022436 000200          200          ;FPS AFTER EXECUTION.
3860 022440 000210          210          ;ERROR FPS.
3861 022442 177777          -1           ;FEC
3862 022444 104200      6$:      ERROR      +200     ;E10<---E10*200X S336
3863 022446 000401          BR          7$
3864 022450 104202      ERROR      +202     ;BUT ENBT X ST336 TO 453 INTO 053
3865 022452
3866
3867 022452      ;TEST ABSD WITH POSITIVE OPERAND
      WWB3:
3868 022452 104413      LPERR
3869 022454 004767 000500      JSR      PC,NATSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
3870 022460 000001      1$:      1          ;FLAG=ABSD.
3871 022462 060705      2$:      60705     ;OPERAND.
3872 022464 124735          124735
3873 022466 060124          60124
3874 022470 073560      3$:      73560     ;RESULT.
3875 022472 060705          60705
3876 022474 124735          124735
3877 022476 060124          60124
3878 022500 073560      4$:      73560     ;ERROR RES.
3879 022502 160705          160705
3880 022504 124735          124735
3881 022506 060124          60124
3882 022510 073560          73560
3883 022512 000217      5$:      217          ;FPS BEFORE EXECUTION.
3884 022514 000200          200          ;FPS AFTER EXECUTION.
3885 022516 000210          210          ;ERROR FPS.
3886 022520 177777          -1           ;EITHER BUT OP1B
3887 022522 104203      6$:      ERROR      +203     ;BUT ST 055 TO 336 INTO 335
3888 022524 000401          BR          7$
3889 022526 104203      ERROR      +203     ;OR BUT ENBT ST 335 TO 452 INTO 052
3890 022530
3891 022530      ;TEST ABSD WITH NEG. OPERAND
      WWB4:
3892 022530 104413      LPERR
3893 022532 004767 000422      JSR      PC,NATSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
3894 022536 000001      1$:      1          ;FLAG=ABSD.
3895 022540 154345      2$:      154345     ;OPERAND.
3896 022542 076567          76567
3897 022544 032123          32123
3898 022546 043234          43234
3899 022550 054345      3$:      54345     ;RESULT.
  
```

3899	022552	076567		76567		
3900	022554	032123		32123		
3901	022556	043234		43234		
3902	022560	154345	4\$:	154345		:ERROR RES.
3903	022562	076567		76567		
3904	022564	032123		32123		
3905	022566	043234		43234		
3906	022570	000217	5\$:	217		:FPS BEFORE EXECUTION.
3907	022572	000200		200		:FPS AFTER EXECUTION.
3908	022574	177777		-1		:ERROR FPS.
3909	022576	177777		-1		
3910	022600	104204	6\$:	ERROR +204		:E10*E10*200X ST 452
3911	022602	000401		BR 7\$		
3912	022604	104171		ERROR +171		
3913	022606		7\$:			
3914						
3915	022606					
	022606	104413				
3916	022610	004767	000344	LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
3917	022614	000002		JSR PC,NATSUB		
3918	022616	012321	1\$:	2		:FLAG=TSTD.
3919	022620	045654	2\$:	12321		:OPERAND.
3920	022622	070107		45654		
3921	022624	034543		70107		
3922	022626	012321	3\$:	34543		:RESULT.
3923	022630	045654		12321		
3924	022632	070107		45654		
3925	022634	034543		70107		
3926	022636	112321	4\$:	34543		:ERROR RES.
3927	022640	045654		112321		
3928	022642	070107		45654		
3929	022644	034543		70107		
3930	022646	000217	5\$:	34543		:FPS BEFORE EXECUTION.
3931	022650	000200		217		:FPS AFTER EXECUTION.
3932	022652	000210		200		:ERROR FPS.
3933	022654	177777		210		
3934	022656	104205	6\$:	-1		:BUT (OP1B) X ST044 TO 336 INTO 334
3935	022660	000401		ERROR +205		
3936	022662	104206		BR 7\$:BUT ENBT ST 334 TO 453 INTO 053
3937	022664			ERROR +206		
3938			7\$:			
3939	022664					
	022664	104413				
3940	022666	004767	000266	LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
3941	022672	000002		JSR PC,NATSUB		
3942	022674	123765	1\$:	2		:FLAG=TSTD.
3943	022676	023407	2\$:	123765		:OPERAND.
3944	022700	034510		23407		
3945	022702	045621		34510		
3946	022704	123765	3\$:	45621		:RESULT.
3947	022706	023407		123765		
3948	022710	034510		23407		
3949	022712	045621		34510		
3950	022714	023765	4\$:	45621		:ERROR RES.
3951	022716	023407		23765		
3952	022720	034510		23407		
3953	022722	045621		34510		
				45621		

3954	022724	000207		5\$:	207					:FPS BEFORE EXECUTION.
3955	022726	000210			210					:FPS AFTER EXECUTION.
3956	022730	000200			200					:ERROR FPS.
3957	022732	177777			-1					
3958	022734	104207		6\$:	ERROR	+207				:BUT OPB1 ST 055 TO 335 INTO 334
3959	022736	000401			BR	7\$				
3960	022740	104210			ERROR	+210				:BUT ENBT ST 334 TO 053 INTO 453
3961	022742			7\$:						
3962										
3963	022742									
3964	022742	104413								
3964	022744	004767	000210		LPERR					:SET UP THE LOOP ON ERROR ADDRESS.
3965	022750	000002			JSR	PC,NATSUB				
3965	022750	000002		1\$:	2					:FLAG=TSTD.
3966	022752	000175		2\$:	175					:OPERAND.
3967	022754	176737			176737					
3968	022756	071727			71727					
3969	022760	037574			37574					
3970	022762	000175		3\$:	175					:RESULT.
3971	022764	176737			176737					
3972	022766	071727			71727					
3973	022770	037574			37574					
3974	022772	000000		4\$:	0					:ERROR RES.
3975	022774	000000			0					
3976	022776	000000			0					
3977	023000	000000			0					
3978	023002	000200		5\$:	200					:FPS BEFORE EXECUTION.
3979	023004	000204			204					:FPS AFTER EXECUTION.
3980	023006	000214			214					:ERROR FPS.
3981	023010	177777			-1					
3982	023012	104211		6\$:	ERROR	+211				:BUT OP1B ST 255 TO 311 OR 312 INTO 310
3983	023014	000401			BR	7\$				
3984	023016	104212			ERROR	+212				:BUT ENBT ST 310 TO 402 INTO 002
3985	023020			7\$:						
3986										
3987	023020									
3988	023020	104413								
3988	023022	004767	000132		LPERR					:SET UP THE LOOP ON ERROR ADDRESS.
3989	023026	000002			JSR	PC,NATSUB				
3989	023026	000002		1\$:	2					:FLAG=TSTD.
3990	023030	100123		2\$:	100123					:OPERAND.
3991	023032	021012			21012					
3992	023034	034565			34565					
3993	023036	043210			43210					
3994	023040	100123		3\$:	100123					:RESULT.
3995	023042	021012			21012					
3996	023044	034565			34565					
3997	023046	043210			43210					
3998	023050	000000		4\$:	0					:ERROR RES.
3999	023052	000000			0					
4000	023054	000000			0					
4001	023056	000000			0					
4002	023060	040203		5\$:	40203					:FPS BEFORE EXECUTION.
4003	023062	040214			040214					:FPS AFTER EXECUTION.
4004	023064	140214			140214					:ERROR FPS.
4005	023066	177777			-1					
4006	023070	104211		6\$:	ERROR	+211				:+
4007	023072	000401			BR	7\$				
4008	023074	104213			ERROR	+213				:BUT FIUV ST 257 TO 355 INTO 255

```

4009 023076
4010
4011 023076
      023076 104413
4012 023100 004767 000054
4013 023104 000002
4014 023106 100137
4015 023110 024613
4016 023112 057024
4017 023114 060137
4018 023116 100137
4019 023120 024613
4020 023122 057024
4021 023124 060137
4022 023126 000000
4023 023130 000000
4024 023132 000000
4025 023134 000000
4026 023136 044200
4027 023140 144214
4028 023142 044214
4029 023144 000014
4030 023146 104211
4031 023150 000401
4032 023152 104214
4033 023154
4034 023154 000167 000414
4035
4036
4037

```

```

7$:
;TEST TSTD -0 OP FIUV=1
WMB9:
      LPERR
      JSR PC,NATSUB ;SET UP THE LOOP ON ERROR ADDRESS.
1$: 2 ;FLAG=TSTD.
2$: 100137 ;OPERAND.
      24613
      57024
      60137
3$: 100137 ;RESULT.
      24613
      57024
      60137
4$: 0 ;ERROR RES.
      0
      0
      0
5$: 44200 ;FPS BEFORE EXECUTION.
      144214 ;FPS AFTER EXECUTION.
      044214 ;ERROR FPS.
      14
6$: ERROR +211 ;+
      BR 7$
      ERROR +214 ;BUT FIUV ST 257 TO 255 INTO 355
7$:
      JMP WMBDONE

```

;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

4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095

:TO IT IS MADE THUS:

```

      JSR      PC,@#NATSUB
      FLAG:   .WORD  X           ;INSTRUCTION TYPE FLAG.
      ACARG:  .WORD  X,X,X,X    ;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      BR      CONT
      ERR2:   ERROR +X          ;FPS ERROR.
      CONT:   CONT           ;RETURN ADDRESS
  
```

```

:THE OPERAND IS SET UP IN NATBF1. THEN
:THE EITHER THE TSTD, NEG, OR ABSD INSTRUCTION IS EXECUTED.
:NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
:IS TO BE EXECUTED: 0 = NEG, 1 = ABSD, 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, ERRES. IF THE FAILURE IN
:THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES 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 = NEG, 1 = ABSD, 2 = TSTD
        MOV      #NATINS,R3
        ADD      R2,R3
        MOV      R3,@#STMP2
        JMP      (R3)           ;GO EXECUTE THE INSTRUCTION.
NATINS: NEG      (R0)
        BR      2$
        ABSD   (R0)
        BR      2$
        TSTD   (R0)
  
```

```

023160 012601
023162 010102
023164 062702 000002
023170 012703 023562
023174 012704 000004
023200 012223
023202 077402
023204 016100 000032
023210 170100
023212 012700 023562
023216 011102
023220 006302
023222 006302
023224 012703 023240
023230 060203
023232 010337 001236
023236 000113
023240 170710
023242 000403
023244 170610
023246 000401
023250 170510
  
```

```

4096 023252 170204          2$:   STFPS  R4           :GET THE FPS.
4097 023254 170305          STST   R5           :GET THE FEC.
4098 023256 010102          MOV    R1,R2
4099 023260 062702 000002    ADD    #2,R2
4100 023264 010237 001240    MOV    R2,@#$TMP3
4101 023270 062702 000010    ADD    #10,R2
4102 023274 010237 001244    MOV    R2,@#$TMP5
4103 023300 012737 023562 001242    MOV    #NATBF1,@#$TMP4
4104 023306 010437 001250    MOV    R4,@#$TMP7
4105 023312 016137 000034 001252    MOV    34(R1),@#$TMP10
4106 023320 010100          MOV    R1,R0           ;WAS THE RESULT CORRECT?
4107 023322 062700 000012    ADD    #12,R0
4108 023326 012702 023562    MOV    #NATBF1,R2
4109 023332 012703 000004    MOV    #4,R3
4110 023336 022022          3$:   CMP    (R0)+,(R2)+
4111 023340 001014          BNE   10$             ;BRANCH IF INCORRECT.
4112 023342 077303          SOB   R3,3$
4113 023344 026104 000034    CMP    34(R1),R4       ;WAS THE FPS CORRECT?
4114 023350 001032          BNE   15$             ;BRANCH IF INCORRECT.
4115 023352 005761 000034    TST   34(R1)           ;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
4116 023356 100003          BPL   4$
4117 023360 026105 000040    CMP    40(R1),R5       ;WAS THE FEC CORRECT.
4118 023364 001037          BNE   20$             ;BRANCH IF INCORRECT.
4119 023366 000161 000050          4$:   JMP    50(R1)         ;RETURN.
4120
4121          ;THE RESULT WAS INCORRECT BUT WAS THIS FAILURE ANTICIPATED?
4122          ;SEE IF THE RESULT WAS ANTICIPATED:
4123 023372          10$:
4124 023372 011105          MOV    (R1),R5
4125 023374 006305          ASL   R5
4126 023376 006305          ASL   R5
4127 023400 062705 023512    ADD    #NATER1,R5
4128 023404 010100          MOV    R1,R0
4129 023406 062700 000022    ADD    #22,R0
4130 023412 012702 023562    MOV    #NATBF1,R2
4131 023416 012703 000004    MOV    #4,R3
4132 023422 022022          11$:  CMP    (R0)+,(R2)+
4133 023424 001003          BNE   12$             ;BRANCH IF NOT ANTICIPATED.
4134 023426 077303          SOB   R3,11$
4135
4136          ;THE ERROR WAS ANTICIPATED SO RETURN.
4137 023430 000161 000042          JMP    42(R1)
4138
4139          ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
4140 023434 000115          12$:  JMP    (R5)           ;GO TO THE PROPER ERROR CALL.
4141
4142          ;THE FPS WAS INCORRECT.
4143 023436 026105 000036          15$:  CMP    36(R1),R5       ;WAS THIS ERROR ANTICIPATED?
4144 023442 001002          BNE   16$             ;BRANCH IF NOT ANTICIPATED.
4145
4146          ;THE FPS ERROR WAS ANTICIPATED SO RETURN.
4147 023444 000161 000046          JMP    46(R1)
4148
4149          ;THE FPS FAILURE WAS NOT ANTICIPATED SO REPORT IT HERE.
4150 023450 011102          16$:  MOV    (R1),R2
4151 023452 006302          ASL   R2
4152 023454 006302          ASL   R2
    
```

```

4153 023456 062702 023530      ADD    #NATER2,R2
4154 023462 000112      JMP    (R2)                ;GO TO THE PROPER ERROR CALL.
4155
4156      ;REPORT THAT THE FEC WAS INCORRECT.
4157 023464 016137 000040 001256 20$: MOV    40(R1),@#$TMP12
4158 023472 010537 001254      MOV    R5,@#$TMP11
4159 023476 011102      MOV    (R1),R2
4160 023500 006302      ASL   R2
4161 023502 006302      ASL   R2
4162 023504 062702 023544      ADD    #NATER3,R2
4163 023510 000112      JMP    (R2)                ;GO TO THE PROPER ERROR CALL.
4164
4165      ;THESE ARE THE ERROR CALLS FOR EACH INDIVIDUAL INSTRUCTION AND CONDITION.
4166 023512 104165  NATER1: ERROR +165      ;NEGD BAD DATA
4167 023514 000403      BR    NATRET
4168 023516 104166      ERROR +166      ;ABSD BAD DATA
4169 023520 000401      BR    NATRET
4170 023522 104167      ERROR +167      ;TSTD BAD DATA
4171 023524 000161 000050  NATER2: JMP    50(R1)
4172
4173      ;FPS INCORRECT:
4174 023530 104170  NATER2: ERROR +170      ;NEGD FPSX
4175 023532 000774      BR    NATRET
4176 023534 104171      ERROR +171      ;ABSD FPSX
4177 023536 000772      BR    NATRET
4178 023540 104172      ERROR +172      ;TSTD FPSX
4179 023542 000770      BR    NATRET
4180
4181      ;FEC INCORRECT:
4182 023544 104173  NATER3: ERROR +173      ;NEGD FECX
4183 023546 000766      BR    NATRET
4184 023550 104174      ERROR +174      ;ABSD FECX
4185 023552 000764      BR    NATRET
4186 023554 104175      ERROR +175      ;TSTD FECX
4187 023556 000762      BR    NATRET
4188
4189 023560 177777      .WORD -1
4190 023562 177777 177777 177777 NATBF1: .WORD -1,-1,-1,-1,-1
4191 023570 177777 177777
4192 023574 104412      WWDONE: 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?).
4193
4194
4201
4202

```

023576 000004

```

:*****
:*TEST 45      SOURCE MODES, MODE 1 (FL=0), TEST
:*
:* THIS IS A TEST OF SOURCE MODE 1
:* USING THE LDFPS INSTR
:*
:*****
TST45: SCOPE

```

```

4203
4204
4205 023600          AAC1:          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      023600 104413
4206
4207 023602 012700 023660          MOV          #AACTP1,R0          ;SET UP TEST DATA IN BUFFER.
4208 023606 012710 147517          MOV          #147517,(R0)
4209 023612 012737 147517 001240          MOV          #147517,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
4210 023620 012737 023634 001236          MOV          #AAC2,@#$TMP2
4211 023626 012737 023720 000004          MOV          #AAC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4212 023634 170110          AAC2:          LDFPS          (R0)          ;TEST INSTRUCTION.
4213
4214 023636 170205          STFPS          R5          ;GET FPS
4215
4216 023640 020027 023660          CMP          R0,#AACTP1          ;IS R0 CORRECT?
4217 023644 001007          BNE          AAC10          ;BR IF NOT.
4218 023646 022705 147517          CMP          #147517,R5          ;IS FPS CORRECT?
4219 023652 001013          BNE          AAC11          ;BR IF NOT.
4220 023654 000437          BR          AACDONE
4221
4222          ;TEST BUFFER AND DATA:
4223 023656 177777          -1
4224 023660 147517          AAC10: 147517
4225 023662 177777          -1
4226
4227          ;REPORT R0 INCORRECT.
4228 023664 012737 023660 001240          AAC10: MOV          #AACTP1,@#$TMP3
4229 023672 010037 001242          MOV          R0,@#$TMP4
4230 023676 104225          1$:          ERROR          +225          ;R0 BAD BUT FSRC FAILED
4231 023700 000425          BR          AACDONE
4232
4233          ;REPORT FPS INCORRECT.
4234 023702 012737 147517 001240          AAC11: MOV          #147517,@#$TMP3 ;REPORT FPS INCORRECT.
4235 023710 010537 001242          MOV          R5,@#$TMP4
4236 023714 104226          1$:          ERROR          +226
4237 023716 000416          BR          AACDONE
4238
4239          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4240          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4241          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4242 023720          AAC20:
4243 023720 011602          MOV          (SP),R2
4244 023722 020227 023636          CMP          R2,#AAC2+2
4245 023726 001405          BEQ          1$
4246 023730 020227 023640          CMP          R2,#AAC2+4
4247 023734 001402          BEQ          1$
4248 023736 000137 046310          JMP          @#CPSPUR
4249 023742 022626          1$:          CMP          (SP)+,(SP)+
4250 023744 010237 001236          MOV          R2,@#$TMP2
4251 023750 104227          2$:          ERROR          +227          ;ODD ADRES
4252 023752 000400          BR          AACDONE          ;BUT FDSTX IN ST 771
4253
4254 023754          AACDONE:
      023754 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?).

4255
 4256
 4257

```

:*****
:*TEST 46 SOURCE MODES, MODE 2 (FL=0), TEST
:*
:* THIS IS A TEST OF SOURCE MODE 2
:* USING THE LDFPS INSTR
:*
:*****
TST46: SCOPE
  
```

023756 000004
 4258
 4259 023760
 023760 104413
 4260
 4261 023762 012700 024040
 4262 023766 012710 145212
 4263 023772 012737 145212 001240
 4264 024000 012737 024014 001236
 4265 024006 012737 024100 000004
 4266
 4267 024014 170120
 4268
 4269 024016 170205
 4270
 4271 024020 020027 024042
 4272 024024 001007
 4273 024026 022705 145212
 4274 024032 001013
 4275 024034 000436
 4276
 4277
 4278
 4279 024036 177777
 4280 024040 177777
 4281 024042 177777
 4282
 4283
 4284
 4285 024044 012737 024042 001240
 4286 024052 010037 001242
 4287 024056 104230
 4288 024060 000424
 4289
 4290
 4291 024062 012737 145212 001240
 4292 024070 010537 001242
 4293 024074 104231
 4294 024076 000415
 4295
 4296
 4297
 4298
 4299 024100
 4300 024100 011602
 4301 024102 020227 024016
 4302 024106 001405

```

BBC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
      MOV #BBCTP1,R0 ;SET UP TEST DATA IN BUFFER.
      MOV #145212,(R0)
      MOV #145212,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
      MOV #BBC2,@#$TMP2
      MOV #BBC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
BBC2: LDFPS (R0)+ ;TEST INSTRUCTION.
      STFPS R5 ;GET FPS
      CMP R0,#BBCTP1+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
BBC10: .WORD -1
      -1
;REPORT R0 INCORRECT.
BBC10: MOV #BBCTP1+2,@#$TMP3
      MOV R0,@#$TMP4
1$: ERROR +230 ;R0 BAD BUT FSRC FAILED
      BR BBCDONE
;REPORT FPS INCORRECT.
BBC11: MOV #145212,@#$TMP3 ;REPORT FPS INCORRECT.
      MOV R5,@#$TMP4
1$: 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: MOV (SP),R2
      CMP R2,#BBC2+2
      BEQ 1$
  
```

```

4303 024110 020227 024020          CMP    R2,#BBC2+4
4304 024114 001402          BEQ    1$
4305 024116 000137 046310          JMP    @#CPSPUR
4306 024122 022626          1$:   CMP    (SP)+,(SP)+
4307 024124 010237 001236          MOV    R2,@#$TMP2
4308 024130 104232          2$:   ERROR  +232          ;ODD ADRES
4309                                     ;BUT FDSTX IN ST 771
4310
4311 024132          BBCDONE:
      024132 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?).

```

4312
4313
4314

```

:*****
:*TEST 47          SOURCE MODES, MODE 4 (FL=0), TEST
:*
:* THIS IS A TEST OF SOURCE MODE 4
:* USING THE LDFPS INSTR
:*
:*****

```

```

      024134 000004          TST47: SCOPE
4315
4316 024136          DDC1:
      024136 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4317
4318 024140 012700 024230          MOV    #DDCTP1+2,R0          ;SET UP THE TEST DATA BUFFER.
4319 024144 012760 105252 177776          MOV    #105252,-2(R0)
4320 024152 012737 105252 001240          MOV    #105252,@#$TMP3          ;SAVE DATA IN CASE OF ERROR.
4321 024160 012737 024174 001236          MOV    #DDC2,@#$TMP2
4322 024166 012737 024274 000004          MOV    #DDC20,@#ERRVEC
4323 024174 170140          DDC2:  LDFPS  -(R0)
4324 024176 170205          STFPS  R5
4325 024200 020027 024226          CMP    R0,#DDCTP1
4326 024204 001015          BNE   DDC10
4327 024206 022705 105252          CMP    #105252,R5
4328 024212 001021          BNE   DDC11
4329 024214 000444          BR    DDCDONE
4330
4331 024216 177777 177777 177777          -1,-1,-1,-1
      024224 177777
4332 024226 177777          DDCTP1: -1
4333 024230 177777 177777 177777          -1,-1,-1,-1
      024236 177777
4334
4335 024240 012737 024226 001240          DDC10: MOV    #DDCTP1,@#$TMP3
4336 024246 010037 001242          MOV    R0,@#$TMP4
4337 024252 104233          1$:   ERROR  +233          ;R0 BAD BUT FSRC FAILED
4338 024254 000424          BR    DDCDONE
4339 024256 012737 105252 001240          DDC11: MOV    #105252,@#$TMP3          ;REPORT FPS INCORRECT.
4340 024264 010537 001242          MOV    R5,@#$TMP4
4341 024270 104234          1$:   ERROR  +234
4342 024272 000415          BR    DDCDONE
4343 024274 011602          DDC20: MOV    (SP),R2
4344 024276 020227 024176          CMP    R2,#DDC2+2

```

```

4345 024302 001405          BEQ      1$
4346 024304 020227 024200    CMP      R2,#DDC2+4
4347 024310 001402          BEQ      1$
4348 024312 000137 046310    JMP      @#CPSPUR
4349 024316 022626          1$:     CMP      (SP)+,(SP)+
4350 024320 010237 001236    MOV      R2,@#$TMP2
4351 024324 104235          2$:     ERROR   +235          ;DDD ADRES
4352 024326 104412          DDCDONE: 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?).
    
```

4353 :*****

```

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

```

4354 024330 000004          TST50: SCOPE
      024332 104413          EEC1:
4355 024334 012700 024436    LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4356 024340 012710 024426    MOV      #EECTP2,R0
4357 024344 012767 103456 000054    MOV      #EECTP1,(R0)
4358 024352 012737 103456 001240    MOV      #103456,EECTP1
4359 024360 012737 024374 001236    MOV      #103456,@#$TMP3
4360 024366 012737 024504 000004    MOV      #EEC2,@#$TMP2
4361 024374 170130          EEC2: MOV      #EEC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4362 024376 170205          LDFPS   @(R0)+          ;TEST INSTRUCTION.
4363 024400 020027 024440    STFPS   R5              ;GET THE FPS.
4364 024404 001021          CMP      R0,#EECTP2+2   ;IS R0 CORRECT?
4365 024406 022705 103456    BNE     EEC10           ;BR IF NOT.
4366 024412 001025          CMP      #103456,R5     ;IS THE FPS CORRECT?
4367 024414 000450          BNE     EEC11           ;BR IF NOT.
4368
4369
4370
4371 024416 177777 177777 177777 ;TEST BUFFER AND DATA:
      024424 177777          -1,-1,-1,-1
4372 024426 177777          EECTP1: -1
4373 024430 177777 177777 177777 -1,-1,-1
4374 024436 024426 177777 177777 EECTP2: EECTP1,-1,-1,-1,
      024444 177777 000000
4375
4376
4377
4378 024450 012737 024440 001240 ;REPORT R0 INCORRECT.
4379 024456 010037 001242    EEC10: MOV      #EECTP2+2,@#$TMP3
4380 024462 104236          MOV      R0,@#$TMP4
4381 024464 000424          1$:     ERROR   +236          ;R0 BAD BUT FSRC FAILED
4382
4383
4384 024466 012737 103456 001240 ;REPORT FPS INCORRECT.
4385 024474 010537 001242    EEC11: MOV      #103456,@#$TMP3 ;REPORT FPS INCORRECT.
4386 024500 104237          MOV      R5,@#$TMP4
      1$:     ERROR   +237
    
```

```
4387 024502 000415          BR      EECDONE
4388          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4389          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4390          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4391 024504 011602          EEC20:  MOV    (SP),R2
4392 024506 020227 024376      CMP    R2,#EEC2+2
4393 024512 001405          BEQ    1$
4394 024514 020227 024400      CMP    R2,#EEC2+4
4395 024520 001402          BEQ    1$
4396 024522 000137 046310      JMP    @#CPSPUR
4397 024526 022626          1$:    CMP    (SP)+,(SP)+
4398 024530 010237 001236      MOV    R2,@#$TMP2
4399 024534 104240          2$:    ERROR  +240          ;DDD ADRES
4400 024536 104412          EECDONE: 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?).

4401          ;*****
          ;*TEST 51          SOURCE MODES, MODE 5 (FL=0), TEST
          ;*
          ;* THIS IS A TEST OF SOURCE MODE 5
          ;* USING THE LDFPS INSTR
          ;*
          ;*****
4402 024540 000004          TST51: SCOPE
          024542 104413          FFC1:    LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
          024544 012700 024644      MOV    #FFCTP2+2,R0      ;SET UP THE TEST DATA BUFFER.
4403 024544 012700 024644      MOV    #FFCTP1,-2(R0)
4404 024550 012760 024632 177776  MOV    #45412,@#FFCTP1
4405 024556 012737 045412 024632  MOV    #45412,@#$TMP3
4406 024564 012737 045412 001240  MOV    #FFC1,@#$TMP2      ;SAVE DATA IN CASE OF ERROR.
4407 024572 012737 024542 001236  MOV    #FFC20,@#ERRVECT
4408 024600 012737 024706 000004  MOV    @-(R0)          ;SET UP FOR TRAPS TO 4.
4409 024606 170150          FFC2:  LDFPS  @-(R0)      ;TEST INSTRUCTION.
4410 024610 170205          STFPS  R5              ;GET THE FPS.
4411 024612 020027 024642      CMP    R0,#FFCTP2      ;IS R0 CORRECT?
4412 024616 001015          BNE   FFC10            ;BR IF NOT.
4413 024620 022705 045412      CMP    #45412,R5       ;IS THE FPS CORRECT?
4414 024624 001021          BNE   FFC11            ;BR IF NOT.
4415 024626 000444          BR    FFCDONE
4416
4417
4418          ;TEST BUFFER AND DATA:
4419 024630 177777          -1
4420 024632 177777          FFCTP1: -1
4421 024634 177777 177777 177777  -1,-1,-1
4422 024642 024632 177777 177777  FFCTP2: FFCTP1,-1,-1,-1
4423 024650 177777
4424
4425          ;REPORT R0 INCORRECT.
4426 024652 012737 024642 001240  FFC10: MOV    #FFCTP2,@#$TMP3
4427 024660 010037 001242      MOV    R0,@#$TMP4
4428 024664 104241          1$:    ER:OR  +241          ;R0 BAD BUT FSRC FAILED
4429 024666 000424          BR    FFCDONE
```

```

4430
4431
4432 024670 012737 045412 001240 :REPORT FPS INCORRECT.
4433 024676 010537 001242 FFC11: MOV #45412,@#$TMP3 ;REPORT FPS INCORRECT.
4434 024702 104242 1$: MOV R5,@#$TMP4
4435 024704 000415 BR +242
4436 :TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4437 :EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4438 :FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4439 024706 011602 FFC20: MOV (SP),R2
4440 024710 020227 024610 CMP R2,#FFC2+2
4441 024714 001405 BEQ 1$
4442 024716 020227 024612 CMP R2,#FFC2+4
4443 024722 001402 BEQ 1$
4444 024724 000137 046310 JMP @#CPSPUR
4445 024730 022626 1$: CMP (SP)+,(SP)+
4446 024732 010237 001236 MOV R2,@#$TMP2
4447 024736 104243 2$: ERROR +243 ;ODD ADRES
4448 024740 104412 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?).

```

```

4449
:*****
:*TEST 52 SOURCE MODES, MODE 6 (FL=0), TEST
:*
:* THIS IS A TEST OF SOURCE MODE 6
:* USING THE LDFPS INSTR
:*
:*****

```

```

4450 024742 000004 TST52: SCOPE
4451 024744 104413 GGC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4452 024746 012700 017635 MOV #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER.
4453 024752 012737 046543 025036 MOV #46543,@#GGCTP1
4454 024760 012737 046543 001240 MOV #46543,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
4455 024766 012737 025004 001236 MOV #GGC2,@#$TMP2
4456 024774 005001 CLR R1
4457 024776 012737 025124 000004 GGC2: MOV #GGC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4458 025004 170160 005201 LDFPS 5201(R0) ;TEST INSTRUCTION.
4459 025010 170204 STFPS R4 ;GET THE FPS.
4460 025012 005701 TST R1 ;WAS PC CORRECT AFTER EXECUTION?
4461 025014 001033 BNE GGC25 ;BR IF NOT.
4462 025016 020027 017635 CMP R0,#GGCTP1-5201 ;IS R0 CORRECT?
4463 025022 001012 BNE GGC10 ;BR IF NOT.
4464 025024 022704 046543 CMP #46543,R4 ;IS THE FPS CORRECT?
4465 025030 001016 BNE GGC11 ;BR IF NOT.
4466 025032 000451 BR GGCDONE

```

```

4467
4468 ;TEST BUFFER AND DATA:
4469 025034 177777 -1
4470 025036 177777 177777 177777 GGCTP1: -1,-1,-1,-1
4471 025044 177777
4472 025046 177777 -1

```

```

4473          :REPORT RO INCORRECT.
4474 025050 012737 017635 001240 GGC10: MOV #GGCTP1-5201,@#$TMP3
4475 025056 010037 001242          MOV R0,@#$TMP4
4476 025062 104244          1$: ERROR +244 ;RO BAD BUT FSRC FAILED
4477 025064 000434          BR GGCDONE
4478
4479          :REPORT FPS INCORRECT.
4480 025066 012737 046543 001240 GGC11: MOV #46543,@#$TMP3 ;REPORT FPS INCORRECT.
4481 025074 010437 001242          MOV R4,@#$TMP4
4482 025100 104245          1$: ERROR +245
4483 025102 000425          BR GGCDONE
4484
4485          :REPORT PC INCORRECT AFTER INSTRUCTION.
4486 025104 012737 025010 001240 GGC25: MOV #GGC2+4,@#$TMP3
4487 025112 012737 025006 001242          MOV #GGC2+2,@#$TMP4
4488 025120 104246          1$: ERROR +246 ;PC X
4489 025122 000415          BR GGCDONE
4490          :TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4491          :EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4492          :FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4493 025124 011602          GGC20: MOV (SP),R2
4494 025126 020227 025006          CMP R2,#GGC2+2
4495 025132 001405          BEQ 1$
4496 025134 020227 025010          CMP R2,#GGC2+4
4497 025140 001402          BEQ 1$
4498 025142 000137 046310          JMP @#CPSPUR
4499 025146 022626          1$: CMP (SP)+,(SP)+
4500 025150 010237 001236          MOV R2,@#$TMP2
4501 025154 104247          2$: ERROR +247 ;ODD ADRES
4502 025156          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?).
    
```

```

4503          :*****
          :*TEST 53 SOURCE MODES, MODE 7 (FL=0), TEST
          :*
          :* THIS IS A TEST OF SOURCE MODE 7
          :* USING THE LDFPS INSTR
          :*
          :*****
    
```

```

4504 025160 000004          TST53: SCOPE
          025162          HHC1:
          025162 104413          LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4505 025164 012700 020071          MOV #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.
4506 025170 012760 025262 005201          MOV #HHCTP1,5201(R0)
4507 025176 012737 004547 025262          MOV #4547,@#HHCTP1
4508 025204 012737 004547 001240          MOV #4547,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
4509 025212 012737 025230 001236          MOV #HHC2,@#$TMP2
4510 025220 005001          CLR R1
4511 025222 012737 025356 000004          MOV #HHC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
4512 025230 170170 005201          HHC2: LDFPS @5201(R0) ;TEST INSTRUCTION.
4513 025234 170204          STFPS R4 ;GET THE FPS.
4514 025236 005701          TST R1 ;WAS PC CORRECT AFTER EXECUTION?
4515 025240 001036          BNE HHC25 ;BR IF NOT.
4516 025242 020027 020071          CMP R0,#HHCTP2-5201 ;IS RO CORRECT?
    
```

```

4517 025246 001015          BNE      HHC10          ;BR IF NOT.
4518 025250 022704 004547  CMP      #4547,R4      ;IS THE FPS CORRECT?
4519 025254 001021          BNE      HHC11          ;BR IF NOT.
4520 025256 000454          BR       HHCDONE
4521
4522
4523          ;TEST BUFFER AND DATA:
4524 025260 177777          -1
4525 025262 177777 177777 177777 HHCTP1: .WORD -1,-1,-1,-1
         025270 177777
4526 025272 177777 177777 177777 HHCTP2: .WORD -1,-1,-1,-1
         025300 177777
4527
4528          ;REPORT R0 INCORRECT.
4529 025302 012737 020071 001240 HHC10: MOV      #HHCTP2-5201,@#$TMP3
4530 025310 010037 001242          MOV      R0,@#$TMP4
4531 025314 104250          1$:      ERROR   +250          ;R0 BAD BUT FSRC FAILED
4532 025316 000434          BR       HHCDONE
4533
4534          ;REPORT FPS INCORRECT.
4535 025320 012737 004547 001240 HHC11: MOV      #4547,@#$TMP3 ;REPORT FPS INCORRECT.
4536 025326 010437 001242          MOV      R4,@#$TMP4
4537 025332 104251          1$:      ERROR   +251
4538 025334 000425          BR       HHCDONE
4539
4540          ;REPORT PC INCORRECT AFTER INSTRUCTION.
4541 025336 012737 025234 001240 HHC25: MOV      #HHC2+4,@#$TMP3
4542 025344 012737 025232 001242          MOV      #HHC2+2,@#$TMP4
4543 025352 104252          1$:      ERROR   +252          ;PC X
4544 025354 000415          BR       HHCDONE
4545          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
4546          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
4547          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
4548 025356 011602          HHC20: MOV      (SP),R2
4549 025360 020227 025232          CMP      R2,#HHC2+2
4550 025364 001405          BEQ     1$
4551 025366 020227 025234          CMP      R2,#HHC2+4
4552 025372 001402          BEQ     1$
4553 025374 000137 046310          JMP     @#CPSPUR
4554 025400 022626          1$:      CMP      (SP)+,(SP)+
4555 025402 010237 001236          MOV      R2,@#$TMP2
4556 025406 104253          2$:      ERROR   +253          ;DDD ADDRESS
4557 025410          HHCDONE:
         025410 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?).

```

4558
 4559
 4566
 4567

```

*****
*TEST 54          SOURCE MODES, MODE 2 GR7 (FL=1), TEST
*
* THIS IS A TEST OF THE LDCLD WITH
* IMMEDIATE ADDRESSING MODE
*
```

```
025412 000004          :*****  
4568 025412 000004 TST54: SCOPE  
4569 025414          IIC1:  
4570 025414 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.  
4571 025416 012737 025442 001236 MOV #IIC2,@#$TMP2 ;SAVE DATA IN CASE OF ERROR.  
4572 025424 012737 025514 000004 MOV #IIC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.  
4573 025432 012700 000300 MOV #300,R0  
4574 025436 170100          LDFPS R0  
4575 025440 005001          CLR R1  
4576 025442 177027          IIC2: LDCLD (R7)+,ACO ;TEST INSTRUCTION.  
4577 025444 005201          5201  
4578 025446 005201          5201  
4579 025450 005201          5201  
4580 025452 005201          5201  
4581 025454 020127 000003          CMP R1,#3 ;WAS PC CORRECT AFTER EXECUTION?  
4582 025460 001421          BEQ IICDONE ;BR IF YES.  
4583  
4584  
4585  
4586          ;REPORT PC INCORRECT AFTER INSTRUCTION.  
4587 025462 012704 025446          IIC3: MOV #IIC2+4,R4  
4588 025466 162701 000003          SUB #3,R1  
4589 025472 006301          ASL R1  
4590 025474 160104          SUB R1,R4  
4591 025476 010437 001242          MOV R4,@#$TMP4  
4592 025502 012737 025446 001240          MOV #IIC2+4,@#$TMP3  
4593 025510 104254          1$: ERROR +254 ;BAD CONSTANT  
4594 025512 000404          BR IICDONE  
4595          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING  
4596          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT  
4597          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.  
4598 025514 011637 001236          IIC20: MOV (SP),@#$TMP2  
4599 025520 022626          CMP (SP)+,(SP)+  
4600 025522 104255          1$: ERROR +255 ;BAD CONSTANT ODD ADD  
4601  
4602 025524          IICDONE:  
4603 025524 104412          RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
4604          ;SEE IF THE USER HAS EXPRESSED  
4605          ;THE DESIRE TO CHANGE THE SOFTWARE  
4606          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
4607          ;THE USER TYPED CONTROL G?).  
4608  
4609  
4610  
4611
```

```
:*****  
:*TEST 55 SOURCE MODES, MODE 2 (FL=1), TEST  
:*  
:* THIS IS A TEST OF THE LDCLD INSTR  
:* WITH MODE 2.  
:*  
:*****
```

```
025526 000004 TST55: SCOPE  
4612 025530          TCC1:  
4613 025530 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.  
4614 025532 016737 000014 001236 MOV TCC2,@#$TMP2 ;SAVE DATA IN CASE OF ERROR.
```



```

4615 025540 012700 000300      MOV      #300,R0
4616 025544 170100      LDFPS   R0
4617 025546 012700 025642      MOV      #TCCBF0,R0      ;SET UP THE TEST DATA BUFFER.
4618 025552 177020      TCC2:   LDCLD   (R0)+,ACO      ;TEST INSTRUCTION.
4619
4620 025554 170204      STFPS   R4      ;GET THE FPS.
4621 025556 012701 025652      MOV      #TCCBF1,R1      ;GET THE RESULT.
4622 025562 012702 000200      MOV      #200,R2
4623 025566 170102      LDFPS   R2
4624 025570 174011      STD     ACO,(R1)
4625 025572 020027 025646      CMP      R0,#TCCBF0+4      ;IS R0 CORRECT?
4626 025576 001407      BEQ     TCC3
4627      ;REPORT R0 INCORRECT.
4628 025600 010037 001242      MOV      R0,@#$TMP4
4629 025604 012737 025646 001240      MOV      #TCCBF0+4,@#$TMP3
4630 025612 104256      1$:     ERROR  +256      ;BAD CONST
4631 025614 000422      BR      TCCDONE
4632
4633 025616 022704 000300      TCC3:   CMP      #300,R4      ;IS THE FPS CORRECT?
4634 025622 001417      BEQ     TCCDONE
4635
4636      ;REPORT FPS INCORRECT.
4637 025624 010437 001242      MOV      R4,@#$TMP4
4638 025630 012737 000300 001240      MOV      #300,@#$TMP3
4639 025636 104257      1$:     ERROR  +257      ;FPS X
4640 025640 000410      BR      TCCDONE
4641
4642
4643      ;TEST BUFFER AND DATA:
4644 025642 001234 067076 054321  TCCBF0: .WORD  01234,67076,54321,012345
4645 025650 012345
4645 025652 177777 177777 177777  TCCBF1: -1,-1,-1,-1
4645 025660 177777
4646
4647 025662      TCCDONE:
4647 025662 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?).

4648
4649
4656
4657      ;*****
;*TEST 56      LDCIF AND LDCLF TEST
;*
;* THIS IS A TEST OF THE LDCIF AND
;* THE LDCLF INSTRUCTIONS.
;*
;*****
4658 025664 000004      TST56: SCOPE
4659
4660      ;ZERO  OPERAND FL=0
4661
4662 025666      KKC1:
4662 025666 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.

```



```

4695
4696 025770 000000 000000 1$: .WORD 0,0 ;FSRC OPERAND.
4697 025774 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
4698 026000 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
4699 026004 000100 4$: 100 ;FPS BEFORE EXECUTION.
4700 026006 000104 104 ;FPS AFTER EXECUTION.
4701 026010 000004 4 ;ANTICIPATED ERRONEOUS FPS.
4702 026012 104260 5$: ERROR +260 ;REPORT RESULT INCORRECT.
4703 026014 000401 BR 6$
4704 026016 104263 ERROR +263 ;FL WAS CLR'ED
4705 026020 6$:
4706 ;OPERAND POSITIVE FL=0
4707 026020 104413 KKC4:
4708 026022 004737 027020 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4709 026026 040000 000000 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
4710 026032 043600 000000 1$: .WORD 40000,0 ;FSRC OPERAND.
4711 026036 047600 000000 2$: .WORD 43600,0 ;EXPECTED RESULT.
4712 026042 000017 3$: .WORD 47600,0 ;ANTICIPATED ERRONEOUS RESULT.
4713 026044 000000 4$: 17 ;FPS BEFORE EXECUTION.
4714 026046 177777 0 ;FPS AFTER EXECUTION.
4715 026050 104264 -1 ;ANTICIPATED ERRONEOUS FPS.
4716 026052 000401 5$: ERROR +264 ;ST 107 BAD
4717 026054 104261 BR 6$ ;CONSTANT 231 INSD
4718 026056 ERROR +261 ;215
4719 6$:
4720 026056 104413 KKC5:
4721 026060 004737 027020 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4722 026064 000001 000000 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
4723 026070 040200 000000 1$: .WORD 1,0 ;FSRC OPERAND.
4724 026074 044200 000000 2$: .WORD 40200,0 ;EXPECTED RESULT.
4725 026100 000017 3$: .WORD 44200,0 ;ANTICIPATED ERRONEOUS RESULT.
4726 026102 000000 4$: 17 ;FPS BEFORE EXECUTION.
4727 026104 177777 0 ;FPS AFTER EXECUTION.
4728 026106 104264 -1 ;ANTICIPATED ERRONEOUS FPS.
4729 026110 000401 5$: ERROR +264 ;REPORT RESULT INCORRECT.
4730 026112 104261 BR 6$
4731 026114 ERROR +261 ;REPORT FPS INCORRECT.
4732 6$:
4733 ;OPERAND= PATTERN FL=0
4734 KKC6:
4735 026114 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4736 026116 004737 027020 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
4737 026122 000252 000000 1$: .WORD 252,0 ;FSRC OPERAND.
4738 026126 042052 000000 2$: .WORD 42052,0 ;EXPECTED RESULT.
4739 026132 046052 000000 3$: .WORD 46052,0 ;ANTICIPATED ERRONEOUS RESULT.
4740 026136 000000 4$: 0 ;FPS BEFORE EXECUTION.
4741 026140 000000 0 ;FPS AFTER EXECUTION.
4742 026142 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
4743 026144 104264 5$: ERROR +264 ;REPORT RESULT INCORRECT.
4744 026146 000401 BR 6$
4745 026150 104261 ERROR +261 ;REPORT FPS INCORRECT.
4746 026152 6$:
4747 ;OPERAND=-40000 FL=0
4748

```

4749 026152
4750 026154 104413
4751 026160 004737 027020
4752 026164 140000 000000
4753 026170 143600 000000
4754 026174 043600 000000
4755 026176 000007
4756 026176 000010
4757 026200 177777
4758 026202 104265
4759 026204 000401
4760 026206 104261
4761
4762
4763 026210
4764 026210 104413
4765 026212 004737 027020
4766 026216 177777 000000
4767 026222 140200 000000
4768 026226 144000 000400
4769 026232 000000
4770 026234 000010
4771 026236 177777
4772 026240 104266
4773 026242 000401
4774 026244 104261
4775
4776
4777 026246
4778 026246 104413
4779 026250 004737 027020
4780 026254 125252 000000
4781 026260 143652 126000
4782 026264 043652 126000
4783 026270 000007
4784 026272 000010
4785 026274 177777
4786 026276 104265
4787 026300 000401
4788 026302 104261
4789
4790
4791 026304
4792 026304 104413
4793 026306 004737 027020
4794 026312 040000 000000
4795 026316 047600 000000
4796 026322 043600 000000
4797 026326 000117
4798 026330 000100
4799 026332 177777
4800 026334 104267
4801 026336 000401
4801 026340 104261

KKC7:
LPERR
JSR PC,@#LDCFSUB
1\$: .WORD -4000,0
2\$: .WORD 143600,0
3\$: .WORD 43600,0
4\$: 7
10
-1
5\$: ERROR +265
BR 6\$
ERROR +261
6\$:
;OPERAND=-1 FL=0
KKC8:
LPERR
JSR PC,@#LDCFSUB
1\$: .WORD -1,0
2\$: .WORD 140200,0
3\$: .WORD 144000,400
4\$: 0
10
-1
5\$: ERROR +266
BR 6\$
ERROR +261
6\$:
;OPERAND=PATTERN FL=0
KKC9:
LPERR
JSR PC,@#LDCFSUB
1\$: .WORD 125252,0
2\$: .WORD 143652,126000
3\$: .WORD 43652,126000
4\$: 7
10
-1
5\$: ERROR +265
BR 6\$
ERROR +261
6\$:
;OPERAND POS FL-1
KKC10:
LPERR
JSR PC,@#LDCFSUB
1\$: .WORD 4000,0
2\$: .WORD 47600,0
3\$: .WORD 43600,0
4\$: 117
100
-1
5\$: ERROR +267 ;ST 107
BR 6\$
ERROR +261

;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;(SET SIGN) ST 146
;REPORT FPS INCORRECT.
;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;ST 372 TO 152 INTO
;112 (BUF XNBT)
;REPORT FPS INCORRECT.
;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;REPORT RESULT INCORRECT.
;REPORT FPS INCORRECT.
;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;CONSTANT
;BAD 237 INST 217
;REPORT FPS INCORRECT.

```

4802 026342
4803
4804
4805 026342
      026342 104413
4806 026344 004737 027020
4807 026350 000000 000001
4808 026354 040200 000000
4809 026360 034200 000000
4810 026364 000100
4811 026366 000100
4812 026370 177777
4813 026372 104267
4814 026374 000401
4815 026376 104261
4816 026400
4817
4818
4819 026400
      026400 104413
4820 026402 004737 027020
4821 026406 000000 000252
4822 026412 042052 000000
4823 026416 036052 000000
4824 026422 000111
4825 026424 000100
4826 026426 177777
4827 026430 104267
4828 026432 000401
4829 026434 104261
4830 026436
4831
4832
4833 026436
      026436 104413
4834 026440 004737 027020
4835 026444 140000 000000
4836 026450 147600 000000
4837 026454 047600 000000
4838 026460 000107
4839 026462 000110
4840 026464 177777
4841 026466 104265
4842 026470 000401
4843 026472 104261
4844 026474
4845
4846
4847 026474
      026474 104413
4848 026476 004737 027020
4849 026502 177777 177777
4850 026506 140200 000000
4851 026512 150000 000000
4852 026516 000100
4853 026520 000110
4854 026522 177777
    
```

```

6$:
;OPERAND=1      FL=1
KKC11:
      LPERR
      JSR      PC,@#LDCFSUB
1$:      .WORD 0,1
2$:      .WORD 40200,0
3$:      .WORD 34200,0
4$:      100
          100
          -1
5$:      ERROR +267
          BR      6$
          ERROR +261
6$:
;OPERAND=      PATTERN FL=1
KKC12:
      LPERR
      JSR      PC,@#LDCFSUB
1$:      .WORD 0,252
2$:      .WORD 42052,0
3$:      .WORD 36052,0
4$:      111
          100
          -1
5$:      ERROR +267
          BR      6$
          ERROR +261
6$:
;OPERAND=-40000,0  FL=1
KKC13:
      LPERR
      JSR      PC,@#LDCFSUB
1$:      .WORD -40000,0
2$:      .WORD 147600,0
3$:      .WORD 47600,0
4$:      107
          110
          -1
5$:      ERROR +265 ;SET SIGN
          BR      6$
          ERROR +261
6$:
;OPERAND=-1,-1  FL=1
KKC14:
      LPERR
      JSR      PC,@#LDCFSUB
1$:      .WORD -1,-1
2$:      .WORD 140200,0
3$:      .WORD 150000,0
4$:      100
          110
          -1
    
```

```

;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;REPORT RESULT INCORRECT.

;REPORT FPS INCORRECT.

;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;REPORT RESULT INCORRECT.

;REPORT FPS INCORRECT.

;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;REPORT FPS INCORRECT.

;SET UP THE LOOP ON ERROR ADDRESS.
;GO EXECUTE THE INSTRUCTION.
;FSRC OPERAND.
;EXPECTED RESULT.
;ANTICIPATED ERRONEOUS RESULT.
;FPS BEFORE EXECUTION.
;FPS AFTER EXECUTION.
;ANTICIPATED ERRONEOUS FPS.
;REPORT FPS INCORRECT.
    
```

4855	026524	104266		5\$:	ERROR	+266		;(BUT XNBT)
4856	026526	000401			BR	6\$		
4857	026530	104261			ERROR	+261		;REPORT FPS INCORRECT.
4858	026532			6\$:				
4859								
4860								
4861	026532							
	026532	104413						
4862	026534	004737	027020		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
4863	026540	125252	125252		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
4864	026544	147652	125253	1\$:	.WORD	125252,125252		;FSRC OPERAND.
4865	026550	047652	125253	2\$:	.WORD	147652,125253		;EXPECTED RESULT.
4866	026554	000105		3\$:	.WORD	47652,125253		;ANTICIPATED ERRONEOUS RESULT.
4867	026556	000110		4\$:	105			;FPS BEFORE EXECUTION.
4868	026560	177777			110			;FPS AFTER EXECUTION.
4869	026562	104265			-1			;ANTICIPATED ERRONEOUS FPS.
4870	026564	000401		5\$:	ERROR	+265		;REPORT RESULT INCORRECT.
4871	026566	104261			BR	6\$		
4872	026570				ERROR	+261		;REPORT FPS INCORRECT.
4873				6\$:				
4874								
4875	026570							
	026570	104413						
4876	026572	004737	027020		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
4877	026576	077777	177500		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
4878	026602	047777	177777	1\$:	.WORD	77777,177500		;FSRC OPERAND.
4879	026606	047777	177776	2\$:	.WORD	47777,177777		;EXPECTED RESULT.
4880	026612	000117		3\$:	.WORD	47777,177776		;ANTICIPATED ERRONEOUS RESULT.
4881	026614	000100		4\$:	117			;FPS BEFORE EXECUTION.
4882	026616	177777			100			;FPS AFTER EXECUTION.
4883	026620	104270			-1			;ANTICIPATED ERRONEOUS FPS.
4884	026622	000401		5\$:	ERROR	+270		;ST 631 INTO RND
4885	026624	104261			BR	6\$		
4886	026626				ERROR	+261		;REPORT FPS INCORRECT.
4887				6\$:				
4888								
4889	026626							
	026626	104413						
4890	026630	004737	027020		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
4891	026634	040000	000100		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
4892	026640	047600	000001	1\$:	.WORD	40000,100		;FSRC OPERAND.
4893	026644	047600	000000	2\$:	.WORD	47600,1		;EXPECTED RESULT.
4894	026650	000102		3\$:	.WORD	47600,0		;ANTICIPATED ERRONEOUS RESULT.
4895	026652	000100		4\$:	102			;FPS BEFORE EXECUTION.
4896	026654	177777			100			;FPS AFTER EXECUTION.
4897	026656	104270			-1			;ANTICIPATED ERRONEOUS FPS.
4898	026660	000401		5\$:	ERROR	+270		;REPORT RESULT INCORRECT.
4899	026662	104261			BR	6\$		
4900	026664				ERROR	+261		;REPORT FPS INCORRECT.
4901				6\$:				
4902								
4903	026664							
	026664	104413						
4904	026666	004737	027020		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
4905	026672	040000	000100		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
4906	026676	047600	000000	1\$:	.WORD	40000,100		;FSRC OPERAND.
4907	026702	047600	000001	2\$:	.WORD	47600,0		;EXPECTED RESULT.
				3\$:	.WORD	47600,1		;ANTICIPATED ERRONEOUS RESULT.

```

4908 026706 000157      4$:      157      ;FPS BEFORE EXECUTION.
4909 026710 000140      ;FPS AFTER EXECUTION.
4910 026712 177777      ;ANTICIPATED ERRONEOUS FPS.
4911 026714 104271      5$:      ERROR    +271    ;ST 631 ... INTO TRNC
4912 026716 000401      BR        6$
4913 026720 104261      ERROR    +261    ;REPORT FPS INCORRECT.
4914 026722
4915
4916 026722      6$:
      ;OPERAND=100000,0 (MOST NEG #) FL=0
      KKC19:
      LPERR
      JSR    PC,@#LDCFSUB    ;SET UP THE LOOP ON ERROR ADDRESS.
      ;GO EXECUTE THE INSTRUCTION.
4917 026724 004737 027020      1$:      .WORD    100000,0    ;FSRC OPERAND.
4918 026730 100000 000000      2$:      .WORD    144000,0    ;EXPECTED RESULT.
4919 026734 144000 000000      3$:      .WORD    143600,0    ;ANTICIPATED ERRONEOUS RESULT.
4920 026740 143600 000000      4$:      7
4921 026744 000007      ;FPS BEFORE EXECUTION.
4922 026746 000010      ;FPS AFTER EXECUTION.
4923 026750 177777      ;ANTICIPATED ERRONEOUS FPS.
4924 026752 104272      5$:      ERROR    +272    ;ST 630 RH*R14+1
4925 026754 000401      BR        6$
4926 026756 104261      ERROR    +261    ;REPORT FPS INCORRECT.
4927 026760
4928
4929      6$:
      ;OPERAND=100000,0      FL=1
      KKC20:
      LPERR
      JSR    PC,@#LDCFSUB    ;SET UP THE LOOP ON ERROR ADDRESS.
      ;GO EXECUTE THE INSTRUCTION.
4931 026762 004737 027020      1$:      .WORD    100000,0    ;FSRC OPERAND.
4932 026766 100000 000000      2$:      .WORD    150000,0    ;EXPECTED RESULT.
4933 026772 150000 000000      3$:      .WORD    147600,0    ;ANTICIPATED ERRONEOUS RESULT.
4934 026776 147600 000000      4$:      107
4935 027002 000107      ;FPS BEFORE EXECUTION.
4936 027004 000110      ;FPS AFTER EXECUTION.
4937 027006 177777      ;ANTICIPATED ERRONEOUS FPS.
4938 027010 104272      5$:      ERROR    +272    ;REPORT RESULT INCORRECT.
4939 027012 000401      BR        6$
4940 027014 104261      ERROR    +261
4941 027016 000506      BR        KKCDONE    ;REPORT FPS INCORRECT.
4942
4943      ;THIS SUBROUTINE, LDCFSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
4944      ;THE LDCIF OR LDCLF INSTRUCTION AND CHECK THE RESULTS. A CALL
4945      ;TO IT IS MADE THUS:
4946      :
4947      :      JSR    PC,@#LDCFSUB
4948      :      ACARG:  .WORD    X,X      ;AC OPERAND
4949      :      RES:    .WORD    X,X      ;EXPECTED RESULT
4950      :      ERRES:  .WORD    X,X      ;ERROR RESULT
4951      :      FPSB:   .WORD    X        ;FPS BEFORE EXECUTION
4952      :      FPSA:   .WORD    X        ;FPS AFTER EXECUTION
4953      :      ERFPS:  .WORD    X        ;ERROR FPS
4954      :      ERR1:   ERROR    +X      ;DATA ERROR
4955      :      BR      CONT
4956      :      ERR2:   ERROR    +X      ;FPS ERROR
4957      :      CONT:   ;RETURN ADDRESS
4958
4959      ;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
4960      ;THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED.
4961      ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
4962      ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL
  
```

```

4963 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCFSUB WILL
4964 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCFSUB WILL RETURN
4965 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF
4966 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
4967 ;LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
4968 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
4969 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB
4970 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
4971 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
4972 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
4973
4974 027020 012601 LDCFSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
4975 027022 016100 000014 MOV 14(R1),R0 ;SET THE FPS.
4976 027026 170100 LDFPS R0
4977 027030 012737 027040 001236 MOV #1$,@#$TMP2
4978 027036 010100 MOV R1,R0
4979
4980 027040 177010 1$: LDCIF (R0),AC0 ;TEST INSTRUCTION LDCIF OR LDCLF.
4981
4982 027042 170204 STFPS R4 ;GET FPS.
4983 027044 012700 027224 MOV #LDCT,R0 ;GET THE RESULT.
4984 027050 012702 000200 MOV #200,R2
4985 027054 170102 LDFPS R2
4986 027056 174010 STD AC0,(R0)
4987
4988 027060 012702 027224 MOV #LDCT,R2 ;SEE IF THE RESULT WAS CORRECT.
4989 027064 010237 001242 MOV R2,@#$TMP4
4990 027070 010137 001240 MOV R1,@#$TMP3
4991 027074 010103 MOV R1,R3
4992 027076 062703 000004 ADD #4,R3
4993 027102 010337 001244 MOV R3,@#$TMP5
4994 027106 010437 001250 MOV R4,@#$TMP7
4995 027112 016137 000016 001252 MOV 16(R1),@#$TMP10
4996 027120 010100 MOV R1,R0
4997 027122 062700 000004 ADD #4,R0
4998 027126 012703 000002 MOV #2,R3
4999 027132 022022 2$: CMP (R0)+,(R2)+
5000 027134 001006 BNE 10$ ;BR IF INCORRECT.
5001 027136 077303 SOB R3,2$
5002
5003 027140 026104 000016 CMP 16(R1),R4 ;SEE IF THE FPS WAS CORRECT.
5004 027144 001020 BNE 15$ ;BR IF INCORRECT.
5005 027146 000161 000030 3$: JMP 30(R1) ;RETURN.
5006
5007 ;RESULT IN CORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
5008 027152 012702 027224 10$: MOV #LDCT,R2
5009 027156 010100 MOV R1,R0
5010 027160 062700 000010 ADD #10,R0
5011 027164 012703 000002 MOV #2,R3
5012 027170 022022 11$: CMP (R0)+,(R2)+
5013 027172 001003 BNE 13$
5014 027174 077303 SOB R3,11$
5015 027176 000161 000022 JMP 22(R1)
5016
5017 ;THE FAILURE WAS NOT ANTICIPATED SO REPORT THE ERROR HERE.
5018 027202 13$:
5019

```


5020 027202 104260
5021 027204 000760
5022
5023
5024
5025 027206 026104 000020
5026 027212 001002
5027 027214 000161 000026
5028
5029
5030 027220
5031 027220 104261
5032 027222 000751
5033
5034
5035 027224 000000 000000 000000
027232 000000
5036
5037 027234
027234 104412

14\$: ERROR +260 ;BAD RES
BR 3\$
:THE FPS WAS INCORRECT SO SEE IF IT WAS ANTICIPATED.
15\$: CMP 20(R1),R4
BNE 16\$
JMP 26(R1)

:FPS ERROR NOT ANTICIPATED SO REPORT IT HERE.
16\$:
17\$: ERROR +261 ;BAD FPS
BR 3\$

:DATA BUFFER:
LDCT: .WORD 0,0,0,0

KKCDONE:
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?).

5038

5045
5046

```
*****
*TEST 57      LDCID AND LDCLD TEST
*
* THIS IS A TEST OF LDCID AND LDCLD
*
*****
```

5047 027236 000004

```
TST57: SCOPE
;OPERAND=0      FL=0,  FD=1
```

5048

LLC1:

5049 027240 104413

LPERR

;SET UP THE LOOP ON ERROR ADDRESS.

5050 027242 004737 030036

JSR

PC,@#LDCDSUB

;GO EXECUTE THE INSTRUCTION.

5051 027246 000000 000000

1\$:

.WORD

0,0

;FSRC OPERAND.

5052 027252 000000 000000 000000

2\$:

.WORD

0,0,0,0

;EXPECTED RESULT.

5053 027260 000000

5054 027262 177777 177777 177777

3\$:

.WORD

-1,-1,-1,-1

;ANTICIPATED ERRONEOUS RESULT.

5055 027270 177777

5056 027272 000213

4\$:

213

;FPS BEFORE EXECUTION.

5057 027274 000204

204

;FPS AFTER EXECUTION.

5058 027276 177777

-1

;ANTICIPATED ERRONEOUS FPS.

5059 027300 104273

5\$:

ERROR

+273

;REPORT RESULT INCORRECT.

5060 027302 000401

BR

6\$

5061 027304 104274

ERROR

+274

;REPORT FPS INCORRECT.

5062 027306

6\$:

```
;OPERAND=0      FL=0,  FD=1
```

5063 027306 104413

LLC2:

LPERR

;SET UP THE LOOP ON ERROR ADDRESS.

5064 027310 004737 030036

JSR

PC,@#LDCDSUB

;GO EXECUTE THE INSTRUCTION.

5065 027314 000000 177777

1\$:

.WORD

0,-1

;FSRC OPERAND.

5066 027320 000000 000000 000000

2\$:

.WORD

0,0,0,0

;EXPECTED RESULT.

5067 027326 000000

5068 027330 004177 177400 000000

3\$:

.WORD

4177,177400,0,0

;ANTICIPATED ERRONEOUS RESULT.

5069 027336 000000

5070 027340 000200

4\$:

200

;FPS BEFORE EXECUTION.

5071 027342 000204

204

;FPS AFTER EXECUTION.

5072 027344 177777

-1

;ANTICIPATED ERRONEOUS FPS.

5073 027346 104275

5\$:

ERROR

+275

;(BUT FL)S+277

5074 027350 000401

BR

6\$

;TO 300 INTO 301

5075 027352 104274

ERROR

+274

;REPORT FPS INCORRECT.

5076 027354

6\$:

```
;OPERAND=0      FL=1   FD=1
```

5077 027354 104413

LLC3:

LPERR

;SET UP THE LOOP ON ERROR ADDRESS.

5078 027356 004737 030036

JSR

PC,@#LDCDSUB

;GO EXECUTE THE INSTRUCTION.

5079 027362 000000 000000

1\$:

.WORD

0,0

;FSRC OPERAND.

5080 027366 000000 000000 000000

2\$:

.WORD

0,0,0,0

;EXPECTED RESULT.

5081 027374 000000

5082 027376 177777 177777 177777

3\$:

.WORD

-1,-1,-1,-1

;ANTICIPATED ERRONEOUS RESULT.

5083 027404 177777

5084 027406 000211

4\$:

211

;FPS BEFORE EXECUTION.

5085 027410 000204

204

;FPS AFTER EXECUTION.

5086 027412 177777

-1

;ANTICIPATED ERRONEOUS FPS.

5087 027414 104273

5\$:

ERROR

+273

;REPORT RESULT INCORRECT.

5088 027416 000401

BR

6\$

5089 027420 104274

ERROR

+274

;REPORT FPS INCORRECT.

5090 027422

6\$:

```

5087
5088 ;OPERAND=40000 FL=0 FD=1
5089 027422 LLC4: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
027422 104413 JSR PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
5090 027424 004737 030036 .WORD 40000,0 ;FSRC OPERAND.
5091 027430 040000 000000 1$: .WORD 43600,0,0,0 ;EXPECTED RESULT.
5092 027434 043600 000000 000000 2$: .WORD 47600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
027442 000000
5093 027444 047600 000000 000000 3$: .WORD 217 ;FPS BEFORE EXECUTION.
027452 000000 4$: 200 ;FPS AFTER EXECUTION.
5094 027454 000217 -1 ;ANTICIPATED ERRONEOUS FPS.
5095 027456 000200 5$: ERROR +276 ;ST 107 BAD CONST
5096 027460 177777 BR 6$
5097 027462 104274 ERROR +274 ;REPORT FPS INCORRECT.
5098 027464 000401
5099 027466 104274
5100 027470
5101
5102 ;OPERAND=-40000 FL=0 FD=1
5103 027470 LLC5: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
027470 104413 JSR PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
5104 027472 004737 030036 .WORD -40000,0 ;FSRC OPERAND.
5105 027476 140000 000000 1$: .WORD 143600,0,0,0 ;EXPECTED RESULT.
5106 027502 143600 000000 000000 2$: .WORD 43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
027510 000000
5107 027512 043600 000000 000000 3$: .WORD 200 ;FPS BEFORE EXECUTION.
027520 000000 4$: 210 ;FPS AFTER EXECUTION.
5108 027522 000200 -1 ;ANTICIPATED ERRONEOUS FPS.
5109 027524 000210 5$: ERROR +277 ;(SET SIGN) ST 176
5110 027526 177777 BR 6$
5111 027530 104274 ERROR +274 ;REPORT FPS INCORRECT.
5112 027532 000401
5113 027534 104274
5114 027536
5115
5116 ;OPERAND=40000,0 FL=1 FD=1
5117 027536 LLC6: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
027536 104413 JSR PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
5118 027540 004737 030036 .WORD 40000,0 ;FSRC OPERAND.
5119 027544 040000 000000 1$: .WORD 47600,0,0,0 ;EXPECTED RESULT.
5120 027550 047600 000000 000000 2$: .WORD 43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
027556 000000
5121 027560 043600 000000 000000 3$: .WORD 317 ;FPS BEFORE EXECUTION.
027566 000000 4$: 300 ;FPS AFTER EXECUTION.
5122 027570 000317 -1 ;ANTICIPATED ERRONEOUS FPS.
5123 027572 000300 5$: ERROR +300 ;ST 107 BAD CONS
5124 027574 177777 BR 6$
5125 027576 104300 ERROR +274 ;REPORT FPS INCORRECT.
5126 027600 000401
5127 027602 104274
5128 027604
5129
5130 ;OPERAND=0,1 FL=1 FD=1
5131 027604 LLC7: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
027604 104413 JSR PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
5132 027606 004737 030036 .WORD 0,1 ;FSRC OPERAND.
5133 027612 000000 000001 1$: .WORD

```

5134	027616	040200	000000	000000	2\$:	.WORD	40200,0,0,0		;EXPECTED RESULT.
	027624	000000							
5135	027626	034200	000000	000000	3\$:	.WORD	34200,0,0,0		;ANTICIPATED ERRONEOUS RESULT.
	027634	000000							
5136	027636	000300			4\$:	300			;FPS BEFORE EXECUTION.
5137	027640	000300				300			;FPS AFTER EXECUTION.
5138	027642	177777				-1			;ANTICIPATED ERRONEOUS FPS.
5139	027644	104300			5\$:	ERROR	+300		;REPORT FPS INCORRECT.
5140	027646	000401				BR	6\$		
5141	027650	104274				ERROR	+274		;REPORT FPS INCORRECT.
5142	027652				6\$:				
5143									
5144									
5145	027652								
	027652	104413							
5146	027654	004737	030036			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
5147	027660	077777	177777		1\$:	JSR	PC,@#LDCDSUB		;GO EXECUTE THE INSTRUCTION.
5148	027664	047777	177777	177000	2\$:	.WORD	77777,177777		;FSRC OPERAND.
	027672	000000				.WORD	47777,177777,177000,0		;EXPECTED RESULT.
5149	027674	177777	177777	177777	3\$:	.WORD	-1,-1,-1,-1		;ANTICIPATED ERRONEOUS RESULT.
	027702	177777							
5150	027704	000317			4\$:	317			;FPS BEFORE EXECUTION.
5151	027706	000300				300			;FPS AFTER EXECUTION.
5152	027710	177777				-1			;ANTICIPATED ERRONEOUS FPS.
5153	027712	104273			5\$:	ERROR	+273		;REPORT RESULT INCORRECT.
5154	027714	000401				BR	6\$		
5155	027716	104274				ERROR	+274		;REPORT FPS INCORRECT.
5156	027720				6\$:				
5157									
5158									
5159									
5160	027720								
	027720	104413							
5161	027722	004767	000110			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
5162	027726	177777	177526		1\$:	JSR	PC,LDCDSUB		;GO EXECUTE THE INSTRUCTION.
5163	027732	142052	000000	000000	2\$:	.WORD	-1,-252		;FSRC OPERAND.
	027740	000000				.WORD	142052,0,0,0		;EXPECTED RESULT.
5164	027742	136052	000000	000000	3\$:	.WORD	136052,0,0,0		;ANTICIPATED ERRONEOUS RESULT.
	027750	000000							
5165	027752	000307			4\$:	307			;FPS BEFORE EXECUTION.
5166	027754	000310				310			;FPS AFTER EXECUTION.
5167	027756	177777				-1			;ANTICIPATED ERRONEOUS FPS.
5168	027760	104300			5\$:	ERROR	+300		;REPORT RESULT INCORRECT.
5169	027762	000401				BR	6\$		
5170	027764	104274				ERROR	+274		;REPORT FPS INCORRECT.
5171	027766				6\$:				
5172									
5173									
5174	027766								
	027766	104413							
5175	027770	004767	000042			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
5176	027774	012345	067012		1\$:	JSR	PC,LDCDSUB		;GO EXECUTE THE INSTRUCTION.
5177	030000	047247	025560	050000	2\$:	.WORD	12345,67012		;FSRC OPERAND.
	030006	000000				.WORD	47247,025560,050000,0		;EXPECTED RESULT.
5178	030010	177777	177777	177777	3\$:	.WORD	-1,-1,-1,-1		;ANTICIPATED ERRONEOUS RESULT.
	030016	177777							
5179	030020	000352			4\$:	352			;FPS BEFORE EXECUTION.

5180 030022 000340
5181 030024 177777
5182 030026 104273
5183 030030 000401
5184 030032 104274
5185 030034 000502

340 :FPS AFTER EXECUTION.
-1 :ANTICIPATED ERRONEOUS FPS.
5\$: ERROR +273 :REPORT RESULT INCORRECT.
BR 6\$
ERROR +274 :REPORT FPS INCORRECT.
6\$: BR LLCDONE

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

5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217

```

JSR PC,@#LDCDSUB
ACARG: .WORD 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
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 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 LDCDSUB RETURNS CONTROL
:TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCDSUB
:COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCDSUB WILL RETURN
:TO THE ERROR CALL AT ERR2, OTHERWISE LDCDSUB 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, ERRES. IF THE FAILURE IN
:THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCDSUB
:WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
:RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCDSUB WILL
:REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

5218 030036 012601
5219 030040 016100 000024
5220 030044 170100
5221 030046 012737 030056 001236
5222 030054 010100
5223 030056 177010
5224
5225 030060 170204
5226 030062 012700 027224
5227 030066 012702 000200
5228 030072 170102
5229 030074 174010
5230
5231
5232 030076 012702 027224
5233 030102 010237 001242
5234 030106 010137 001240
5235 030112 010103
5236 030114 062703 000004

```

LDCDSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
MOV 24(R1),R0 ;SET THE FPS.
LDFPS R0
MOV #1$,@#$TMP2
MOV R1,R0
1$: LDCID (R0),ACO ;TEST INSTRUCTION, LDCID OR LDCLD.

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

;SEE IF THE RESULT IS CORRECT.
MOV #LDCT,R2
MOV R2,@#$TMP4
MOV R1,@#$TMP3
MOV R1,R3
ADD #4,R3
    
```

```

5237 030120 010337 001244      MOV      R3,@#$TMP5
5238 030124 010437 001250      MOV      R4,@#$TMP7
5239 030130 016137 000026 001252      MOV      26(R1),@#$TMP10
5240 030136 010100      MOV      R1,R0
5241 030140 062700 000004      ADD      #4,R0
5242 030144 012703 000002      MOV      #2,R3
5243 030150 022022      2$:     CMP      (R0)+,(R2)+
5244 030152 001006      BNE      10$           ;BR IF INCORRECT.
5245 030154 077303      SOB      R3,2$
5246
5247 030156 026104 000026      CMP      26(R1),R4           ;IS THE FPS CORRECT?
5248 030162 001020      BNE      15$           ;BR IF INCORRECT.
5249 030164 000161 000040      3$:     JMP      40(R1)           ;RETURN.
5250
5251      ;THE RESULT WAS INCORRECT SO SEE IF THE ERROR WAS ANTICIPATED.
5252 030170 012702 027224      10$:    MOV      #LDCT,R2
5253 030174 010100      MOV      R1,R0
5254 030176 062700 000014      ADD      #14,R0
5255 030202 012703 000002      MOV      #2,R3
5256 030206 022022      11$:    CMP      (R0)+,(R2)+
5257 030210 001003      BNE      13$
5258 030212 077303      SOB      R3,11$
5259 030214 000161 000032      JMP      32(R1)
5260 030220
5261      ;ERROR NOT ANTICIPATED SO REPORT RESULT INCORRECT HERE.
5262 030220 104273      14$:    ERROR   +273           ;BAD RES
5263 030222 000760      BR       3$
5264
5265      ;THE FPS WAS INCORRECT. SEE IF FAILURE WAS ANTICIPATED.
5266 030224 026104 000030      15$:    CMP      30(R1),R4
5267 030230 001002      BNE      16$
5268 030232 000161 000036      JMP      36(R1)
5269      ;FPS ERROR WAS NOT ANTICIPATED SO REPORT FAILURE HERE.
5270 030236
5271      16$:
5272 030236 104274      17$:    ERROR   +274           ;BAD FPS
5273 030240 000751      BR       3$
5274
5275 030242
      030242 104412      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?).
    
```

5276
 5285
 5286

```

:*****
:*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.
:*
:*****
TST60: SCOPE
    
```

5287 030244 000004

```

5288 ;NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
5289 MMC1:
030246 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5290 030250 004767 001334 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5291 030254 012345 067012 034567 1$: .WORD 12345,67012,34567,012345 ;ACO OPERAND.
030262 012345
5292 030264 000010 2$: .WORD 10 ;EXPONENT OPERAND.
5293 030266 042145 067012 034567 3$: .WORD 42145,67012,34567,012345 ;EXPECTED RESULT.
030274 012345
5294 030276 002145 067012 034567 4$: .WORD 2145,67012,34567,012345 ;ANTICIPATED ERRONEOUS RESULT.
030304 012345
5295 030306 047217 5$: 47217 ;FPS BEFORE EXECUTION.
5296 030310 047200 47200 ;FPS AFTER EXECUTION.
5297 030312 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
5298 030314 177777 -1 ;EXPECTED FEC.
5299 030316 104304 6$: ERROR +304 ;E12+E12+200 BAD
5300 030320 000400 7$: BR 7$ ;ST 624
5301 030322 104305 7$: ERROR +305 ;REPORT FPS INCORRECT.
;ST 625 INTO 304
5302
5303 ;NON-ZERO RES NEG.
5304 MMC2:
030324 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5305 030326 004737 031610 JSR PC,@#LDXSUB ;EXPON=377
5306 030332 123456 070123 045670 1$: .WORD 123456,70123,45670,123456 ;ACO OPERAND.
030340 123456
5307 030342 000177 2$: .WORD 177 ;EXPONENT OPERAND.
5308 030344 177656 070123 045670 3$: .WORD 177656,70123,45670,123456 ;EXPECTED RESULT.
030352 123456
5309 030354 137656 070123 045670 4$: .WORD 137656,70123,45670,123456 ;ANTICIPATED ERRONEOUS RESULT.
030362 123456
5310 030364 047207 5$: 47207 ;FPS BEFORE EXECUTION.
5311 030366 047210 47210 ;FPS AFTER EXECUTION.
5312 030370 147210 147210 ;ANTICIPATED ERRONEOUS FPS.
5313 030372 177777 -1 ;EXPECTED FEC.
5314 030374 104304 6$: ERROR +304 ;REPORT RESULT INCORRECT.
5315 030376 000401 7$: BR 7$ ;REPORT FPS INCORRECT.
5316 030400 104305 7$: ERROR +305
5317 030402
5318
5319 ;NON-ZERO RES, EXP=256=(56)REAL
5320 MMC3:
030402 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5321 030404 004737 031610 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5322 030410 073261 057645 043323 1$: .WORD 73261,057645,43323,101760 ;ACO OPERAND.
030416 101760
5323 030420 000056 2$: .WORD 56 ;EXPONENT OPERAND.
5324 030422 053461 057645 043323 3$: .WORD 53461,057645,43323,101760 ;EXPECTED RESULT.
030430 101760
5325 030432 177777 177777 177777 4$: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
030440 177777
5326 030442 047200 5$: 47200 ;FPS BEFORE EXECUTION.
5327 030444 047200 47200 ;FPS AFTER EXECUTION.
5328 030446 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
5329 030450 177777 -1 ;EXPECTED FEC.
5330 030452 104301 6$: ERROR +301 ;REPORT RESULT INCORRECT.
5331 030454 000401 7$: BR 7$ ;REPORT FPS INCORRECT.
5332 030456 104305 7$: ERROR +305
    
```

```

5333 030460      7$:
5334
5335      ;EXP=27 (EXCESS 200)=-151 (OCT)
5336 030460      MMC4:
      030460 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5337 030462 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5338 030466 012223 024252 062720 1$:      .WORD      12223,24252,62720,21222 ;ACO OPERAND.
      030474 021222
5339 030476 177627      2$:      .WORD      -151      ;EXPONENT OPERAND.
5340 030500 005623 024252 062720 3$:      .WORD      5623,24252,62720,21222 ;EXPECTED RESULT.
      030506 021222
5341 030510 177777 177777 177777 4$:      .WORD      -1,-1,-1,-1      ;ANTICIPATED ERRONEOUS RESULT.
      030516 177777
5342 030520 047200      5$:      47200      ;FPS BEFORE EXECUTION.
5343 030522 047200      47200      ;FPS AFTER EXECUTION.
5344 030524 147200      147200      ;ANTICIPATED ERRONEOUS FPS.
5345 030526 177777      -1      ;EXPECTED FEC.
5346 030530 104301      6$:      ERROR      +301      ;REPORT RESULT INCORRECT.
5347 030532 000401      BR      7$
5348 030534 104306      ERROR      +306      ;(BUT EZBT) ST 544 TO 504 INTO 704 0 (BUT EXBT) ST 704 INTO
5349 030536      7$:
5350
5351      ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
5352      ; FIV=1
5353 030536      MMC5:
      030536 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5354 030540 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5355 030544 030131 032334 035363 1$:      .WORD      30131,32334,35363,73031 ;ACO OPERAND.
      030552 073031
5356 030554 177600      2$:      .WORD      -200      ;EXPONENT OPERAND.
5357 030556 000131 032334 035363 3$:      .WORD      00131,32334,35363,73031 ;EXPECTED RESULT.
      030564 073031
5358 030566 000000 000000 000000 4$:      .WORD      0,0,0,0      ;ANTICIPATED ERRONEOUS RESULT.
      030574 000000
5359 030576 042200      5$:      42200      ;FPS BEFORE EXECUTION.
5360 030600 142204      142204      ;FPS AFTER EXECUTION.
5361 030602 042202      42202      ;ANTICIPATED ERRONEOUS FPS.
5362 030604 000012      12      ;EXPECTED FEC.
5363 030606 104307      6$:      ERROR      +307      ;(BUT EXBT) ST 704 TO 64 INST 264
5364 030610 000401      BR      7$
5365 030612 104310      ERROR      +310      ;(BUT FIU) ST 264 X
5366 030614      7$:
5367
5368      ;EXP=0 (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
5369 030614      MMC6:
      030614 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5370 030616 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5371 030622 140414 024344 045464 1$:      .WORD      140414,24344,45464,74045 ;ACO OPERAND.
      030630 074045
5372 030632 177600      2$:      .WORD      -200      ;EXPONENT OPERAND.
5373 030634 100014 024344 045464 3$:      .WORD      100014,24344,45464,74045 ;-0 ;EXPECTED RESULT.
      030642 074045
5374 030644 000000 000000 000000 4$:      .WORD      0,0,0,0      ;ANTICIPATED ERRONEOUS RESULT.
      030652 000000
5375 030654 042200      5$:      42200      ;FPS BEFORE EXECUTION.
5376 030656 142214      142214      ;FPS AFTER EXECUTION.
5377 030660 042214      42214      ;ANTICIPATED ERRONEOUS FPS.

```



```

5378 030662 000012          12          ;EXPECTED FEC.
5379 030664 104307          6$:  ERROR  +307          ;REPORT RESULT INCORRECT.
5380 030666 000401          BR          7$
5381 030670 104310          ERROR  +310          ;REPORT FPS INCORRECT.
5382 030672          7$:
5383
5384          ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
5385
5386 030672          MMC7:
      030672 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5387 030674 004737 031610          JSR  PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5388 030700 051525 035455 005675 1$: .WORD 51525,35455,5675,05152 ;ACO OPERAND.
      030706 005152
5389 030710 177600          2$: .WORD -200          ;EXPONENT OPERAND.
5390 030712 000000 000000 000000 3$: .WORD 0,0,0,0          ;EXPECTED RESULT.
      030720 000000
5391 030722 000125 035455 005675 4$: .WORD 00125,35455,5675,05152 ;ANTICIPATED ERRONEOUS RESULT.
      030730 005152
5392 030732 045200          45200          ;FPS BEFORE EXECUTION.
5393 030734 045204          45204          ;FPS AFTER EXECUTION.
5394 030736 145204          145204          ;ANTICIPATED ERRONEOUS FPS.
5395 030740 177777          -1          ;EXPECTED FEC.
5396 030742 104311          6$:  ERROR  +311          ;(BUT FIU) ST 264 X          ;REPORT RESULT INCORRECT.
5397 030744 000401          BR          7$
5398 030746 104302          ERROR  +302          ;REPORT FPS INCORRECT.
5399 030750          7$:
5400
5401          ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
5402 030750          MMC8:
      030750 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5403 030752 004737 031610          JSR  PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5404 030756 061626 062636 046566 1$: .WORD 61626,62636,46566,67606 ;ACO OPERAND.
      030764 067606
5405 030766 176173          2$: .WORD -1605          ;EXPONENT OPERAND.
5406 030770 076626 062636 046566 3$: .WORD 76626,62636,46566,67606 ;EXPECTED RESULT.
      030776 067606
5407 031000 000000 000000 000000 4$: .WORD 0,0,0,0          ;ANTICIPATED ERRONEOUS RESULT.
      031006 000000
5408 031010 042200          5$: 42200          ;FPS BEFORE EXECUTION.
5409 031012 142200          142200          ;FPS AFTER EXECUTION.
5410 031014 042204          42204          ;ANTICIPATED ERRONEOUS FPS.
5411 031016 000012          12          ;EXPECTED FEC.
5412 031020 104312          6$:  ERROR  +312          ;(BUT EZBT) ST 544 TO 704 INTO 504
5413 031022 000401          BR          7$
5414 031024 104302          ERROR  +302          ;REPORT FPS INCORRECT.
5415 031026          7$:
5416          ;EXP=-17416 (EXCESS 200)=-17616 (OCT), FIU=0
5417 031026          MMC9:
      031026 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5418 031030 004737 031610          JSR  PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5419 031034 071727 037475 076777 1$: .WORD 71727,37475,76777,17273 ;ACO OPERAND.
      031042 017273
5420 031044 160162          2$: .WORD -17616          ;EXPONENT OPERAND.
5421 031046 000000 000000 000000 3$: .WORD 0,0,0,0          ;EXPECTED RESULT.
      031054 000000
5422 031056 074527 037475 076777 4$: .WORD 74527,37475,76777,17273 ;ANTICIPATED ERRONEOUS RESULT.
      031064 017273
    
```

```

5423 031066 045200      5$:      45200      ;FPS BEFORE EXECUTION.
5424 031070 045204      ;FPS AFTER EXECUTION.
5425 031072 145200      ;ANTICIPATED ERRONEOUS FPS.
5426 031074 177777      -1          ;EXPECTED FEC.
5427 031076 104313      6$:      ERROR      +313      ;(BUT FIU) ST 504
5428 031100 000401      BR          7$
5429 031102 104302      ERROR      +302      ;REPORT FPS INCORRECT.
5430 031104
5431
5432      ;EXP=-1601 (EXCESS 200)=-2001 (OCT), FIU=1
5433 031104      MMC10:
      031104 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5434 031106 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5435 031112 001020 030405 006070 1$:      .WORD      01020,30405,06070,00102 ;ACO OPERAND.
      031120 000102
5436 031122 175777      2$:      .WORD      -2001      ;EXPONENT OPERAND.
5437 031124 037620 030405 006070 3$:      .WORD      37620,30405,06070,00102 ;EXPECTED RESULT.
      031132 000102
5438 031134 000000 000000 000000 4$:      .WORD      0,0,0,0      ;ANTICIPATED ERRONEOUS RESULT.
      031142 000000
5439 031144 042200      5$:      42200      ;FPS BEFORE EXECUTION.
5440 031146 142200      ;FPS AFTER EXECUTION.
5441 031150 042204      ;ANTICIPATED ERRONEOUS FPS.
5442 031152 000012      12          ;EXPECTED FEC.
5443 031154 104312      6$:      ERROR      +312      ;(BUT FIU) ST 504
5444 031156 000401      BR          7$
5445 031160 104302      ERROR      +302      ;REPORT FPS INCORRECT.
5446 031162
5447
5448      ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
5449 031162      MMC11:
      031162 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5450 031164 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5451 031170 012131 014151 016171 1$:      .WORD      12131,14151,16171,10111 ;ACO OPERAND.
      031176 010111
5452 031200 001006      2$:      .WORD      1006      ;EXPONENT OPERAND.
5453 031202 041531 014151 016171 3$:      .WORD      41531,14151,16171,10111 ;EXPECTED RESULT.
      031210 010111
5454 031212 000000 000000 000000 4$:      .WORD      0,0,0,0      ;ANTICIPATED ERRONEOUS RESULT.
      031220 000000
5455 031222 041200      5$:      41200      ;FPS BEFORE EXECUTION.
5456 031224 141202      ;FPS AFTER EXECUTION.
5457 031226 041204      ;ANTICIPATED ERRONEOUS FPS.
5458 031230 000010      10          ;EXPECTED FEC.
5459 031232 104314      6$:      ERROR      +314      ;(BUT FIV) ST 104
5460 031234 000401      BR          7$
5461 031236 104302      ERROR      +302      ;REPORT FPS INCORRECT.
5462 031240
5463
5464      ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
5465 031240      MMC12:
      031240 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5466 031242 004737 031610      JSR      PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5467 031246 027262 025242 023222 1$:      .WORD      27262,25242,23222,21202 ;ACO OPERAND.
      031254 021202
5468 031256 016115      2$:      .WORD      16115      ;EXPONENT OPERAND.
5469 031260 000000 000000 000000 3$:      .WORD      0,0,0,0      ;EXPECTED RESULT.
    
```

5470 031266 000000
031270 063262 025242 023222 4\$: .WORD 63262,25242,23222,21202 ;ANTICIPATED ERRONEOUS RESULT.
031276 021202
5471 031300 046200 5\$: 46200 ;FPS BEFORE EXECUTION.
5472 031302 046206 46206 ;FPS AFTER EXECUTION.
5473 031304 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
5474 031306 177777 -1 ;EXPECTED FEC.
5475 031310 104315 6\$: ERROR +315 ;(BUT FIV) ST 104
5476 031312 000401 BR 7\$
5477 031314 104302 ERROR +302 ;REPORT FPS INCORRECT.
5478 031316 7\$:
5479
5480 ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1
5481

5482 031316 104413 MMC13:
031316 004737 031610 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5483 031320 030313 032333 034353 1\$: JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5484 031324 036373 .WORD 30313,32333,34353,36373 ;ACO OPERAND.
031332 036373
5485 031334 010611 2\$: .WORD 10611 ;EXPONENT OPERAND.
5486 031336 002313 032333 034353 3\$: .WORD 2313,32333,34353,36373 ;EXPECTED RESULT.
031344 036373
5487 031346 000000 000000 000000 4\$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
031354 000000
5488 031356 041200 5\$: 41200 ;FPS BEFORE EXECUTION.
5489 031360 141202 141202 ;FPS AFTER EXECUTION.
5490 031362 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
5491 031364 000010 10 ;EXPECTED FEC.
5492 031366 104316 6\$: ERROR +316 ;(BUT FIV) ST 144
5493 031370 000401 BR 7\$
5494 031372 104302 ERROR +302 ;REPORT FPS INCORRECT.
5495 031374 7\$:
5496
5497 ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
5498

5499 031374 104413 MMC14:
031374 004737 031610 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5500 031376 040414 042434 044454 1\$: JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5501 031402 046474 .WORD 40414,42434,44454,46474 ;ACO OPERAND.
031410 046474
5502 031412 016723 2\$: .WORD 16723 ;EXPONENT OPERAND.
5503 031414 000000 000000 000000 3\$: .WORD 0,0,0,0 ;EXPECTED RESULT.
031422 000000
5504 031424 024614 042434 044454 4\$: .WORD 24614,42434,44454,46474 ;ANTICIPATED ERRONEOUS RESULT.
031432 046474
5505 031434 046200 5\$: 46200 ;FPS BEFORE EXECUTION.
5506 031436 046206 46206 ;FPS AFTER EXECUTION.
5507 031440 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
5508 031442 177777 -1 ;EXPECTED FEC.
5509 031444 104317 6\$: ERROR +317 ;(BUT FIV) ST 144
5510 031446 000401 BR 7\$
5511 031450 104302 ERROR +302 ;REPORT FPS INCORRECT.
5512 031452 7\$:
5513
5514 ;EXP= 254 (OCT)= 454 (EXCESS 200) FIV=1
5515

5516 031452 MMC15:

```

5517 031452 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5518 031454 004737 031610  JSR          PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5518 031460 050515 052535 054555 1$: .WORD      50515,52535,54555,56575 ;ACO OPERAND.
5519 031466 056575          ;
5519 031470 000254          2$: .WORD      254 ;EXPONENT OPERAND.
5520 031472 013115 052535 054555 3$: .WORD      13115,52535,54555,56575 ;EXPECTED RESULT.
5521 031500 056575          ;
5521 031502 000000 000000 000000 4$: .WORD      0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
5522 031510 000000          ;
5522 031512 041200          5$: 41200 ;FPS BEFORE EXECUTION.
5523 031514 141202          141202 ;FPS AFTER EXECUTION.
5524 031516 041204          41204 ;ANTICIPATED ERRONEOUS FPS.
5525 031520 000010          10 ;EXPECTED FEC.
5526 031522 104320          6$: ERROR      +320 ;(BUT FIV) ST344
5527 031524 000401          BR          7$
5528 031526 104302          ERROR      +302 ;REPORT FPS INCORRECT.
5529 031530          7$:
5530
5531 ;EXP= 313 (OCT)= 513(EXCESS 200) FIV=0
5532
5533 MMC16:
5534 031530 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5535 031532 004737 031610  JSR          PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
5535 031536 060616 062636 064656 1$: .WORD      60616,62636,64656,66676 ;ACO OPERAND.
5536 031544 066676          ;
5536 031546 000313          2$: .WORD      313 ;EXPONENT OPERAND.
5537 031550 000000 000000 000000 3$: .WORD      0,0,0,0 ;EXPECTED RESULT.
5538 031556 000000          ;
5538 031560 022616 062636 064656 4$: .WORD      22616,62636,64656,66676 ;ANTICIPATED ERRONEOUS RESULT.
5539 031566 066676          ;
5539 031570 046200          5$: 46200 ;FPS BEFORE EXECUTION.
5540 031572 046206          46206 ;FPS AFTER EXECUTION.
5541 031574 146202          146202 ;ANTICIPATED ERRONEOUS FPS.
5542 031576 177777          -1 ;EXPECTED FEC.
5543 031600 104321          6$: ERROR      +321 ;(BUT FIV) ST 344
5544 031602 000401          BR          7$
5545 031604 104302          ERROR      +302 ;REPORT FPS INCORRECT.
5546 031606          7$:
5547 031606 000540          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:

```

5552 :
5553 :
5554 : JSR          PC,@#LDXSUB
5554 : ACARG: .WORD      X,X,X,X ;AC OPERAND
5555 : EXP: .WORD      X ;EXPONENT
5556 : RES: .WORD      X,X,X,X ;EXPECTED RESULT
5557 : ERRES: .WORD     X,X,X,X ;ERROR RESULT
5558 : FPSB: .WORD      X ;FPS BEFORE EXECUTION
5559 : FPSA: .WORD      X ;FPS AFTER EXECUTION
5560 : ERFPS: .WORD     X ;ERROR FPS.
5561 : FEC: .WORD      X ;EXPECTED FEC
5562 : ERR1: ERROR     +X ;DATA ERROR.
5563 : BR          CONT
5564 : ERR2: ERROR     +X ;FPS ERROR.
5565 : CONT: ;RETURN ADDRESS

```

```

5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582 031610 012601
5583 031612 012700 000200
5584 031616 170100
5585 031620 010100
5586 031622 172410
5587 031624 012737 031646 001236
5588 031632 016100 000032
5589 031636 170100
5590 031640 010100
5591 031642 062700 000010
5592
5593 031646 176410
5594
5595 031650 170204
5596 031652 170305
5597 031654 012700 000200
5598 031660 170100
5599 031662 012700 032100
5600 031666 174010
5601 031670 010437 001250
5602 031674 016137 000034 001252
5603 031702 010537 001254
5604 031706 016137 000040 001256
5605 031714 010102
5606 031716 010237 001240
5607 031722 062702 000010
5608 031726 011237 001242
5609 031732 062702 000002
5610 031736 010237 001244
5611 031742 012737 032100 001246
5612 031750 012702 032100
5613 031754 010103
5614 031756 062703 000012
5615 031762 012700 000004
5616 031766 022223
5617 031770 001014
5618 031772 077003
5619 031774 020461 000034
5620 032000 001026
5621 032002 005761 000034
5622 032006 100003

```

; 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.

```

LDXSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
MOV #200,R0 ;LOAD THE ACO OPERAND.
LDFPS R0
MOV R1,R0
LDD (R0),ACO
MOV #1$,@#$TMP2 ;SET UP THE FPS.
MOV 32(R1),R0
LDFPS R0
MOV R1,R0
ADD #10,R0

1$: LDEXP (R0),ACO ;TEST INSTRUCTION.

STFPS R4 ;GET THE FPS.
STST R5 ;GET THE FEC.
MOV #200,R0 ;GET THE RESULT.
LDFPS R0
MOV #LDXT,R0
STD ACO,(R0)
MOV R4,@#$TMP7
MOV 34(R1),@#$TMP10
MOV R5,@#$TMP11
MOV 40(R1),@#$TMP12
MOV R1,R2
MOV R2,@#$TMP3
ADD #10,R2
MOV (R2),@#$TMP4
ADD #2,R2
MOV R2,@#$TMP5
MOV #LDXT,@#$TMP6 ;SEE IF THE RESULT WAS CORRECT.
MOV #LDXT,R2
MOV R1,R3
ADD #12,R3
MOV #4,R0
2$: CMP (R2)+,(R3)+ ;BRANCH IF NOT CORRECT.
BNE 10$
SOB R0,2$
CMP R4,34(R1) ;SEE IF THE FPS WAS CORRECT.
BNE 15$ ;BRANCH IF NOT CORRECT.
TST 34(R1)
BPL 3$

```

```
5623 032010 020561 000040      CMP    R5,40(R1)      ;SEE IF THE FEC WAS CORRECT.
5624 032014 001027              BNE    20$            ;BRANCH IF NOT CORRECT.
5625
5626 032016 000161 000050      3$:    JMP    50(R1)      ;RETURN.
5627
5628      ;THE RESULT WAS INCORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
5629 032022 012702 032100      10$:   MOV    #LDXT,R2
5630 032026 010103              MOV    R1,R3
5631 032030 062703 000022              ADD    #22,R3
5632 032034 012700 000004              MOV    #4,R0
5633 032040 022223      11$:   CMP    (R2)+,(R3)+
5634 032042 001003              BNE    12$
5635 032044 077003              SOB    R0,11$
5636 032046 000161 000042              JMP    42(R1)
5637
5638      ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
5639 032052      12$:
5640 032052 104301      13$:   ERROR  +301          ;BAD RES
5641 032054 000760              BR     3$
5642
5643      ;SEE IF THE FPS ERROR WAS ANTICIPATED.
5644 032056 026104 000036      15$:   CMP    36(R1),R4
5645 032062 001002              BNE    16$
5646 032064 000161 000046              JMP    46(R1)
5647 032070
5648      ;THE FPS WAS NOT ANTICIPATED SO REPORT IT HERE.
5649 032070 104302      17$:   ERROR  +302          ;BAD FPS
5650 032072 000751              BR     3$              ;BUT EZBTY8
5651                          ;ST 063
5652
5653 032074      20$:
5654      ;REPORT FEC INCORRECT.
5655 032074 104303      21$:   ERROR  +303          ;BAD FEC
5656 032076 000747              BR     3$
5657
5658      ;DATA BUFFER:
5659 032100 000000 000000 000000 LDXT:  .WORD  0,0,0,0
5660 032106 000000
5661 032110
5661 032110 104412      MMCDONE:
                    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?).
```

5662
5663
5670
5671

```
*****
*TEST 61      DESTINATION MODES, MODE 1 (FL=0), TEST
*
* THIS IS A TEST OF DESTINATION MODE 1 USING
* THE STFPS INSTRUCTION
*
*****
TST61:  SCOPE
```

5672 032112 000004

```

5673
5674 032114          NNC1:
      032114 104413    LPERR                ;SET UP THE LOOP ON ERROR ADDRESS.
5675 032116 012700 032214    MOV #NNCTB0,R0      ;SET UP THE DATA BUFFER.
5676 032122 012701 000006    MOV #6,R1
5677 032126 012720 177777    1$: MOV #-1,(R0)+
5678 032132 077103    SOB R1,1$
5679 032134 012700 102345    MOV #102345,R0
5680 032140 012737 032162 001236    MCV #NNC2,@#$TMP2
5681 032146 012737 032314 000004    MOV #NNC25,@#ERRVECT ;SET UP FOR TRAPS TO 4.
5682 032154 170100    LDFPS R0          ;SET UP FPS.
5683 032156 012700 032220    MOV #NNCTB1,R0
5684
5685 032162 170210          NNC2: STFPS (R0)      ;TEST INSTRUCTION.
5686 032164 020027 032220    CMP R0,#NNCTB1    ;IS R0 CORRECT?
5687 032170 001017          BNE NNC10         ;BRANCH IF NOT CORRECT.
5688 032172 023727 032220 102345    CMP @#NNCTB1,#102345 ;IS RESULT CORRECT?
5689 032200 001023          BNE NNC15         ;BRANCH IF NOT CORRECT.
5690 032202 023727 032222 177777    CMP @#NNCTB1+2,#-1 ;IS THE RESULT CORRECT?
5691 032210 001030          BNE NNC20         ;BRANCH IF NOT CORRECT.
5692 032212 000453          BR NNCDONE
5693
5694          ;TEST DATA BUFFER:
    
```

```

5696 032214 177777 177777 NNCTB0: .WORD -1,-1
5697 032220 177777 177777 177777 NNCTB1: .WORD -1,-1,-1,-1
032226 177777

5698
5699
5700 032230 010037 001242 ;REPORT RO INCORRECT.
5701 032234 012737 032220 001240 NNC10: MOV RO,@#$TMP4
5702 032242 104377 MOV #NNCTB1,@#$TMP3
032242 000001 1$: ERROR +377
032244 .WORD 1

5703 ;RO BAD (BUT
5704 032246 000435 BR NNCDONE ; FDST)X
5705
5706 ;REPORT RESULT INCORRECT.
5707 032250 012737 102345 001240 NNC15: MOV #102345,@#$TMP3 ; ST 634
5708 032256 013737 032220 001242 MOV @#NNCTB1,@#$TMP4
5709 032264 104377 1$: ERROR +377
032264 000002 .WORD 2 ;BAD DATA
032266

5710
5711 032270 000424 BR NNCDONE
5712
5713
5714 ;REPORT RESULT INCORRECT.
5715 032272 012737 177777 001240 NNC20: MOV #-1,@#$TMP3
5716 032300 013737 032222 001242 MOV @#NNCTB1+2,@#$TMP4
5717 032306 104377 1$: ERROR +377
032306 000003 .WORD 3
032310

5718 ;(BUT GR7,FL)
5719 032312 000413 BR NNCDONE ;ST 357 TO 416
5720 ;INTO 417
5721
5722 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5723 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
5724 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5725 032314 011604 NNC25: MOV (SP),R4
5726 032316 020427 032164 CMP R4,#NNC2+2
5727 032322 001402 BEQ 1$
5728 032324 000137 046310 JMP @#CPSPUR
5729
5730 032330 011637 001236 1$: MOV (SP),@#$TMP2
5731 032334 022626 CMP (SP)+,(SP)+
5732 032336 104377 2$: ERROR +377
032340 000004 .WORD 4
032342

5733 ;(BUT FDST)+ ST634
5734
5735 032342 NNCDONE:
032342 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?).

5736
5737
5738 ;:*****

```


;*TEST 62 DESTINATION MODES, MODE 2 (FL=0), TEST

;* THIS IS A TEST OF DESTINATION MODE 2 USING
 ;* THE STFPS INSTRUCTION

 TST62: SCOPE

5739	032344	000004			
5740					
5741	032346				OOC1:
	032346	104413			LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5742	032350	012700	032446		MOV #00CTB0,R0 ;SET UP THE DATA BUFFER.
5743	032354	012701	000006		MOV #6,R1
5744	032360	012720	177777		1\$: MOV #-1,(R0)+
5745	032364	077103			SOB R1,1\$
5746	032366	012700	105412		MOV #105412,R0
5747	032372	012737	032414	001236	MOV #OOC2,@#\$TMP2
5748	032400	012737	032546	000004	MOV #OOC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
5749	032406	170100			LDFPS R0 ;SET UP FPS.
5750	032410	012700	032452		MOV #00CTB1,R0
5751					
5752	032414	170220			OOC2: STFPS (R0)+ ;TEST INSTRUCTION.
5753	032416	020027	032454		CMP R0,#00CTB1+2 ;IS R0 CORRECT?
5754	032422	001017			BNE OOC10 ;BRANCH IF NOT CORRECT.
5755	032424	023727	032452	105412	CMP @#00CTB1,#105412 ;IS THE RESULT CORRECT?
5756	032432	001023			BNE OOC15 ;BRANCH IF NOT CORRECT.
5757	032434	023727	032454	177777	CMP @#00CTB1+2,#-1 ;IS THE RESULT CORRECT?
5758	032442	001030			BNE OOC20 ;BRANCH IF NOT CORRECT.
5759	032444	000453			BR OOCDONE
5760					
5761					;TEST DATA BUFFER:
5762	032446	177777	177777		OOCB0: .WORD -1,-1
5763	032452	177777	177777	177777	OOCB1: .WORD -1,-1,-1,-1
	032460	177777			
5764					
5765					;REPORT R0 INCORRECT.
5766	032462	010037	001242		OOC10: MOV R0,@#\$TMP4
5767	032466	012737	032454	001240	MOV #00CTB1+2,@#\$TMP3
5768	032474				1\$: ERROR +377
	032474	104377			.WORD 5
	032476	000005			
5769					
5770	032500	000435			BR OOCDONE ;R0 BAD (BUT
5771					; FDST)X
5772					
5773	032502	012737	105412	001240	;REPORT RESULT INCORRECT.
5774	032510	013737	032452	001242	OOC15: MOV #105412,@#\$TMP3 ; ST 634
5775	032516				MOV @#00CTB1,@#\$TMP4
	032516	104377			1\$: ERROR +377
	032520	000006			.WORD 6
5776					
5777	032522	000424			BR OOCDONE ;BAD DATA
5778					
5779					
5780					;REPORT RESULT INCORRECT.
5781	032524	012737	177777	001240	OOC20: MOV #-1,@#\$TMP3
5782	032532	013737	032454	001242	MOV @#00CTB1+2,@#\$TMP4

5783 032540
 032540 104377
 032542 000007
 5784
 5785 032544 000413
 5786
 5787
 5788
 5789
 5790
 5791 032546 011604
 5792 032550 020427 032416
 5793 032554 001402
 5794 032556 000137 046310
 5795
 5796 032562 011637 001236
 5797 032566 022626
 5798 032570
 032570 104377
 032572 000010
 5799
 5800
 5801 032574
 032574 104412

1\$: ERROR +377
 .WORD 7
 BR OOCDONE ;(BUT GR7,FL)
 ;ST 357 TO 416
 ;INTO 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,#OOC2+2
 BEQ 1\$
 JMP @#CPSPUR

1\$: MOV (SP),@#\$TMP2
 CMP (SP)+,(SP)+

2\$: 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?).

5802
 5803
 5804
 5805

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

032576 000004
 5806
 5807 032600
 032600 104413
 5808 032602 012700 032700
 5809 032606 012701 000006
 5810 032612 012720 177777
 5811 032616 077103
 5812 032620 012700 105555
 5813 032624 012737 032646 001236
 5814 032632 012737 033000 000004
 5815 032640 170100
 5816 032642 012700 032706
 5817
 5818 032646 170240
 5819 032650 020027 032704
 5820 032654 001017
 5821 032656 023727 032704 105555
 5822 032664 001023

TST63: SCOPE

PPC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
 MOV #PPCTB0,R0 ;SET UP THE DATA BUFFER.
 MOV #6,R1

1\$: MOV #-1,(R0)+
 SOB R1,1\$
 MOV #105555,R0
 MOV #PPC2,@#\$TMP2
 MOV #PPC25,@#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.
 CMP @#PPCTB1,#105555 ;IS THE RESULT CORRECT?
 BNE PPC15 ;BRANCH IF NOT CORRECT.

```

5823 032666 023727 032706 177777      CMP      @#PPCTB1+2,#-1  ;IS THE RESULT CORRECT?
5824 032674 001030                      BNE      PPC20          ;BRANCH IF NOT CORRECT.
5825 032676 000453                      BR       PPCDONE
5826
5827                                     ;TEST DATA BUFFER:
5828 032700 177777 177777                PPCTB0: .WORD  -1,-1
5829 032704 177777 177777 177777        PPCTB1: .WORD  -1,-1,-1,-1
5830
5831                                     ;REPORT RO INCORRECT.
5832 032714 010037 001242                PPC10:  MOV      RO,@#$TMP4
5833 032720 012737 032704 001240        MOV      #PPCTB1,@#$TMP3
5834 032726                                     1$:
5835 032726 104377                          ERROR    +377
5836 032732 000435                          .WORD   11
5837                                     ;RO BAD (BUT
5838                                     ;      FDST)X
5839 032734 012737 105555 001240        ;REPORT RESULT INCORRECT.
5840 032742 013737 032704 001242        PPC15:  MOV      #105555,@#$TMP3
5841 032750                                     MOV      @#PPCTB1,@#$TMP4
5842 032750 104377                          1$:
5843 032752 000012                          ERROR    +377
5844                                     .WORD   12
5845                                     ;BAD DATA
5846                                     BR       PPCDONE
5847 032756 012737 177777 001240        ;REPORT RESULT INCORRECT.
5848 032764 013737 032706 001242        PPC20:  MOV      #-1,@#$TMP3
5849 032772                                     MOV      @#PPCTB1+2,@#$TMP4
5850 032772 104377                          1$:
5851 032774 000013                          ERROR    +377
5852                                     .WORD   13
5853                                     ;(BUT GR7,FL)
5854                                     ;ST 357 TO 416
5855                                     ;INTO 417
5856                                     ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5857                                     ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
5858                                     ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5859 033000 011604                      PPC25:  MOV      (SP),R4
5860 033002 020427 032650                CMP      R4,#PPC2+2
5861 033006 001402                      BEQ     1$
5862 033010 000137 046310                JMP     @#CPSPUR
5863 033014 011637 001236                1$:    MOV      (SP),@#$TMP2
5864 033020 022626                      CMP     (SP)+,(SP)+
5865 033022                                     2$:
5866 033022 104377                          ERROR    +377
5867 033024 000014                          .WORD   14
5868                                     ;(BUT FDST)+ ST634
5869 033026 104412                      PPCDONE:
5870 033026                                     RSETUP
5871                                     ;GO INITIALIZE THE FPS AND STACK; AND
5872                                     ;SEE IF THE USER HAS EXPRESSED
5873                                     ;THE DESIRE TO CHANGE THE SOFTWARE
    
```

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

5868
 5869
 5870
 5871

```

:*****
:*TEST 64      DESTINATION MODES, MODE 3 (FL=0), TEST
:*
:* THIS IS A TEST OF DESTINATION MODE 3 USING
:* THE STFPS INSTRUCTION
:*
:*****
TST64: SCOPE
  
```

```

5872 033030 000004
5873 033032
5874 033032 104413
5875 033034 012700 033136
5876 033040 012701 000010
5877 033044 012720 177777
5878 033050 077103
5879 033052 012700 106653
5880 033056 012737 033104 001236
5881 033064 012737 033242 000004
5882 033072 170100
5883 033074 012700 033152
5884 033100 012710 033142
5885 033104 170230
5886 033106 020027 033154
5887 033112 001021
5888 033114 023727 033142 106653
5889 033122 001025
5890 033124 023727 033152 033142
5891 033132 001032
5892 033134 000455
5893
5894
5895 033136 177777 177777
5896 033142 177777 177777 177777
5897 033150 177777
5898 033152 177777 177777
5899
5900 033156 010037 001242
5901 033162 012737 033154 001240
5902 033170
5903 033170 104377
5904 033172 000015
5905
5906 033174 000435
5907
5908 033176 012737 106653 001240
5909 033204 013737 033142 001242
5910 033212
5911 033212 104377
5912 033214 000016
  
```

```

QQC1:
  LPERR                                ;SET UP THE LOOP ON ERROR ADDRESS.
  MOV #QQCTB0,R0                       ;SET UP THE DATA BUFFER.
  MOV #10,R1
  1$: MOV #-1,(R0)+
      SOB R1,1$
      MOV #106653,R0
      MOV #QQC2,@#$TMP2
      MOV #QQC25,@#ERRVECT             ;SET UP FOR TRAPS TO VECTOR 4.
      LDFPS R0                          ;SET UP FPS.
      MOV #QQCTB2,R0
      MOV #QQCTB1,(R0)

QQC2: STFPS @ (R0)+                    ;TEST INSTRUCTION.
      CMP R0,#QQCTB2+2                 ;IS R0 CORRECT?
      BNE QQC10                         ;BRANCH IF NOT CORRECT.
      CMP @#QQCTB1,#106653             ;IS THE RESULT CORRECT?
      BNE QQC15                         ;BRANCH IF NOT CORRECT.
      CMP @#QQCTB2,#QQCTB1            ;IS THE RESULT CORRECT?
      BNE QQC20                         ;BRANCH IF NOT CORRECT.
      BR QQCDONE

;TEST DATA BUFFER:
QQCTB0: .WORD -1,-1
QQCTB1: .WORD -1,-1,-1,-1
QQCTB2: .WORD -1,-1

;REPORT R0 INCORRECT.
QQC10: MOV R0,@#$TMP4
        MOV #QQCTB2+2,@#$TMP3
  1$:   ERROR +377
        .WORD 15
;R0 BAD (BUT
; FDST)X

;REPORT RESULT INCORRECT.
QQC15: MOV #106653,@#$TMP3
        MOV @#QQCTB1,@#$TMP4
  1$:   ERROR +377
        .WORD 16
; ST 634
  
```

```

5910                                     ;BAD DATA
5911 033216 000424                     BR      QQCDONE
5912
5913
5914                                     ;REPORT RESULT INCORRECT.
5915 033220 012737 033152 001240 QQC20: MOV    #QQCTB2,@#$TMP3           ;(BUT FDST)
5916 033226 013737 033144 001242     MOV    @#QQCTB1+2,@#$TMP4
5917 033234                               1$:
      033234 104377                       ERROR  +377
      033236 000017                       .WORD  17
5918 033240 000413                       BR      QQCDONE
5919
5920
5921                                     ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5922                                     ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
5923                                     ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5924 033242 011604                               QQC25: MOV    (SP),R4
5925 033244 020427 033106                       CMP     R4,#QQC2+2
5926 033250 001402                               BEQ     1$
5927 033252 000137 046310                       JMP     @#CPSPUR
5928
5929 033256 011637 001236                       1$:    MOV    (SP),@#$TMP2
5930 033262 022626                               CMP     (SP)+,(SP)+
5931 033264                               2$:
      033264 104377                       ERROR  +377
      033266 000020                       .WORD  20
5932                                     ;(BUT FDST)+ ST634
5933
5934 033270                               QQCDONE:
      033270 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?).

5935
5936
5937
5938
                                     ;*****
                                     ;*TEST 65      DESTINATION MODES, MODE 5 (FL=0), TEST
                                     ;*
                                     ;* THIS IS A TEST OF DESTINATION MODE 5 USING
                                     ;* THE STFPS INSTRUCTION
                                     ;*
                                     ;*****
5939 033272 000004                               TST65: SCOPE
5940
5941 033274                               RRC1:
      033274 104413                               LPERR
5942 033276 012700 033402                       MOV    #RRC2B0,R0           ;SET UP THE LOOP ON ERROR ADDRESS.
5943 033302 012701 000006                       MOV    #6,R1               ;SET UP THE DATA BUFFER.
5944 033306 012720 177777                       1$:    MOV    #-1,(R0)+
5945 033312 077103                               SOB    R1,1$
5946 033314 012700 004301                       MOV    #004301,R0
5947 033320 012737 033350 001236                       MOV    #RRC2,@#$TMP2
5948 033326 012737 033506 000004                       MOV    #RRC25,@#ERRVECT   ;SET UP FOR TRAPS TO VECTOR 4.
5949 033334 170100                               LDFPS  R0                   ;SET UP FPS.
  
```

```

5950 033336 012700 033420      MOV      #RRCTB2+2,R0
5951 033342 012760 033406 177776      MOV      #RRCTB1,-2(R0)
5952
5953 033350 170250      RRC2:   STFPS  @-(R0)      ;TEST INSTRUCTION.
5954 033352 020027 033416      CMP      R0,#RRCTB2      ;IS R0 CORRECT?
5955 033356 001021      BNE     RRC10             ;BRANCH IF NOT CORRECT.
5956 033360 023727 033406 004301      CMP      @#RRCTB1,#004301 ;IS THE RESULT CORRECT?
5957 033366 001025      BNE     RRC15             ;BRANCH IF NOT CORRECT.
5958 033370 023727 033416 033406      CMP      @#RRCTB2,#RRCTB1 ;IS THE RESULT CORRECT?
5959 033376 001032      BNE     RRC20             ;BRANCH IF NOT CORRECT.
5960 033400 000455      BR      RRCDONE
5961
5962      ;TEST DATA BUFFER:
5963 033402 177777 177777      RRC20:  .WORD  -1,-1
5964 033406 177777 177777 177777      RRC21:  .WORD  -1,-1,-1,-1
5965 033414 177777
5966 033416 177777 177777      RRC22:  .WORD  -1,-1
5967
5968 033422 010037 001242      ;REPORT R0 INCORRECT.
5969 033426 012737 033416 001240      RRC10:  MOV      R0,@#$TMP4
5970 033434      1$:    MOV      #RRCTB2,@#$TMP3
5971 033434 104377      ERROR  +377
5972 033436 000021      .WORD  21
5973
5974      ;RO BAD (BUT
5975 033440 000435      BR      RRCDONE          ; FDST)X
5976
5977      ;REPORT RESULT INCORRECT.
5978 033442 012737 004301 001240      RRC15:  MOV      #004301,@#$TMP3      ; ST 634
5979 033450 013737 033406 001242      MOV      @#RRCTB1,@#$TMP4
5980 033456      1$:    ERROR  +377
5981 033460 000022      .WORD  22
5982
5983      ;BAD DATA
5984 033462 000424      BR      RRCDONE
5985
5986      ;REPORT RESULT INCORRECT.
5987 033464 012737 033416 001240      RRC20:  MOV      #RRCTB2,@#$TMP3      ;BUT FDST)
5988 033472 013737 033410 001242      MOV      @#RRCTB1+2,@#$TMP4
5989 033500      1$:    ERROR  +377
5990 033500 104377      .WORD  23
5991 033502 000023
5992
5993      ;(BUT GR7,FL)
5994 033504 000413      BR      RRCDONE          ;ST 357 TO 416
5995
5996      ;INTO 417
5997
5998      ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5999      ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
6000      ;TO THE SPURIOUS TRAP TO 4 HANDLER.
6001 033506 011604      RRC25:  MOV      (SP),R4
6002 033510 020427 033352      CMP      R4,#RRC2+2
6003 033514 001402      BEQ     1$
6004 033516 000137 046310      JMP     @#CPSPUR
6005
6006 033522 011637 001236      1$:    MOV      (SP),@#$TMP2
6007 033526 022626      CMP     (SP)+,(SP)+
    
```

6000 033530
 033530 104377
 033532 000024

2\$:
 ERROR +377
 .WORD 24

;(BUT FDST)+ ST634

6001
 6002
 6003 033534
 033534 104412

RRC DONE:
 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?).

6004
 6005
 6006

```

:*****
:*TEST 66      DESTINATION MODES, MODE 6 (FL=0), TEST
:*
:* THIS IS A TEST OF DESTINATION MODE 6 USING
:* THE STFPS INSTRUCTION
:*
:*****
TST66: SCOPE
    
```

6007 033536 000004
 6008
 6009 033540

SSC1:

```

LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
MOV        #SSCTB0,R0 ;SET UP THE DATA BUFFER.
MOV        #6,R1
1$: MOV     #-1,(R0)+
SOB        R1,1$
MOV        #102514,R0
MOV        #SSC2,@#$TMP2
MOV        #SSC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS     R0 ;SET UP FPS.
CLR        R1
MOV        #SSCTB1-5201,R0
    
```

6010 033542 012700 033652
 6011 033546 012701 000006
 6012 033552 012720 177777
 6013 033556 077103
 6014 033560 012700 102514
 6015 033564 012737 033610 001236
 6016 033572 012737 033752 000004
 6017 033600 170100
 6018 033602 005001
 6019 033604 012700 026455
 6020
 6021 033610 170260 005201
 6022 033614 020127 000000
 6023 033620 001070
 6024 033622 020027 026455
 6025 033626 001017
 6026 033630 023727 033656 102514
 6027 033636 001023
 6028 033640 023727 033660 177777
 6029 033646 001030
 6030 033650 000456

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
    
```

6031
 6032
 6033 033652 177777 177777
 6034 033656 177777 177777 177777
 033664 177777

```

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

6035
 6036
 6037 033666 010037 001242
 6038 033672 012737 026455 001240
 6039 033700
 033700 104377

```

;REPORT R0 INCORRECT.
SSC10: MOV    R0,@#$TMP4
MOV     #SSCTB1-5201,@#$TMP3
1$:     ERROR +377
    
```

CKFPCBO FP11F FLTG PNT PRT C MACRO M1113 11-DEC-79 17:03 PAGE H 12
T66 DESTINATION MODES, MODE 6 (FL=0), TEST 26-8

SEQ 0150

033702 000025

.WORD 25


```

6041                                     ;R0 BAD
6042 033704 000440                     BR      SSCDONE
6043
6044                                     ;REPORT RESULT INCORRECT.
6045 033706 012737 102534 001240 SSC15: MOV      #102534,@#$TMP3
6046 033714 013737 033656 001242      MOV      @#$SCTB1,@#$TMP4
6047 033722                               1$:
        033722 104377                     ERROR   +377
        033724 000026                     .WORD   26
6048                                     ;BAD DATA
6049 033726 000427                     BR      SSCDONE
6050
6051
6052                                     ;REPORT RESULT INCORRECT.
6053 033730 012737 177777 001240 SSC20: MOV      #-1,@#$TMP3
6054 033736 013737 033660 001242      MOV      @#$SCTB1+2,@#$TMP4
6055 033744                               1$:
        033744 104377                     ERROR   +377
        033746 000027                     .WORD   27
6056                                     ;(BUT GR7,FL)
6057 033750 000416                     BR      SSCDONE                       ;ST 357 TO 416
6058                                     ;INTO 417
6059
6060                                     ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
6061                                     ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
6062                                     ;TO THE SPURIOUS TRAP TO 4 HANDLER.
6063 033752 011604                               SSC25: MOV      (SP),R4
6064 033754 020427 033612                     CMP      R4,#$SSC2+2
6065 033760 001402                               BEQ      1$
6066 033762 000137 046310                     JMP      @#$CPSPUR
6067
6068 033766 011637 001236                     1$: MOV      (SP),@#$TMP2
6069 033772 022626                               CMP      (SP)+,(SP)+
6070 033774                               2$:
        033774 104377                     ERROR   +377
        033776 000030                     .WORD   30
6071                                     ;(BUT FDST)+ ST634
6072 034000 000402                     BR      SSCDONE
6073
6074                                     ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
6075 034002                               SSC30:
6076 034002                               1$:
        034002 104377                     ERROR   +377
        034004 000031                     .WORD   31
6077                                     ;PC NOT
6078                                     ;INCREMENTED
6079                                     ;BY 2
6080
6081 034006                               SSCDONE:
        034006 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?).
6082
6083
6084                                     ;:*****
    
```

```

: *TEST 67      DESTINATION MODES, MODE 7 (FL=0), TEST
: *
: * THIS IS A TEST OF DESTINATION MODE 7 USING
: * THE STFPS INSTRUCTION
: *
: *****
TST67: SCOPE
6085 034010 000004
6086 034012
6087 034014 104413
6088 034012 012700 034132
6089 034020 012701 000010
6090 034024 012720 177777
6091 034030 077103
6092 034032 012700 103747
6093 034036 012737 034070 001236
6094 034044 012737 034236 000004
6095 034052 170100
6096 034054 005001
6097 034056 012700 026745
6098 034062 012760 034136 005201
6099 034070 170270 005201
6100 034074 022701 000000
6101 034100 001072
6102 034102 020027 026745
6103 034106 001021
6104 034110 023727 034136 103747
6105 034116 001025
6106 034120 023727 034140 177777
6107 034126 001032
6108 034130 000460
6109
6110
6111 034132 177777 177777
6112 034136 177777 177777 177777
6113 034144 177777
6114 034146 177777 177777
6115
6116 034152 010037 001242
6117 034156 012737 026745 001240
6118 034164
6119 034164 104377
6120 034166 000032
6121
6122
6123
6124 034172 012737 103747 001240
6125 034200 013737 034136 001242
6126 034206
6127 034206 104377
6128 034210 000033
6129 034212 000427

: *TEST 67      DESTINATION MODES, MODE 7 (FL=0), TEST
: *
: * THIS IS A TEST OF DESTINATION MODE 7 USING
: * THE STFPS INSTRUCTION
: *
: *****
TST67: SCOPE
TTC1:
      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      MOV      #TTCB0,R0      ;SET UP THE DATA BUFFER.
      MOV      #10,R1
1$:   MOV      #-1,(R0)+
      SOB     R1,1$
      MOV      #103747,R0
      MOV      #TTC2,@#$TMP2
      MOV      #TTC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
      LDFPS   R0              ;SET UP FPS.
      CLR     R1
      MOV      #TTCB2-5201,R0
      MOV      #TTCB1,5201(R0)

TTC2:  STFPS   @5201(R0)      ;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,@#$TMP4
      MOV     #TTCB2-5201,@#$TMP3
1$:   ERROR   +377
      .WORD   32
      BR     TTCDONE ;R0 BAD

:REPORT RESULT INCORRECT.
TTC15: MOV     #103747,@#$TMP3
      MOV     @#TTCB1,@#$TMP4
1$:   ERROR   +377
      .WORD   33
      BR     TTCDONE ;BAD DATA
  
```

```

6129
6130
6131 :REPORT RESULT INCORRECT.
6132 034214 012737 177777 001240 TTC20: MOV #-1,@#$TMP3
6133 034222 013737 034140 001242 MOV @#TTCB1+2,@#$TMP4
6134 034230 1$:
        ERROR +377
        .WORD 34
6135 : (BUT GR7,FL)
6136 034234 000416 BR TTCDONE ;ST 357 TO 416
6137 ;INTO 417
6138
6139 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
6140 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
6141 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
6142 034236 011604 TTC25: MOV (SP),R4
6143 034240 020427 034072 CMP R4,#TTC2+2
6144 034244 001402 BEQ 1$
6145 034246 000137 046310 JMP @#CPSPUR
6146 034252 011637 001236 1$: MOV (SP),@#$TMP2
6147 034256 022626 CMP (SP)+,(SP)+
6148 034260 2$:
        ERROR +377
        .WORD 35
6149 : (BUT FSDT)+ ST634
6150 034264 000402 BR TTCDONE
6151
6152 :REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
6153 034266 TTC30:
6154 034266 1$:
        ERROR +377
        .WORD 36
6155 :PC NOT
6156 :INCREMENTED
6157 034272 TTCDONE:
        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?).
6158
6165 :*****
        :*TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST
        :*
        :* THIS IS A TEST OF DESTINATION MODE
        :* 2 USING STCOL WITH REGISTER 0
        :*
        :*****
6166 034274 000004 TST70: SCOPE
        UUC1:
        LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
        MOV #300,R0 ;SET UP FPS.
        LDFPS R0
        MOV #UUCTP1,R0 ;SET UP THE ACO OPERAND.
        LDD (R0),ACO
        MOV #UUC2,@#$TMP2
        MOV #UUCBFO,R0
    
```

```

6173
6174 034326 175420          UUC2:  STCDL  ACO,(RO)+      ;TEST INSTRUCTION.
6175
6176 034330 020027 034374      CMP    RO,#UUCBFO+4      ;IS RO CORRECT?
6177 034334 001420          BEQ    UUCDONE           ;BRANCH IF CORRECT.
6178
6179                          ;REPORT RO INCORRECT.
6180 034336 010037 001242      UUC3:  MOV    RO,@#$TMP4
6181 034342 012737 034374 001240  MOV    #UUCBFO+4,@#$TMP3
6182 034350          1$:
        034350 104377          ERROR  +377
        034352 000037          .WORD  37
6183
6184 034354 000410          BR     UUCDONE           ;RO NOT INCR BY 4
6185
6186 034356 000000 000000 000000  UUCTP1: .WORD  0,0,0,0
        034364 000000          ;TEST DATA BUFFER:
6187 034366 177777          -1
6188 034370 177777 177777 177777  UUCBFO: .WORD  -1,-1,-1
6189
6190 034376          UUCDONE:
        034376 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?).
    
```

```

6191
6198
        ;*****
        ;*TEST 71      DESTINATION MODES, MODE 4 (FL=1), TEST
        ;*
        ;* THIS IS A TEST OF DESTINATION MODE
        ;* 4 USING STCDL WITH REGISTER 0
        ;*
        ;*****
    
```

```

6199 034400 000004          TST71: SCOPE
6200 034402          VVC1:
        034402 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6201 034404 012700 000300      MOV    #300,RO        ;SET UP FPS.
6202 034410 170100          LDFPS  RO
6203 034412 012700 034462      MOV    #VVC1,RO      ;SET UP THE ACO OPERAND.
6204 034416 172410          LDD    (RO),ACO
6205 034420 012737 034432 001236  MOV    #VVC2,@#$TMP2
6206 034426 012700 034500      MOV    #VVCBFO+4,RO
6207
6208 034432 175440          VVC2:  STCDL  ACO,-(RO)      ;TEST INSTRUCTION.
6209
6210 034434 020027 034474      CMP    RO,#VVCBFO    ;IS RO CORRECT?
6211 034440 001420          BEQ    VVCDONE
6212
6213                          ;REPORT RO INCORRECT.
6214 034442 010037 001242      VVC3:  MOV    RO,@#$TMP4
6215 034446 012737 034474 001240  MOV    #VVCBFO,@#$TMP3
6216 034454          1$:
        034454 104377          ERROR  +377
        034456 000040          .WORD  40
6217
        ;RO NOT DECR BY 4
    
```

6218 034460 000410 BR VVCDONE
6219 ;TEST DATA BUFFER:
6220 034462 000000 000000 000000 VVCTP1: .WORD 0,0,0,0
034470 000000
6221 034472 177777 -1
6222 034474 177777 177777 177777 VVCBF0: .WORD -1,-1,-1
6223
6224 034502 VVCDONE:
034502 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?).

6225
6235

;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.
;*

TST72: SCOPE

034504 000004

6236
6237 ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
6238 WWC1:

6239 034506 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
034510 004737 035654 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6240 034514 020000 000000 000000 1\$: .WORD 2000,0,0,0 ;ACO OPERAND.
034522 000000
6241 034524 000000 000000 2\$: .WORD 0,0 ;EXPECTED RESULT.
6242 034530 177777 177777 3\$: .WORD -1,-1 ;ERROR RES.
6243 034534 040300 4\$: 40300 ;FPS BEFORE EXECUTION.
6244 034536 040304 40304 ;FPS AFTER EXECUTION.
6245 034540 140304 140304 ;ANTICIPATED ERRONEOUS FPS.
6246 034542 177777 -1 ;REPORT RESULT INCORRECT.
6247 034544 104322 5\$: ERROR +322 ;RESULT INCORP.
6248 034546 000401 BR 6\$
6249 034550 104325 6\$: ERROR +325 ;EITHER (BUT FLAG)
6250 034552 ;ST 662
6251 ;OR CLEAR FLAG
6252 ;ST 774
6253

6254 ;EXP=0 (OCT) FL=1 FIC=0
6255 WWC2:

6256 034552 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
034554 004737 035654 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6257 034560 040000 000000 000000 1\$: .WORD 4000,0,0,0 ;AC ;ACO OPERAND.
034566 000000
6258 034570 000000 000000 2\$: .WORD 0,0 ;EXPECTED RESULT.
6259 034574 177777 177777 3\$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6260 034600 040313 4\$: 40313 ;FPS BEFORE EXECUTION.
6261 034602 040304 40304 ;FPS AFTER EXECUTION.
6262 034604 140304 140304 ;ANTICIPATED ERRONEOUS FPS.
6263 034606 177777 -1 ;EXPECTED FEC.


```
6315 ;EXP=30 (OCT) FL=1 FIC=1
6316 034772 WWC7: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
034772 104413 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6317 034774 004737 035654 .WORD 46000,1,0,0 ;ACO OPERAND.
6318 035000 046000 000001 000000 1$:
035006 000000 2$: .WORD 200,1 ;EXPECTED RESULT.
6319 035010 000200 000001 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6320 035014 177777 177777 4$: 40700 ;FPS BEFORE EXECUTION.
6321 035020 040700 40700 ;FPS AFTER EXECUTION.
6322 035022 040700 -1 ;ANTICIPATED ERRONEOUS FPS.
6323 035024 177777 -1 ;EXPECTED FEC.
6324 035026 177777 5$: ERROR +322 ;REPORT RESULT INCORRECT.
6325 035030 104322 BR 6$
6326 035032 000401 ERROR +323 ;REPORT FPS INCORRECT.
6327 035034 104323 6$:
6328 035036
6329
6330 ;EXP=27 (OCT) FL=1 FIC=1
6331 035036 WWC8: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
035036 104413 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6332 035040 004737 035654 .WORD 45600,1,0,0 ;ACO OPERAND.
6333 035044 045600 000001 000000 1$:
035052 000000 2$: .WORD 100,0 ;EXPECTED RESULT.
6334 035054 000100 000000 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6335 035060 177777 177777 4$: 40707 ;FPS BEFORE EXECUTION.
6336 035064 040707 40700 ;FPS AFTER EXECUTION.
6337 035066 040700 -1 ;ANTICIPATED ERRONEOUS FPS.
6338 035070 177777 -1 ;EXPECTED FEC.
6339 035072 177777 5$: ERROR +322 ;REPORT RESULT INCORRECT.
6340 035074 104322 BR 6$
6341 035076 000401 ERROR +323 ;REPORT FPS INCORRECT.
6342 035100 104323 6$:
6343 035102
6344
6345 ;EXP=17 (OCT) FL=0 FIC=1
6346 035102 WWC9: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
035102 104413 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6347 035104 004737 035654 .WORD 43600,0,0,0 ;ACO OPERAND.
6348 035110 043600 000000 000000 1$:
035116 000000 2$: .WORD 40000,-1 ;EXPECTED RESULT.
6349 035120 040000 177777 3$: .WORD 0,-1 ;ANTICIPATED ERRONEOUS RESULT.
6350 035124 000000 177777 4$: 40600 ;FPS BEFORE EXECUTION.
6351 035130 040600 40600 ;FPS AFTER EXECUTION.
6352 035132 040600 140604 ;ANTICIPATED ERRONEOUS FPS.
6353 035134 140604 -1 ;EXPECTED FEC.
6354 035136 177777 5$: ERROR +332 ;BAD CONSTANT ST 066
6355 035140 104332 BR 6$
6356 035142 000401 ERROR +333 ;REPORT FPS INCORRECT.
6357 035144 104333 6$:
6358 035146
6359
6360 ;EXP=20 (OCT) FL=0 FIC=1
6361 035146 WWC10: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
035146 104413 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6362 035150 004737 035654 .WORD 44000,0,0,0 ;ACO OPERAND.
6363 035154 044000 000000 000000 1$:
035162 000000
```

6364	035164	000000	177777			2\$:	.WORD	0,-1		:EXPECTED RESULT.
6365	035170	177777	177777			3\$:	.WORD	-1,-1		:ANTICIPATED ERRONEOUS RESULT.
6366	035174	040600				4\$:	40600			:FPS BEFORE EXECUTION.
6367	035176	140605					140605			:FPS AFTER EXECUTION.
6368	035200	040600					40600			:ANTICIPATED ERRONEOUS FPS.
6369	035202	000006					6		:EXPECTED FEC.	
6370	035204	104322				5\$:	ERROR	+322		:REPORT RESULT INCORRECT.
6371	035206	000401					BR	6\$		
6372	035210	104334					ERROR	+334		:BAD CONSTANT ST 066
6373	035212					6\$:				
6374										
6375										
6376	035212								:EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1	
									WVC11:	
		104413					LPERR			:SET UP THE LOOP ON ERROR ADDRESS.
6377	035214	004737	035654				JSR	PC,@#STCSUB		:GO EXECUTE THE INSTRUCTION.
6378	035220	142000	000000	000000		1\$:	.WORD	142000,0,0,0		:ACO OPERAND.
		035226	000000							
6379	035230	177600	177777			2\$:	.WORD	177600,-1		:EXPECTED RESULT.
6380	035234	000200	000000			3\$:	.WORD	200,0		:ANTICIPATED ERRONEOUS RESULT.
6381	035240	040600				4\$:	40600			:FPS BEFORE EXECUTION.
6382	035242	040610					40610			:FPS AFTER EXECUTION.
6383	035244	040600					40600			:ANTICIPATED ERRONEOUS FPS.
6384	035246	177777					-1		:EXPECTED FEC.	
6385	035250	104335				5\$:	ERROR	+335		:(BUT ENBT) ST 632
6386	035252	000401					BR	6\$		
6387	035254	104336					ERROR	+336		:(SET FN) ST 473
6388	035256					6\$:				
6389										
6390									:EXP=37 (OCT), FL=1, FIC=1, AC NEG.	
6391	035256								WVC12:	
		104413					LPERR			:SET UP THE LOOP ON ERROR ADDRESS.
6392	035260	004737	035654				JSR	PC,@#STCSUB		:GO EXECUTE THE INSTRUCTION.
6393	035264	147600	000000	000000		1\$:	.WORD	147600,0,0,0		:ACO OPERAND.
		035272	000000							
6394	035274	140000	000000			2\$:	.WORD	140000,0		:EXPECTED RESULT.
6395	035300	137777	000000			3\$:	.WORD	137777,0		:ANTICIPATED ERRONEOUS RESULT.
6396	035304	040700				4\$:	40700			:FPS BEFORE EXECUTION.
6397	035306	040710					40710			:FPS AFTER EXECUTION.
6398	035310	177777					-1			:ANTICIPATED ERRONEOUS FPS.
6399	035312	177777					-1		:EXPECTED FEC.	
6400	035314	104337				5\$:	ERROR	+337		:(BUT COUT) ST 375
6401	035316	000401					BR	6\$:ST 275 TO 074
6402	035320	104323					ERROR	+323		:INTO 274
6403	035322					6\$:				
6404										
6405									:EXP=37 (OCT), FL=1, FIC=1, AC NEG	
6406	035322								WVC13:	
		104413					LPERR			:SET UP THE LOOP ON ERROR ADDRESS.
6407	035324	004737	035654				JSR	PC,@#STCSUB		:GO EXECUTE THE INSTRUCTION.
6408	035330	147600	000000	001000		1\$:	.WORD	147600,0,1000,0		:ACO OPERAND.
		035336	000000							
6409	035340	137777	177777			2\$:	.WORD	137777,177777		:EXPECTED RESULT.
6410	035344	140000	177777			3\$:	.WORD	140000,177777		:ANTICIPATED ERRONEOUS RESULT.
6411	035350	040707				4\$:	40707			:FPS BEFORE EXECUTION.
6412	035352	040710					40710			:FPS AFTER EXECUTION.
6413	035354	177777					-1			:ANTICIPATED ERRONEOUS FPS.
6414	035356	177777					-1		:EXPECTED FEC.	


```

6415 035360 104340          5$:      ERROR  +340          ;(BUT COUT) ST 375
6416 035362 000401          6$:      BR          6$          ;TO 274 INTO 074
6417 035364 104323          ERROR  +323          ;REPORT FPS INCORRECT.
6418 035366
6419
6420          ;EXP=41 (OCT), AC NEG, FL=1, FIC=1
6421 035366          WWC14:
      035366 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6422 035370 004737 035654          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6423 035374 150200 000000 000000 1$:      .WORD 150200,0,0,0 ;ACO OPERAND.
      035402 000000
6424 035404 000000 000000 2$:      .WORD 0,0 ;EXPECTED RESULT.
6425 035410 177777 177777 3$:      .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6426 035414 040700          4$:      40700 ;FPS BEFORE EXECUTION.
6427 035416 140705          140705 ;FPS AFTER EXECUTION.
6428 035420 177777          -1 ;ANTICIPATED ERRONEOUS FPS.
6429 035422 000006          6 ;EXPECTED FEC.
6430 035424 104322          5$:      ERROR  +322          ;REPORT RESULT INCORRECT.
6431 035426 000401          BR          6$
6432 035430 104341          ERROR  +341          ;(BUT EZBT) ST 377
6433 035432
6434          ;EXP=40 (OCT), AC NEG, FL=1, FIC=1
6435 035432          WWC15:
      035432 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6436 035434 004737 035654          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6437 035440 150000 000001 000000 1$:      .WORD 150000,1,0,0 ;ACO OPERAND.
      035446 000000
6438 035450 000000 000000 2$:      .WORD 0,0 ;EXPECTED RESULT.
6439 035454 100000 177600 3$:      .WORD 100000,-200 ;ANTICIPATED ERRONEOUS RESULT.
6440 035460 040700          4$:      40700 ;FPS BEFORE EXECUTION.
6441 035462 140705          140705 ;FPS AFTER EXECUTION.
6442 035464 040700          40700 ;ANTICIPATED ERRONEOUS FPS.
6443 035466 000006          6 ;EXPECTED FEC.
6444 035470 104342          5$:      ERROR  +342          ;(BUT COUT) ST 360
6445 035472 000401          BR          6$          ;TO 654 INTO 454
6446 035474 104323          ERROR  +323          ;REPORT FPS INCORRECT.
6447 035476
6448
6449          ;EXP=40, AC NEGATIVE, FL=1, FIC=1
6450 035476          WWC16:
      035476 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
6451 035500 004737 035654          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6452 035504 150001 000000 000000 1$:      .WORD 150001,0,0,0 ;ACO OPERAND.
      035512 000000
6453 035514 000000 000000 2$:      .WORD 0,0 ;EXPECTED RESULT.
6454 035520 077400 000000 3$:      .WORD 77400,0 ;ANTICIPATED ERRONEOUS RESULT.
6455 035524 040700          4$:      40700 ;FPS BEFORE EXECUTION.
6456 035526 140705          140705 ;FPS AFTER EXECUTION.
6457 035530 177777          -1 ;ANTICIPATED ERRONEOUS FPS.
6458 035532 000006          6 ;EXPECTED FEC.
6459 035534 104343          5$:      ERROR  +343          ;REPORT RESULT INCORRECT.
6460 035536 000401          BR          6$
6461 035540 104323          ERROR  +323          ;REPORT FPS INCORRECT.
6462 035542
6463
6464
6465          ;EXP 40 (OCT), AC MOST NEG LONG INT, FL=1
  
```

```

6466 ;FIC=1
6467 035542 WWC17:
6468 035542 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6469 035544 004737 035654 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6469 035550 150000 000000 000000 1$: .WORD 150000,0,0,0 ;ACO OPERAND.
6470 035556 000000 2$: .WORD 100000,0 ;EXPECTED RESULT.
6471 035560 100000 000000 3$: .WORD 0,0 ;ANTICIPATED ERRONEOUS RESULT.
6472 035570 040700 4$: 40700 ;FPS BEFORE EXECUTION.
6473 035572 040710 40710 ;FPS AFTER EXECUTION.
6474 035574 140705 140705 ;ANTICIPATED ERRONEOUS FPS.
6475 035576 177777 -1 ;EXPECTED FEC.
6476 035600 104344 5$: ERROR +344 ;(BUT NBIT) ST 654
6477 035602 000401 BR 6$ ;OR (BUT COUT) ST 454
6478 035604 104323 ERROR +323 ;REPORT FPS INCORRECT.
6479 035606 6$:
6480
6481 ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
6482
6483 035606 WWC18:
6484 035606 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6485 035610 004737 035654 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
6485 035614 144000 000001 000000 1$: .WORD 144000,1,0,0 ;ACO OPERAND.
6486 035622 000000 2$: .WORD 100000,-1 ;EXPECTED RESULT.
6487 035630 100000 177400 3$: .WORD 100000,177400 ;ANTICIPATED ERRONEOUS RESULT.
6488 035634 040600 4$: 40600 ;FPS BEFORE EXECUTION.
6489 035636 040610 40610 ;FPS AFTER EXECUTION.
6490 035640 140605 140605 ;ANTICIPATED ERRONEOUS FPS.
6491 035642 177777 -1 ;EXPECTED FEC.
6492 035644 104345 5$: ERROR +345 ;(BUT FL) ST 633
6493 035646 000401 BR 6$ ;TO 655 INTO 654
6494 035650 104323 ERROR +323 ;REPORT FPS INCORRECT.
6495
6496 035652 000534 6$: BR WWC DONE
6497
6498 ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
6499 ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS. A CALL
6500 ;TO IT IS MADE THUS:
6501
6502 JSR PC,@#STCSUB
6503 ACARG: .WORD X,X,X,X ;AC OPERAND
6504 RES: .WORD X,X ;EXPECTED RESULT
6505 ERRES: .WORD X,X ;ERROR RESULT
6506 FPSB: .WORD X ;FPS BEFORE EXECUTION
6507 FPSA: .WORD X ;FPS AFTER EXECUTION
6508 ERFPS: .WORD X ;ERROR FPS.
6509 FEC: .WORD X ;EXPECTED FEC
6510 ERR1: ERROR +X ;DATA ERROR.
6511 BR CONT
6512 ERR2: ERROR +X ;FPS ERROR.
6513 CONT: ;RETURN ADDRESS
6514
6515 ;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
6516 ;THE STCDI OR STCDL INSTRUCTION IS EXECUTED.
6517 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
6518 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
  
```

6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530 035654 012601
6531 035656 012700 000200
6532 035662 170100
6533 035664 010100
6534 035666 172410
6535 035670 012702 036134
6536 035674 012700 000004
6537 035700 012722 177777
6538 035704 077003
6539 035706 016100 000020
6540 035712 170100
6541 035714 012737 035726 001236
6542 035722 012700 036134
6543 035726 175410
6544
6545 035730 170204
6546 035732 170305
6547 035734 010102
6548 035736 010237 001240
6549 035742 062702 000010
6550 035746 010237 001244
6551 035752 012737 036134 001242
6552 035760 010437 001250
6553 035764 016137 000022 001252
6554 035772 010102
6555 035774 062702 000010
6556 036000 012700 036134
6557 036004 012703 000002
6558 036010 022022
6559 036012 001014
6560 036014 077303
6561 036016 016102 000022
6562 036022 020204
6563 036024 001025
6564 036026 005702
6565 036030 100003
6566 036032 026105 000026
6567 036036 001027
6568
6569 036040 000161 000036
6570
6571
6572 036044 010102
6573 036046 062702 000014
6574 036052 012700 036134
6575 036056 012703 000002

```

;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      #2$,@#$TMP2
        MOV      #STCIBF,R0
2$:     STCDL   ACO,(R0)     ;TEST INSTRUCTION.

        STFPS   R4          ;GET THE FPS.
        STST   R5          ;GET THE FEC.
        MOV      R1,R2
        MOV      R2,@#$TMP3
        ADD     #10,R2
        MOV      R2,@#$TMP5
        MOV      #STCIBF,@#$TMP4
        MOV      R4,@#$TMP7
        MOV      22(R1),@#$TMP10
        MOV      R1,R2
        ADD     #10,R2
        MOV      #STCIBF,R0  ;SEE IF THE RESULT IS CORRECT.
        MOV      #2,R3
3$:     CMP     (R0)+,(R2)+
        BNE    15$
        SOB     R3,3$
        MOV      22(R1),R2
        CMP     R2,R4        ;SEE IF THE FPS IS CORRECT.
        BNE    20$         ;BRANCH IF INCORRECT.
        TST    R2
        BPL    4$
        CMP     26(R1),R5   ;SEE IF THE FEC IS CORRECT.
        BNE    25$         ;BRANCH IF INCORRECT.

4$:     JMP     36(R1)      ;RETURN.
;DATA ERROR:
;SEE IF THE FAILURE WAS ANTICIPATED.
15$:    MOV      R1,R2
        ADD     #14,R2
        MOV      #STCIBF,R0
        MOV      #2,R3

```

6576 036062 022022
 6577 036064 001003
 6578 036066 077303
 6579 036070 000161 000030
 6580 036074
 6581
 6582 036074 104322
 6583 036076 000760
 6584
 6585
 6586 036100 020461 000024
 6587 036104 001002
 6588 036106 000161 000034
 6589 036112
 6590
 6591 036112 104323
 6592 036114 000751
 6593
 6594
 6595 036116 016137 000026 001256
 6596 036124 010537 001254
 6597 036130 104324
 6598 036132 000742
 6599
 6600
 6601 036134 177777 177777 177777
 036142 177777
 6602
 6603 036144
 036144 104412

16\$: CMP (R0)+,(R2)+
 BNE 17\$
 SOB R3,16\$
 JMP 30(R1)
 17\$:
 ;FAILURE WAS NOT ANTICIPATED SO REPORT INCORRECT RESULT HERE.
 18\$: ERROR +322 ;DATA BAD
 BR 4\$
 ;FPS INCORRECT, SO SEE IF FAILURE WAS ANTICIPATED.
 20\$: CMP R4,24(R1)
 BNE 21\$
 JMP 34(R1)
 21\$:
 ;NOT ANTICIPATED SO REPORT BAD FPS HERE.
 22\$: ERROR +323 ;FPS BAD
 BR 4\$
 ;REPORT INCORRECT FEC.
 25\$: MOV 26(R1),@#\$TMP12
 MOV R5,@#\$TMP11
 26\$: ERROR +324
 BR 4\$

;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?).

6604
 6605
 6613

::*****
 ;*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.
 ;*
 ;*****

036146 000004
 6614
 6615
 6616
 6617 036150
 036150 104413
 6618 036152 004737 035654
 6619 036156 047777 177777 177777
 036164 177777
 6620 036166 077777 177600
 6621 036172 077777 177777
 6622 036176 040100
 6623 036200 040100

TST73: SCOPE
 ;EXPONENT=37, FL=1
 XXC1:
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
 1\$: .WORD 47777,-1,-1,-1 ;ACO OPERAND.
 2\$: .WORD 77777,177600 ;EXPECTED RESULT.
 3\$: .WORD 77777,177777 ;ANTICIPATED ERRONEOUS RESULT.
 4\$: 40100 ;FPS BEFORE EXECUTION.
 40100 ;FPS AFTER EXECUTION.

6624 036202 177777
6625 036204 177777
6626 036206 104346
6627 036210 000401
6628 036212 104323
6629 036214
6630
6631 036214 104412
036214

-1
-1
5\$: ERROR +346
BR 6\$
ERROR +323
6\$:
XXCDONE:
RSETUP

:ANTICIPATED ERRONEOUS FPS.
:EXPECTED FEC.
:X11(1,0)+0 ST 773X
:REPORT FPS INCORRECT.
: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?).

6632
6633
6640

:*****
:*TEST 74 STEXP TEST
:*
:* THIS IS A TEST OF THE STEXP
:* INSTRUCTION
:*
:*****

036216 000004
6641
6642
6643 036220
036220 104413
6644 036222 004737 036506
6645 036226 020000 000000 000000
036234 000000
6646 036236 177700
6647 036240 052525
6648 036242 040000
6649 036244 040010
6650 036246 040000
6651 036250 104347
6652 036252 000401
6653 036254 104352
6654 036256
6655
6656
6657 036256
036256 104413
6658 036260 004737 036506
6659 036264 040000 000000 000000
036272 000000
6660 036274 000000
6661 036276 052525
6662 036300 040000
6663 036302 040004
6664 036304 040000
6665 036306 104347
6666 036310 000401
6667 036312 104353
6668
6669 036314
6670

TST74: SCOPE
: EXP = 100 (EXCESS 200)
YYC1:
LPERR
JSR PC,@#STXSUB ;SET UP THE LOOP ON ERROR ADDRESS.
.WORD 2000,0,0,0 ;AC
2\$: -100 ;EXP RES
3\$: 52525 ;ERROR EXP.
4\$: 40000 ;FPSB
40010 ;FPSA
40000 ;ERROR FPS
5\$: ERROR +347 ;BAD EXP
BR 6\$
ERROR +352 ;+(BUT ENBT) ST 376
6\$:
: EXP = 200 (EXCESS 200)
YYC2:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
.WORD 4000,0,0,0 ;ACO OPERAND.
2\$: 0 ;EXPECTED EXPONENT RESULT.
3\$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
4\$: 40000 ;FPS BEFORE EXECUTION.
40004 ;FPS AFTER EXECUTION.
40000 ;ANTICIPATED ERRONEOUS FPS.
5\$: ERROR +347 ;REPORT RESULT INCORRECT.
BR 6\$
ERROR +353 ;(BUT EZBT) ST 071
;(TO 072 INT 272
6\$:

```

6671 ; EXP = 201 (EXCESS 200)
6672
6673 036314 104413 YYC3: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
036314 004737 036506 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
6674 036316 004737 036506 1$: .WORD 40200,0,0,0 ;ACO OPERAND.
6675 036322 040200 000000 000000 2$: 1 ;EXPECTED EXPONENT RESULT.
036330 000000 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
6676 036332 000001 4$: 40000 ;FPS BEFORE EXECUTION.
6677 036334 052525 5$: 40000 ;FPS AFTER EXECUTION.
6678 036336 040000 6$: 40004 ;ANTICIPATED ERRONEOUS FPS.
6679 036340 040000 7$: ERROR +347 ;REPORT RESULT INCORRECT.
6680 036342 040004 8$: BR 6$ ;(BUT EZBT) ST 071
6681 036344 104347 9$: ERROR +354 ;TO 272 INTO 072
6682 036346 000401
6683 036350 104354
6684 036352
6685
6686 ; EXP = 375 (EXCESS 200)
6687
6688 036352 104413 YYC4: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
036352 004737 036506 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
6689 036354 004737 036506 1$: .WORD 77200,0,0,0 ;ACO OPERAND.
6690 036360 077200 000000 000000 2$: 175 ;EXPECTED EXPONENT RESULT.
036366 000000 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
6691 036370 000175 4$: 40000 ;FPS BEFORE EXECUTION.
6692 036372 052525 5$: 40000 ;FPS AFTER EXECUTION.
6693 036374 040000 6$: 40010 ;ANTICIPATED ERRONEOUS FPS.
6694 036376 040000
6695 036400 040010
  
```

```

6697 036402 104347      5$:      ERROR  +347      ;REPORT RESULT INCORRECT.
6698 036404 000401      BR          6$
6699 036406 104355      ERROR  +355      ;(BUT ENBT) ST 376
6700 036410      6$:      ;TO 471 INTO 071
6701
6702      ; EXP = 1 (EXCESS 200)
6703
6704 036410      YYC5:
      036410 104413      LPERR
6705 036412 004737 036506      JSR      PC,@#STXSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
      000000 000000 1$:      .WORD  200,0,0,0      ;GO EXECUTE THE INSTRUCTION.
      036424 000000      ;ACO OPERAND.
6707 036426 177601      2$:      -177      ;EXPECTED EXPONENT RESULT.
6708 036430 052525      3$:      52525      ;ANTICIPATED ERRONEOUS RESULT.
6709 036432 040000      4$:      40000      ;FPS BEFORE EXECUTION.
6710 036434 040010      ;FPS AFTER EXECUTION.
6711 036436 040000      ;ANTICIPATED ERRONEOUS FPS.
6712 036440 104347      5$:      ERROR  +347      ;REPORT RESULT INCORRECT.
6713 036442 000401      BR          6$
6714 036444 104352      ERROR  +352      ;REPORT FPS INCORRECT.
6715 036446      6$:
6716
6717      ; EXP = 156 (EXCESS 200)
6718
6719 036446      YYC6:
      036446 104413      LPERR
6720 036450 004737 036506      JSR      PC,@#STXSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
      036454 033400 000000 000000 1$:      .WORD  33400,0,0,0      ;GO EXECUTE THE INSTRUCTION.
      036462 000000      ;ACO OPERAND.
6722 036464 177756      2$:      -22      ;EXPECTED EXPONENT RESULT.
6723 036466 052525      3$:      52525      ;ANTICIPATED ERRONEOUS RESULT.
6724 036470 047707      4$:      47707      ;FPS BEFORE EXECUTION.
6725 036472 047710      ;FPS AFTER EXECUTION.
6726 036474 177777      -1      ;ANTICIPATED ERRONEOUS FPS.
6727 036476 104347      5$:      ERROR  +347      ;REPORT RESULT INCORRECT.
6728 036500 000401      BR          6$
6729 036502 104350      ERROR  +350      ;REPORT FPS INCORRECT.
6730
6731 036504 000510      6$:      BR      YYCDONE
  
```

```

6732      ;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
6733      ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
6734      ;TO IT IS MADE THUS:
6735
  
```

```

6736      :
6737      :      JSR      PC,@#STXSUB
6738      :      ACARG:  .WORD  X,X,X,X      ;AC OPERAND
6739      :      RES:    .WORD  X      ;EXPECTED RESULT
6740      :      ERRES:  .WORD  X      ;ERROR RESULT
6741      :      FPSB:   .WORD  X      ;FPS BEFORE EXECUTION
6742      :      FPSA:   .WORD  X      ;FPS AFTER EXECUTION
6743      :      ERFPS:  .WORD  X      ;ERROR FPS.
6744      :      ERR1:   ERROR  +X      ;DATA ERROR.
6745      :      BR      CONT
6746      :      ERR2:   ERROR  +X      ;FPS ERROR.
6747      :      CONT:   ;RETURN ADDRESS
  
```

```

6748      ;THE OPERANDS ARE SET UP (USING ACO AS THE ACCUMULATOR). THEN
6749
  
```

```

6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764 036506 012601
6765 036510 010102
6766 036512 010237 001240
6767 036516 062702 000010
6768 036522 012237 001244
6769 036526 012737 036574 001236
6770 036534 012737 123456 036714
6771 036542 012737 076543 036716
6772 036550 012700 000200
6773 036554 170100
6774 036556 010100
6775 036560 172410
6776 036562 016100 000016
6777 036566 170100
6778 036570 012700 036714
6779 036574 175010
6780 036576 170204
6781 036600 010437 001250
6782 036604 016137 000016 001252
6783 036612 013737 036714 001242
6784 036620 026137 000010 036714
6785 036626 001411
6786 036630 026137 000012 036714
6787 036636 001002
6788 036640 000161 000022
6789
6790
6791 036644
6792 036644 104347
6793 036646 000161 000030
6794
6795 036652 020461 000016
6796 036656 001407
6797 036660 020461 000020
6798 036664 001002
6799 036666 000161 000026
6800
6801
6802 036672
6803 036672 104350
6804 036674 000764
6805
6806

```

```

;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
;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.

```

```

STXSUB: MOV      (SP)+,R1      ;GET A POINTER TO THE ARGUMENTS.
        MOV      R1,R2
        MOV      R2,@#STMP3
        ADD      #10,R2
        MOV      (R2)+,@#STMP5
        MOV      #1$,@#STMP2
        MOV      #123456,@#STXBF
        MOV      #76543,@#STXBF+2
        MOV      #200,R0
        LDFPS   R0
        MOV      R1,R0      ;SET UP THE ACO OPERAND.
        LDD     (R0),ACO
        MOV      16(R1),R0  ;SET THE FPS.
        LDFPS   R0
        MOV      #STXBF,R0
1$:     STXP    ACO,(R0)    ;TEST INSTRUCTION.
        STFPS   R4        ;GET FPS.
        MOV      R4,@#STMP7
        MOV      16(R1),@#STMP10
        MOV      @#STXBF,@#STMP4
        CMP     10(R1),@#STXBF ;WAS RESULT CORRECT?
        BEQ     5$        ;BRANCH IF CORRECT.
        CMP     12(R1),@#STXBF ;OTHERWISE SEE IF THE FAILURE WAS ANTICIPATED.
        BNE     2$
        JMP     22(R1)

```

;IF NOT ANTICIPATED REPORT ERROR HERE.

```

2$:
3$:     ERROR   +347      ;EXP BAD
4$:     JMP     30(R1)

```

```

5$:     CMP     R4,16(R1)  ;SEE IF THE FPS IS CORRECT.
        BEQ     10$      ;BRANCH IF CORRECT.
        CMP     R4,20(R1) ;SEE IF THE FAILURE WAS ANTICIPATED.
        BNE     6$
        JMP     26(R1)

```

;FPS ERROR WAS NOT ANTICIPATED SO REPORT ERROR HERE.

```

6$:
7$:     ERROR   +350      ;FPS BAD
        BR     4$

```

;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.


```

6807 036676 022737 076543 036716 10$: CMP #76543,@#STXBF+2
6808 036704 001760 BEQ 4$
6809 036706 104351 11$: ERROR +351 ;FDFL+0 ST 347X
6810 036710 000756 BR 4$
6811
6812 036712 177777 STXBF: -1
6813 036714 177777 177777 177777 .WORD -1,-1,-1,-1,-1
036722 177777 177777
6814
6815 036726 YYCDONE:
036726 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?).
  
```

6816
6827

```

:*****
:*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
:*
:*****
  
```

```

036730 000004
6828
6829 036732 ZTC1:
036732 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6830 036734 012700 040000 MOV #40000,R0 ;SET FPS. FID=1.
6831 036740 170100 LDFPS R0
6832
6833 036742 170003 ZTC2: .WORD 170003 ;ILLEGAL FPP
6834 ;OP CODE
6835 036744 012700 037120 MOV #ZTCBF,R0 ;SET UP THE OUTPUT BUFFER.
6836 036750 012710 177777 MOV #-1,(R0)
6837 036754 012760 177777 000002 MOV #-1,2(R0)
6838 036762 012737 036770 001236 MOV #ZTC3,@#STMP2
6839 036770 170310 ZTC3: STST (R0) ;GET FEC AND
;FEA
6840 ;GET FPS.
6841 036772 170204 STFPS R4
6842 036774 012700 037120 MOV #ZTCBF,R0
6843 037000 011037 001240 MOV (R0),@#STMP3
6844 037004 016037 000002 001242 MOV 2(R0),@#STMP4
6845 037012 012737 000002 001244 MOV #2,@#STMP5
6846 037020 012737 036742 001246 MOV #ZTC2,@#STMP6
6847 037026 010437 001250 MOV R4,@#STMP7
6848 037032 012737 140000 001252 MOV #140000,@#STMP10
6849
6850 037040 022710 000002 CMP #2,(R0) ;SEE IF FEC IS CORRECT.
6851 037044 001010 BNE ZTC5 ;BRANCH IF INCORRECT.
6852 037046 022760 036742 000002 CMP #ZTC2,2(R0) ;SEE IF FEA, ADDRESS, IS CORRECT.
6853 037054 001006 BNE ZTC10 ;BRANCH IF INCORRECT.
6854 037056 022704 140000 CMP #140000,R4 ;SEE IF FPS IS CORRECT.
6855 037062 001013 BNE ZTC15 ;BRANCH IF INCORRECT.
  
```

```
6856 037064 000422 BR ZZCDONE
6857
6858 :REPORT FEC INCORRECT
6859 037066 ZZC5:
6860 037066 104356 1$: ERROR +356 ;STST BAD
6861 037070 000420 BR ZZCDONE ;FECX
6862
6863 :REPORT FEA INCORRECT
6864 037072 022760 177777 000002 ZZC10: CMP #-1,2(RO)
6865 037100 001402 BEQ ZZC12
6866 037102 104357 1$: ERROR +357 ;STST BAD FEA
6867 037104 000412 BR ZZCDONE
6868 037106 ZZC12:
6869 037106 104360 1$: ERROR +360 ;SET FD FL ST 636
6870 037110 000410 BR ZZCDONE
6871
6872 :REPORT FPS INCORRECT
6873 037112 ZZC15:
6874 037112 104361 1$: ERROR +361 ;FPS X AFTER ST ST
6875 037114 000406 BR ZZCDONE
6876
6877 :DATA BUFFER:
6878 037116 177777 -1
6879 037120 177777 177777 177777 ZZCBF: .WORD -1,-1,-1,-1
6880 037130 177777 -1
6881
6882 037132 ZZCDONE:
037132 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?).
```

6891
 6892 037134 000004
 037136
 6893 037136 104413
 037140 005067 140426
 6894 037144 170127 040000
 6895
 6896 037150 012767 077406 133142
 6897 037156 012767 077400 133136
 6898 037164 012767 077400 133132
 6899 037172 012767 077000 133126
 6900 037200 012767 077406 133122
 6901 037206 012767 077406 133122
 6902
 6903 037214 005067 133140
 6904 037220 012767 000200 133134
 6905 037226 012767 000400 133130
 6906 037234 012767 000600 133124
 6907 037242 012767 000600 133120
 6908 037250 012767 177600 133120
 6909
 6910 037256 012767 077406 133014
 6911 037264 012767 077406 133010
 6912 037272 012767 077406 133004
 6913 037300 012767 077006 133000
 6914 037306 012767 077400 132774
 6915 037314 012767 077406 132774
 6916
 6917 037322 005067 133012
 6918 037326 012767 000200 133006
 6919 037334 012767 000400 133002
 6920 037342 012767 000600 132776
 6921 037350 012767 000600 132772
 6922 037356 012767 177600 132772
 6923
 6924 037364 016767 140660 141670
 6925 037372 012701 117760
 6926 037376 012702 117770
 6927 037402 012703 117772
 6928 037406 012737 037716 000250
 6929 037414 012737 000340 000252
 6930 037422 012737 000024 172516
 6931 037430 005237 177572
 6932 037434 012737 117760 117770
 6933
 6934 037442 170000
 6935
 6936
 6937
 6938

```

:*****
:*TEST 76      ENABLE D-SPACE AND SEE I-SPACE IS FORCED
:*
:*THIS IS A TEST THAT WILL ENABLE D-SPACE BUT MAKE IT NON-RESIDENT
:*SO THAT AN INSTRUCTION THAT ACCESSES D-SPACE WHEN IT NORMALLY
:*SHOULDN'T WILL CAUSE A TRAP/ABORT.
:*
:*****
TST76: SCOPE
ZZF1:
LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
CLR        ;MAKE SURE MEMORY MANAGEMENT IS OFF.
MMRO       #40000 ;LOAD FPS STATUS.
LDFPS
MOV        #77406,KDPDR0 ;MAKE KDPDR0 RESIDENT.
MOV        #77400,KDPDR1 ;MAKE KDPDR1 NON-RESIDENT.
MOV        #77400,KDPDR2 ;MAKE KDPDR2 NON-RESIDENT.
MOV        #77000,KDPDR3 ;MAKE KDPDR3 NON-RESIDENT FOR ADDRESSES 60000-77756.
MOV        #77406,KDPDR4 ;MAKE KDPDR4 RESIDENT FOR ADDRESSES 77760-77776.
MOV        #77406,KDPDR7 ;MAKE KDPDR7 RESIDENT (I/O PAGE).
CLR        KDPAR0      ;MAP D-PAGE 0 FOR 0-4K.
MOV        #200,KDPAR1 ;MAP D-PAGE 1 FOR 4-8K.
MOV        #400,KDPAR2 ;MAP D-PAGE 2 FOR 8-12K.
MOV        #600,KDPAR3 ;MAP D-PAGE 3 FOR ACCESSING ADDRESSES 60000-77756.
MOV        #600,KDPAR4 ;MAP D-PAGE 4 FOR ACCESSING ADDRESSES 77760-77776.
MOV        #177600,KDPAR7 ;MAP D-PAGE 7 FOR I/O PAGE.
MOV        #77406,KIPDR0 ;MAKE KIPDR0 RESIDENT.
MOV        #77406,KIPDR1 ;MAKE KIPDR1 RESIDENT.
MOV        #77406,KIPDR2 ;MAKE KIPDR2 RESIDENT.
MOV        #77006,KIPDR3 ;MAKE KIPDR3 RESIDENT FOR USING ADDRESSES 60000-77756.
MOV        #77400,KIPDR4 ;MAKE KIPDR4 NON-RESIDENT FOR USING ADDRESSES 77760-77776.
MOV        #77406,KIPDR7 ;MAKE KIPDR7 RESIDENT (I/O PAGE).
CLR        KIPAR0      ;MAP I-PAGE 0 FOR 0-4K.
MOV        #200,KIPAR1 ;MAP I-PAGE 1 FOR 4-8K.
MOV        #400,KIPAR2 ;MAP I-PAGE 2 FOR 8-12K.
MOV        #600,KIPAR3 ;MAP I-PAGE 3 FOR ACCESSING ADDRESSES 60000-77756.
MOV        #600,KIPAR4 ;MAP I-PAGE 4 FOR ACCESSING ADDRESSES 77760-77776.
MOV        #177600,KIPAR7 ;MAP I-PAGE 7 FOR I/O PAGE.
MOV        MMVECT,$TMP14 ;MOVE MM TRAP VECTOR TO $TMP14 FOR TEMP STORAGE.
MOV        #DATA,R1      ;SET UP R1.
MOV        #DATA+10,R2   ;SET UP R2.
MOV        #DATA+12,R3   ;SET UP R3.
MOV        #TRAPV,@#MMVECT ;SET UP FOR FP TRAPS FOR THIS TEST.
MOV        #340,@#MMVECT+2
MOV        #24,@#MMR3    ;TURN ON 22-BIT KERNEL D-SPACE.
INC        @#MMRO       ;TURN ON MEMORY MANAGEMENT.
MOV        #DATA,@#DATA+10 ;SET UP ADDRESS POINTER.

CFCC      ;* TEST INSTRUCTION WHICH SHOULD NEVER INVOKE D-SPACE.
;* THIS INSTRUCTION WILL TEST FOR A WORST-CASE HARDWARE PROBLEM.

:*****ALL REFERENCES TO MICRO-FLOWS REFER TO *FP11-F-2 REV A* FLOWS*****
;THE COMMENTS FOR EACH TEST LINE ARE WRITTEN SO YOU CAN GO TO THE MICRO FLOWS

```

6939
 6940
 6941
 6942
 6943
 6944
 6945
 6946 037444 010100
 6947 037446 170410
 6948 037450 177010
 6949 037452 172410
 6950 037454 170310
 6951
 6952
 6953
 6954 037456 010100
 6955 037460 170520
 6956 037462 170527 001000
 6957
 6958 037466 010100
 6959 037470 170420
 6960
 6961
 6962
 6963
 6964
 6965 037472 170427 001000
 6966
 6967 037476 010100
 6968 037500 177020
 6969 037502 177027 001000
 6970
 6971 037506 010100
 6972 037510 172420
 6973 037512 172427 042572
 6974
 6975 037516 010100
 6976 037520 170320
 6977 037522 170327 001000
 6978
 6979
 6980
 6981 037526 010200
 6982 037530 170530
 6983 037532 170537 117760
 6984
 6985 037536 010200
 6986 037540 170430
 6987 037542 170437 117760
 6988
 6989 037546 010200
 6990 037550 177030
 6991 037552 177037 117760
 6992
 6993 037556 010200
 6994 037560 172430
 6995 037562 172437 117760

;AND PINPOINT THE PROBLEM AREA. FROM THERE, HARDWARE ANALYSIS SHOULD BE EASIER.
 ;* INSTRUCTION GROUPS ISOLATED BY BLANK LINES ARE TO BE EXECUTED TOGETHER
 ;* DUE TO PROPER SETUP PURPOSES. I.E. THE LOCATION OR ADDRESS HAS TO BE
 ;* INITIALIZED PROPERLY BEFORE THE INSTRUCTION CAN BE ACCOMPLISHED.

;* TESTING MODE 1 REG 0.

MOV R1,R0 ;SETTING UP R0.
 CLRF (R0) ;TESTING BLOCKS 27-K AND 27-R.
 LDCIF (R0),ACO ;TESTING BLOCKS 28-F AND 28-P.
 LDF (R0),ACO ;TESTING BLOCKS 4-J, 4-X, 4-Z AND 4-BB.
 STST (R0) ;TESTING BLOCKS 33-E AND 33-P.

;* TESTING MODE 2 REG 0 AND 7.

MOV R1,R0 ;SETTING UP R0.
 TSTF (R0)+ ;TESTING BLOCK 21-AA.
 TSTF #1000

MOV R1,R0 ;CORRECTING R0.
 CLRF (R0)+ ;TESTING BLOCKS 27-K AND 27-R.
 ;**NOTE**: THE LOCATION AFTER THE CLRF, AND STST MODE 2 REG 7 INSTRUCTIONS
 ;*WILL* BE CHANGED ON SUBSEQUENT PASSES, BUT IS **NOT** INCORRECT. THE
 ;ACTUAL CONTENTS OF THOSE LOCATIONS IS IMMATERIAL, AS THIS TEST INSURES
 ;THAT THE INSTRUCTION DOES EXECUTE WITHOUT ACCESSING THAT LOCATION AS
 ;A D-SPACE ACCESS.
 CLRF #1000

MOV R1,R0 ;CORRECTING R0.
 LDCIF (R0)+,ACO ;TESTING BLOCKS 28-F AND 28-P.
 LDCIF #1000,ACO

MOV R1,R0 ;CORRECTING R0.B
 LDF (R0)+,ACO ;TESTING BLOCKS 4-NN, 4-X, 4-Z AND 4-BB.
 LDF #1000,ACO

MOV R1,R0 ;CORRECTING R0.
 STST (R0)+ ;TESTING BLOCKS 33-J AND 33-P.
 STST #1000

;* TESTING MODE 3 REG 0 AND 7.

MOV R2,R0 ;SETTING UP R0.
 TSTF @ (R0)+ ;TESTING BLOCK 21-N.
 TSTF @#DATA

MOV R2,R0 ;CORRECTING R0.
 CLRF @ (R0)+ ;TESTING BLOCKS 27-U, 27-T AND 27-R.
 CLRF @#DATA

MOV R2,R0 ;CORRECTING R0.
 LDCIF @ (R0)+,ACO ;TESTING BLOCKS 28-L, 28-N AND 28-P.
 LDCIF @#DATA,ACO

MOV R2,R0 ;CORRECTING R0.
 LDF @ (R0)+,ACO ;TESTING BLOCKS 4-R, 4-T, 4-X, 4-Z AND 4-BB.
 LDF @#DATA,ACO

```

6996
6997 037566 010200      MOV      R2,R0      ;CORRECTING R0.
6998 037570 170330      STST     @ (R0)+    ;TESTING BLOCKS 33-L, 33-N AND 33-P.
6999 037572 170337 117760 STST     @#DATA
7000
7001                      ;* TESTING MODE 4 REG 0.
7002
7003 037576 010200      MOV      R2,R0      ;SETTING UP R0.
7004 037600 172440      LDF      -(R0),ACO ;TESTING BLOCKS 4-J, 4-X, 4-Z AND 4-BB.
7005
7006                      ;* TESTING MODE 5 REG 0.
7007
7008 037602 010300      MOV      R3,R0      ;SETTING UP R0.
7009 037604 170550      TSTF    @-(R0)     ;TESTING BLOCK 21-U.
7010
7011 037606 010300      MOV      R3,R0      ;CORRECTING R0.
7012 037610 170450      CLRF    @-(R0)     ;TESTING BLOCKS 27-X, 27-T AND 27-R.
7013
7014 037612 010300      MOV      R3,R0      ;CORRECTING R0.
7015 037614 177050      LDCIF   @-(R0),ACO ;TESTING BLOCKS 28-S, 28-N AND 28-P.
7016
7017 037616 010300      MOV      R3,R0      ;CORRECTING R0.
7018 037620 172450      LDF      @-(R0),ACO ;TESTING BLOCKS 4-U, 4-T, 4-X, 4-Z AND 4-BB.
7019
7020 037622 010300      MOV      R3,R0      ;CORRECTING R0.
7021 037624 170350      STST     @-(R0)     ;TESTING BLOCKS 33-S, 33-N AND 33-P.
7022
7023                      ;* TESTING MODE 6 REG 7.
7024
7025 037626 170567 060126  TSTF    DATA      ;TESTING BLOCK 21-O.
7026 037632 170467 060122  CLRF    DATA      ;TESTING BLOCKS 27-DD, 27-T AND 27-R.
7027 037636 177067 060116  LDCIF   DATA,ACO  ;TESTING BLOCKS 28-T, 28-N AND 28-P.
7028 037642 172467 060112  LDF      DATA,ACO  ;TESTING BLOCKS 4-DD, 4-T, 4-X, 4-Z AND 4-BB.
7029 037646 170367 060106  STST     DATA      ;TESTING BLOCKS 33-T, 33-N AND 33-P.
7030
7031                      ;* TESTING MODE 7 REG 0 AND 7.
7032
7033 037652 010200      MOV      R2,R0      ;SETTING UP R0.
7034 037654 170470 000000  CLRF    @0(R0)     ;TESTING BLOCKS 27-GG, 27-JJ, 27-T AND 27-R.
7035 037660 170477 060104  CLRF    @DATA+10
7036 037664 177070 000000  LDCIF   @0(R0),ACO ;TESTING BLOCKS 28-W, 28-Z, 28-N AND 28-P.
7037 037670 177077 060074  LDCIF   @DATA+10,ACO
7038 037674 172470 000000  LDF      @0(R0),ACO ;TESTING BLOCKS 4-GG, 4-JJ, 4-T, 4-X 4-Z AND 4-BB.
7039 037700 172477 060064  LDF      @DATA+10,ACO
7040 037704 170370 000000  STST     @0(R0)     ;TESTING BLOCKS 33-W, 33-Z, 33-N AND 33-P.
7041 037710 170377 060054  STST     @DATA+10
7042
7043 037714 000431      BR       ENDTST     ;BRANCH TO END OF TEST ROUTINE.
7044
7045 037716 042767 000001 137646 TRAPV: BIC     #1,MMR0   ;TURN OFF MEMORY MANAGEMENT.
7046 037724 016767 137642 141306 MOV      MMR0,$TMP3 ;TRANSFER MMR0 TO $TMP3 FOR ERROR PRINTING.
7047 037732 005267 141302      INC     $TMP3      ;REPLACE BIT CLEARED TURNING OFF MEMORY MANAGEMENT.
7048 037736 016767 137634 141272 MOV      MMR2,$TMP2 ;MOVE THE TRAP INSTRUCTION ADDRESS TO $TMP13.
7049 037744 005067 137622      CLR    MMR0       ;CLEAR ERROR BITS.
7050 037750 012667 141312 MOV      (SP)+,$TMP16 ;POP STACK AND SAVE 1ST CONTENTS.
7051 037754 012667 141310 MOV      (SP)+,$TMP17 ;POP STACK AGAIN AND SAVE 2ND CONTENTS.
7052 037760 104362      ERROR   +362      ;FPP TRAP/ABORT ERROR CALL.
  
```

```
7053 037762 016746 141302      MOV    $TMP17,-(SP)  ;PUSH 2ND SAVED CONTENTS BACK ON STACK.
7054 037766 016746 141274      MOV    $TMP16,-(SP)  ;PUSH 1ST SAVED CONTENTS BACK ON STACK.
7055 037772 005267 137574      INC    MMRO          ;TURN ON MEMORY MANAGEMENT.
7056 037776 000002                RTI                ;RETURN FROM INTERRUPT.
7057
7058 040000 005067 137566      ENDTST: CLR    MMRO  ;TURN OFF MEMORY MANAGEMENT.
7059 040004 016767 141252 140236  MOV    $TMP14,MMVECT ;RESTORE MMVECT TO ITS ORIGINAL CONTENTS.
7060 040012                IDONE:
      040012 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?).
```

7061

7071

```

*****
*TEST 77      AUTO INCREMENT/DECREMENT TEST
*
* THIS TEST INSURES THAT AUTO INCREMENT/DECREMENT WORKS PROPERLY AND
* *ONLY* WHEN IT IS SUPPOSED TO. THIS IS DONE BY ENABLING 22-BIT KERNEL
* D-SPACE, BUT MAKING IT NON-RESIDENT, FORCING A MEMORY MANAGEMENT TRAP
* CONDITION. THIS ENABLES EXAMINING OF SR1 FOR PROPER CONTENTS.
*
*****
  
```

```

040014 000004
7072 040016 104413
      040016 005067 137546
7073 040020 005067 170127 040000
7074 040024 170127 040000
7075 040030 005067 141236
7076 040034 012767 000252 002336
7077 040042 012767 125252 002332
7078 040050 012767 125252 002326
7079 040056 012767 125252 002322
7080 040064 172467 002310
7081 040070 172567 002304
7082 040074 172667 002300
7083 040100 172767 002274
7084 040104 012700 042400
7085 040110 012701 000030
7086 040114 005020
7087 040116 077102
7088 040120 174067 002254
7089 040124 174167 002260
7090 040130 174267 002264
7091 040134 174367 002270
7092
7093 040140 012767 077406 132152
7094 040146 012767 077406 132146
7095 040154 012767 077400 132142
7096 040162 012767 077406 132136
7097 040170 012767 077406 132140
7098
7099 040176 012767 077406 132074
7100 040204 012767 077406 132070
7101 040212 012767 077406 132064
7102 040220 012767 077406 132060
7103 040226 012767 077406 132062
7104
7105 040234 005067 132120
7106 040240 012767 000200 132114
7107 040246 012767 000400 132110
7108 040254 012767 000600 132104
7109 040262 012767 177600 132106
7110
7111 040270 005067 132044
7112 040274 012767 000200 132040
7113 040302 012767 000400 132034
7114 040310 012767 000600 132030
7115 040316 012767 177600 132032
7116
7117 040324 012705 042460
  
```

```

TST77: SCOPE
INCDCT:
LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
CLR            ;MAKE SURE MEMORY MANAGEMENT IS OFF.
MMRO          ;LOAD FLOATING POINT STATUS.
#40000        ;CLEAR THE TEMPORARY LOCATION.
LDFPS        ;CLEAR UPPER BYTE - ALTERNATING BITS IN LOWER BYTE.
CLR           ;MOVE ALTERNATING BITS TO 2ND WORD.
$TMP20       ;MOVE ALTERNATING BITS TO 3RD WORD.
#252,STORE+2 ;MOVE ALTERNATING BITS TO 4TH WORD.
#125252,STORE+4
#125252,STORE+6
MOV           ;LOAD AC0.
STORE,AC0    ;LOAD AC1.
LDF          ;LOAD AC2.
STORE,AC1    ;LOAD AC3.
LDF          ;MOVE ADDRESS OF STORE TO R0.
STORE,AC2    ;MOVE LOOP COUNTER (CLEARING 30 WORDS) TO R1.
#STORE,R0    ;CLEAR THE WORD.
#30,R1       ;SUBTRACT 1 FROM R1 AND BRANCH IF NOT 0.
(R0)+        ;STORE AC0.
CLR          ;STORE AC1.
SOB          ;STORE AC2.
R1,1$       ;STORE AC3.
AC0,STORE+10
AC1,STORE+20
AC2,STORE+30
AC3,STORE+30
MOV          ;MAKE KDPDR0 RESIDENT.
#77406,KDPDR0
MOV          ;MAKE KDPDR1 RESIDENT.
#77406,KDPDR1
MOV          ;MAKE KDPDR2 NON-RESIDENT.
#77400,KDPDR2
MOV          ;MAKE KDPDR3 RESIDENT.
#77406,KDPDR3
MOV          ;MAKE KDPDR7 RESIDENT.
#77406,KDPDR7
MOV          ;MAKE KIPDR0 RESIDENT.
#77406,KIPDR0
MOV          ;MAKE KIPDR1 RESIDENT.
#77406,KIPDR1
MOV          ;MAKE KIPDR2 RESIDENT.
#77406,KIPDR2
MOV          ;MAKE KIPDR3 RESIDENT.
#77406,KIPDR3
MOV          ;MAKE KIPDR7 RESIDENT.
#77406,KIPDR7
CLR         ;MAP D-PAGE 0 FOR 0-4K.
KDPAR0      ;MAP D-PAGE 1 FOR 4-8K.
#200,KDPAR1 ;MAP D-PAGE 2 FOR 8-12K.
#400,KDPAR2 ;MAP D-PAGE 3 FOR 12-16K.
#600,KDPAR3 ;MAP D-PAGE 7 FOR I/O PAGE.
#177600,KDPAR7
MOV          ;MAP I-PAGE 0 FOR 0-4K.
KIPAR0      ;MAP I-PAGE 1 FOR 4-8K.
#200,KIPAR1 ;MAP I-PAGE 2 FOR 8-12K.
#400,KIPAR2 ;MAP I-PAGE 3 FOR 12-16K.
#600,KIPAR3 ;MAP I-PAGE 7 FOR I/O PAGE.
#177600,KIPAR7
MOV          ;SET UP BYTE TABLE POINTER R5.
#BYTABL,R5
  
```

```

7118 040330 016767 137714 140724      MOV      MMVECT,$TMP14      ;TEMPORARILY STORE THE MMVECT VALUE.
7119 040336 012737 041230 000250      MOV      #TRPV,@#MMVECT    ;SET UP FOR FP TRAPS FOR THIS TEST.
7120 040344 012737 000340 000252      MOV      #340,@#MMVECT+2
7121 040352 016767 137442 140704      MOV      IOTRAP,$TMP15     ;TEMPORARILY STORE THE IOTRAP VALUE.
7122 040360 012737 041162 000020      MOV      #FALTRP,@#IOTRAP  ;SET UP FOR FAILURE OF TRAPS FOR THIS TEST.
7123 040366 012737 000340 000022      MOV      #340,@#IOTRAP+2
7124
7125 040374 012767 000024 132114      MOV      #24,MMR3          ;TURN ON 22-BIT KERNEL D-SPACE.
7126 040402 012767 042364 001764      MOV      #NODAT,NODAT+10  ;SET UP ADDRESS POINTER.
7127 040410 012700 042364              MOV      #NODAT,R0         ;SET UP R0.
7128 040414 012702 042374              MOV      #NODAT+10,R2     ;SET UP R2.
7129 040420 012703 042376              MOV      #NODAT+12,R3     ;SET UP R3.
7130 040424 010067 002010              MOV      R0,STORE+40      ;STORE R0.
7131 040430 010267 002006              MOV      R2,STORE+42      ;STORE R2.
7132 040434 010367 002004              MOV      R3,STORE+44      ;STORE R3.
7133 040440 005267 137126              INC      MMRO              ;TURN ON MEMORY MANAGEMENT.
7134
7135
7136
7137
7138
7139
7140
7141
7142 040444 170501
7143 040446 170401
7144 040450 177001
7145 040452 172401
7146 040454 170301
7147 040456 005067 137110
7148 040462 172467 001712
7149 040466 172567 001706
7150 040472 005267 137074
7151
7152
7153
7154 040476 010001
7155 040500 010004
7156 040502 170511
7157 040504 000004
7158 040506 170411
7159 040510 000004
7160 040512 177011
7161 040514 000004
7162 040516 172411
7163 040520 000004
7164 040522 170311
7165 040524 000004
7166
7167
7168
7169 040526 170521
7170 040530 000004
7171
7172 040532 010001
7173 040534 170421
7174 040536 000004
  
```

```

;*****ALL REFERENCES TO MICRO-FLOWS REFER TO *FP11-F-2 REV A* FLOWS*****
;THE COMMENTS FOR EACH TEST LINE ARE WRITTEN SO YOU CAN GO TO THE MICRO
;FLOW AND PINPOINT THE PROBLEM AREA. FROM THERE, HARDWARE ANALYSIS SHOULD
;BE EASIER.
  
```

```

;* THE FOLLOWING TESTS ARE FOR MODE 0 REG 1 (THESE SHOULD *NOT* ABORT).
  
```

```

TSTF      R1              ;FDST-NOTCLR PAGE 21.
CLRF      R1              ;FDST MODES PAGE 27.
LDCIF     R1,AC0          ;SOURCE MODES PAGE 28.
LDF       R1,AC0          ;FSRC MODES PAGE 4.
STST      R1              ;DEST MODES PAGE 33.
CLR       MMRO            ;TURN OFF MEMORY MANAGEMENT.
LDF       STORE,AC0       ;RESTORE AC0.
LDF       STORE,AC1       ;RESTORE AC1.
INC       MMRO            ;TURN ON MEMORY MANAGEMENT.
  
```

```

;* THE FOLLOWING TESTS ARE FOR MODE 1 REG 1.
  
```

```

MODE1: MOV      R0,R1          ;SET UP R1.
        MOV      R0,R4          ;MOVE 'START' VALUE INTO R4.
        TSTF     (R1)           ;FDST-NOTCLR PAGE 21.
        IOT      ;FORCE A TRAP.
        CLRF     (R1)           ;FDST MODES PAGE 27.
        IOT      ;FORCE A TRAP.
        LDCIF    (R1),AC0       ;SOURCE MODES PAGE 28.
        IOT      ;FORCE A TRAP.
        LDF      (R1),AC0       ;FSRC MODES PAGE 4.
        IOT      ;FORCE A TRAP.
        STST     (R1)           ;DEST MODES PAGE 33.
        IOT      ;FORCE A TRAP.
  
```

```

;* THE FOLLOWING TESTS ARE FOR MODE 2 REG 1.
  
```

```

LABEL1: TSTF     (R1)+          ;FDST-NOTCLR PAGE 21.
        IOT      ;FORCE A TRAP.
        MOV      R0,R1          ;CORRECT R1.
        CLRF     (R1)+          ;FDST MODES PAGE 27.
        IOT      ;FORCE A TRAP.
  
```



```

7175
7176 040540 010001      MOV      R0,R1      ;CORRECT R1.
7177 040542 177021      LDCIF   (R1)+,ACO  ;SOURCE MODES PAGE 28.
7178 040544 000004      IOT                      ;FORCE A TRAP.
7179
7180 040546 010001      MOV      R0,R1      ;CORRECT R1.
7181 040550 172421      LDF     (R1)+,ACO  ;FSRC MODES PAGE 4.
7182 040552 000004      IOT                      ;FORCE A TRAP.
7183
7184 040554 010001      MOV      R0,R1      ;CORRECT R1.
7185 040556 170321      STST   (R1)+       ;DEST MODES PAGE 33.
7186 040560 000004      IOT                      ;FORCE A TRAP.
7187
7188                      ;* THE FOLLOWING TESTS ARE FOR MODE 3 REG 1 AND 7.
7189
7190 040562 010201      MOV      R2,R1      ;SET UP R1 FOR MODE 3.
7191 040564 010204      MOV      R2,R4      ;MOVE 'START' VALUE INTO R4.
7192 040566 170531      TSTF   @(R1)+      ;FDST-NOTCLR PAGE 21.
7193 040570 000004      IOT                      ;FORCE A TRAP.
7194 040572 170537 042364      TSTF   @#NODAT
7195 040576 000004      IOT                      ;FORCE A TRAP.
7196
7197 040600 010201      MOV      R2,R1      ;CORRECT R1.
7198 040602 170431      CLRF   @(R1)+      ;FDST MODES PAGE 27.
7199 040604 000004      IOT                      ;FORCE A TRAP.
7200 040606 170437 042364      CLRF   @#NODAT
7201 040612 000004      IOT                      ;FORCE A TRAP.
7202
7203 040614 010201      MOV      R2,R1      ;CORRECT R1.
7204 040616 177031      LDCIF   @(R1)+,ACO  ;SOURCE MODES PAGE 28.
7205 040620 000004      IOT                      ;FORCE A TRAP.
7206 040622 177037 042364      LDCIF   @#NODAT,ACO
7207 040626 000004      IOT                      ;FORCE A TRAP.
7208
7209 040630 010201      MOV      R2,R1      ;CORRECT R1.
7210 040632 172431      LDF     @(R1)+,ACO  ;FSRC MODES PAGE 4.
7211 040634 000004      IOT                      ;FORCE A TRAP.
7212 040636 172437 042364      LDF     @#NODAT,ACO
7213 040642 000004      IOT                      ;FORCE A TRAP.
7214
7215 040644 010201      MOV      R2,R1      ;CORRECT R1.
7216 040646 170331      STST   @(R1)+      ;DEST MODES PAGE 33.
7217 040650 000004      IOT                      ;FORCE A TRAP.
7218 040652 170337 042364      STST   @#NODAT
7219 040656 000004      IOT                      ;FORCE A TRAP.
7220
7221                      ;* THE FOLLOWING TESTS ARE FOR MODE 4 REG 1.
7222
7223 040660 010201      MOV      R2,R1      ;SET UP R1 FOR MODE 4.
7224 040662 170541      TSTF   -(R1)       ;FDST-NOTCLR PAGE 21.
7225 040664 000004      IOT                      ;FORCE A TRAP.
7226
7227 040666 010201      MOV      R2,R1      ;CORRECT R1.
7228 040670 170441      CLRF   -(R1)       ;FDST MODES PAGE 27.
7229 040672 000004      IOT                      ;FORCE A TRAP.
7230
7231 040674 010201      MOV      R2,R1      ;CORRECT R1.
  
```

7232	040676	177041	LDCIF	-(R1),ACO	:SOURCE MODES PAGE 28.
7233	040700	000004	IOT		:FORCE A TRAP.
7234					
7235	040702	010201	MOV	R2,R1	:CORRECT R1.
7236	040704	172441	LDF	-(R1),ACO	:FSRC MODES PAGE 4.
7237	040706	000004	IOT		:FORCE A TRAP.
7238					
7239	040710	010201	MOV	R2,R1	:CORRECT R1.
7240	040712	170341	STST	-(R1)	:DEST MODES PAGE 33.
7241	040714	000004	IOT		:FORCE A TRAP.
7242					
7243					
7244					
7245	040716	010301	MOV	R3,R1	:SET UP R1 FOR MODE 5.
7246	040720	010304	MOV	R3,R4	:MOVE 'START' VALUE INTO R4.
7247	040722	170551	TSTF	@-(R1)	:FDST-NOTCLR PAGE 21.
7248	040724	000004	IOT		:FORCE A TRAP.
7249					
7250	040726	010301	MOV	R3,R1	:CORRECT R1.
7251	040730	170451	CLRF	@-(R1)	:FDST MODES PAGE 27.
7252	040732	000004	IOT		:FORCE A TRAP.
7253					
7254	040734	010301	MOV	R3,R1	:CORRECT R1.
7255	040736	177051	LDCIF	@-(R1),ACO	:SOURCE MODES PAGE 28.
7256	040740	000004	IOT		:FORCE A TRAP.
7257					
7258	040742	010301	MOV	R3,R1	:CORRECT R1.
7259	040744	172451	LDF	@-(R1),ACO	:FSRC MODES PAGE 4.
7260	040746	000004	IOT		:FORCE A TRAP.
7261					
7262	040750	010301	MOV	R3,R1	:CORRECT R1.
7263	040752	170351	STST	@-(R1)	:DEST MODES PAGE 33.
7264	040754	000004	IOT		:FORCE A TRAP.
7265					
7266					
7267					
7268	040756	010001	MOV	R0,R1	:SET UP R1 FOR MODE 6.
7269	040760	010004	MOV	R0,R4	:MOVE 'START' VALUE INTO R4.
7270	040762	170561	TSTF	0(R1)	:FDST-NOTCLR PAGE 21.
7271	040766	000004	IOT		:FORCE A TRAP.
7272	040770	170567	TSTF	NODAT	
7273	040774	000004	IOT		:FORCE A TRAP.
7274	040776	170461	CLRF	0(R1)	:FDST MODES PAGE 27.
7275	041002	000004	IOT		:FORCE A TRAP.
7276	041004	170467	CLRF	NODAT	
7277	041010	000004	IOT		:FORCE A TRAP.
7278	041012	177061	LDCIF	0(R1),ACO	:SOURCE MODES PAGE 28.
7279	041016	000004	IOT		:FORCE A TRAP.
7280	041020	177067	LDCIF	NODAT,ACO	
7281	041024	000004	IOT		:FORCE A TRAP.
7282	041026	172461	LDF	0(R1),ACO	:FSRC MODES PAGE 4.
7283	041032	000004	IOT		:FORCE A TRAP.
7284	041034	172467	LDF	NODAT,ACO	
7285	041040	000004	IOT		:FORCE A TRAP.
7286	041042	170361	STST	0(R1)	:DEST MODES PAGE 33.
7287	041046	000004	IOT		:FORCE A TRAP.
7288	041050	170367	STST	NODAT	

;* THE FOLLOWING TESTS ARE FOR MODE 5 REG 1.

;* THE FOLLOWING TESTS ARE FOR MODE 6 REG 1 AND 7.

```
7289 041054 000004      IOT                ;FORCE A TRAP.
7290
7291                    ;* THE FOLLOWING TESTS ARE FOR MODE 7 REG 1 AND 7.
7292
7293 041056 010201      MOV      R2,R1      ;SET UP R1 FOR MODE 7.
7294 041060 010204      MOV      R2,R4      ;MOVE 'START' VALUE TO R4.
7295 041062 170571 000000  TSTF    @0(R1)      ;FDST-NOTCLR PAGE 21.
7296 041066 000004      IOT                ;FORCE A TRAP.
7297 041070 170577 001300  TSTF    @NODAT+10
7298 041074 000004      IOT                ;FORCE A TRAP.
7299 041076 170471 000000  CLRF    @0(R1)      ;FDST MODES PAGE 27.
7300 041102 000004      IOT                ;FORCE A TRAP.
7301 041104 170477 001264  CLRF    @NODAT+10
7302 041110 000004      IOT                ;FORCE A TRAP.
7303 041112 177071 000000  LDCIF   @0(R1),ACO  ;SOURCE MODES PAGE 28.
7304 041116 000004      IOT                ;FORCE A TRAP.
7305 041120 177077 001250  LDCIF   @NODAT+10,ACO
7306 041124 000004      IOT                ;FORCE A TRAP.
7307 041126 172471 000000  LDF     @0(R1),ACO  ;FSRC MODES PAGE 4.
7308 041132 000004      IOT                ;FORCE A TRAP.
7309 041134 172477 001234  LDF     @NODAT+10,ACO
7310 041140 000004      IOT                ;FORCE A TRAP.
7311 041142 170371 000000  STST    @0(R1)      ;DEST MODES PAGE 33.
7312 041146 000004      IOT                ;FORCE A TRAP.
7313 041150 170377 001220  STST    @NODAT+10
7314 041154 000004      IOT                ;FORCE A TRAP.
7315 041156 000167 001360  JMP     ENDTES      ;BRANCH TO END TEST.
7316
```

```

7318 041162 005067 136404      FALTRP: CLR      MMRO      ;TURN OFF MEMORY MANAGEMENT.
7319 041166 011667 140066      MOV      (SP), $TMP13 ;MOVE NEXT INSTRUCTION ADDRESS TO $TMP13.
7320
7321      ;THIS NEXT SECTION NOW CORRECTS THE CONTENTS OF $TMP13 SO THAT IT POINTS
7322      ;TO THE PREVIOUS FPP INSTRUCTION. IT DOES THIS BY SUBTRACTING 2 FROM THE
7323      ;ADDRESS IN $TMP13, REPLACING THE 170000 THAT THE BIC INSTRUCTION USES,
7324      ;AND BIT CLEARING THE INSTRUCTION WITH 170000. IF THE INSTRUCTION THAT
7325      ;$TMP13 IS POINTING TO IS NOT AN FPP INSTRUCTION, THE 170000 WILL NOT
7326      ;CLEAR, SATISFYING THE NEXT BRANCH. THE ADDRESS IS AGAIN CORRECTED,
7327      ;AND THE TESTING PROCESS STARTS OVER. THIS CONTINUES UNTIL $TMP13 IS
7328      ;POINTING TO AN FPP INSTRUCTION, AND NORMALLY WILL NOT BE EXECUTED MORE
7329      ;THAN THREE TIMES BEFORE FINDING THE INSTRUCTION.
7330 041172 162767 000002 140060 1$: SUB      #2, $TMP13      ;SUBTRACT 2 FROM $TMP13.
7331 041200 012767 170000 000004      MOV      #170000, 2$+4 ;SET UP BIC DATA LOCATION.
7332 041206 047727 140046 170000 2$: BIC      @ $TMP13, #170000 ;TEST TO SEE IF FPP INSTRUCTION.
7333 041214 001366      BNE      1$              ;BRANCH BACK FOR ANOTHER TRY IF NOT.
7334 041216 012767 000364 140046      MOV      #364, $TMP20   ;MOVE FAILURE TO ABORT ERROR TO $TMP20.
7335 041224 000167 001022      JMP      MULTER         ;JUMP TO MULTIPLE ERROR HANDLER.
7336
7337 041230 016767 136340 140000 TRPV:  MOV      SR1, $TMP2      ;MOVE SR1 TO $TMP2 FOR TESTING.
7338 041236 016767 136334 140014      MOV      MMR2, $TMP13  ;TRANSFER ADDRESS OF INST. CAUSING TRAP TO $TMP13.
7339 041244 005067 136322      CLR      MMRO          ;TURN OFF MEMORY MANAGEMENT.
7340 041250 112767 000365 140014      MOVVB   #365, $TMP20   ;MOVE 365, THE MODE 0 ERROR, TO LOWER BYTE IN ERROR POINTER.
7341 041256 022767 040476 137774      CMP      #MODE1, $TMP13 ;SEE IF INSTRUCTION CAUSING TRAP IS BEFORE MODE 1 (MODE 0).
7342 041264 002402      BLT      1$              ;BRANCH AROUND MODE 0 ERROR JUMP IF NOT.
7343 041266 000167 000760      JMP      MULTER         ;JUMP TO ERROR NEST.
7344 041272 017767 137762 137766 1$: MOV      @ $TMP13, $TMP16 ;MOVE INSTRUCTION CAUSING TRAP TO $TMP16.
7345 041300 112767 000363 137764      MOVVB   #363, $TMP20   ;MOVE 363, SR1 WRONG ERROR, TO LOWER BYTE IN ERROR POINTER.
7346 041306 005067 137726      CLR      $TMP3         ;CLEAR CALCULATED LOCATION.
7347 041312 012767 041504 137724      MOV      #65$, $TMP5   ;MOVE NEXT CHECK ADDRESS TO $TMP5.
7348 041320 022767 040530 137732      CMP      #LABEL1, $TMP13 ;SEE IF TRAP IS BEFORE MODE 2 REG 1 CLR INST.
7349 041326 100053      BPL      61$           ;BRANCH TO SR1=0 TEST IF SO.
7350 041330 012767 000060 000004      MOV      #60, 2$+4     ;SET UP BIC DATA POSITION.
7351 041336 046727 137724 000060 2$: BIC      $TMP16, #60   ;TEST TO SEE IF MODE 6 OR 7 INSTRUCTION.
7352 041344 001444      BEQ      61$           ;BRANCH DIRECTLY TO BYTE TABLE TESTING IF SO.
7353      ;THIS NEXT ROUTINE DETERMINES WHICH REGISTER WAS IN THE INSTRUCTION, AND
7354      ;LOADS THE START AND END VALUES OF EITHER R1 OR R7 (PROGRAM COUNTER) INTO
7355      ;$TMP17 AND $TMP3 RESPECTIVELY. THEY ARE THEN SUBTRACTED TO FIND THE
7356      ;DIFFERENCE THAT ACTUALLY OCCURED. IF NO DIFFERENCE WAS FOUND, THE TEST
7357      ;FOR ZERO IN SR1 IS ACCOMPLISHED. IF A DIFFERENCE IS FOUND, THE DIFFERENCE
7358      ;IS SHIFTED LEFT 3 PLACES, THE TOP BYTE IS CLEARED, AND THE REGISTER
7359      ;OF THE INSTRUCTION IS ADDED. $TMP3 NOW CONTAINS WHAT SHOULD APPEAR
7360      ;IN SR1, ACCORDING TO WHAT ACTUALLY HAPPENED TO THE REGISTER.
7361 041346 042767 177770 137712 4$: BIC      #177770, $TMP16 ;BIT CLEAR THE INSTRUCTION, LEAVING THE REG EXPOSED.
7362 041354 026727 137706 000007      CMP      $TMP16, #7    ;COMPARE REGISTER TO DETERMINE IF IT IS REG 7.
7363 041362 001405      BEQ      5$              ;BRANCH TO THE REG 7 SETUP IF EQUAL TO REG 7.
7364 041364 010467 137700      MOV      R4, $TMP17    ;MOVE THE START VALUE TO $TMP17.
7365 041370 010167 137644      MOV      R1, $TMP3     ;MOVE THE END VALUE TO $TMP3.
7366 041374 000410      BR      6$              ;BRANCH TO CONTINUE.
7367 041376 016767 137656 137664 5$: MOV      $TMP13, $TMP17 ;MOVE THE START VALUE TO $TMP17.
7368 041404 062767 000002 137656      ADD      #2, $TMP17    ;ADD 2 TO START VALUE FOR NORMAL INCREMENTING.
7369 041412 011667 137622      MOV      (SP), $TMP3   ;MOVE THE END VALUE TO $TMP3.
7370 041416 166767 137646 137614 6$: SUB      $TMP17, $TMP3 ;FIND THE DIFFERENCE THAT OCCURED.
7371 041424 001414      BEQ      61$           ;BRANCH TO TEST FOR SR1=0 IF NO DIFFERENCE.
7372 041426 006367 137606      ASL      $TMP3         ;ARITHMETIC SHIFT LEFT $TMP3 3
7373 041432 006367 137602      ASL      $TMP3         ;PLACES TO PUT DIFFERENCE FOUND
7374 041436 006367 137576      ASL      $TMP3         ;IN BITS 3 THROUGH 7.

```

7375	041442	042767	177400	137570		BIC	#177400,\$TMP3	:BIT CLEAR UPPER BYTE OF \$TMP3.
7376	041450	066767	137612	137562		ADD	\$TMP16,\$TMP3	:ADD THE REGISTER THAT WAS CHANGED, AND
7377	041456	111567	137614		61\$:	MOVB	(R5),\$TMP22	:MOVE EXPECTED DATA TO \$TMP22.
7378	041462	126725	137550			CMPB	\$TMP2,(R5)+	:COMPARE SR1 WITH TABLE DATA.
7379	041466	001004				BNE	62\$:BRANCH TO ERROR JUMP IF WRONG.
7380	041470	005305				DEC	R5	:CORRECT R5 BEFORE NEXT COMPARE.
7381	041472	126725	137542			CMPB	\$TMP3,(R5)+	:COMPARE CALCULATED WITH TABLE DATA.
7382	041476	001402				BEQ	65\$:BRANCH AROUND ERROR JUMP IF OK.
7383	041500	000167	000452		62\$:	JMP	7\$:JUMP TO ERROR REPORT IF INCORRECT.
7384	041504	132767	000001	137561	65\$:	BITB	#1,\$TMP20+1	:TEST TO SEE IF BIT 8 IS SET.
7385	041512	001402				BEQ	66\$:BRANCH AROUND AC SKIP JUMP IF NOT.
7386	041514	000167	000614			JMP	RETURN	:JUMP TO RETURN - AC TESTS ARE TO BE SKIPPED.
7387	041520	112767	000366	137544	66\$:	MOVB	#366,\$TMP20	:MOVE 366, AC LOAD ERROR, TO ERROR POINTER.
7388	041526	010067	000724			MOV	R0,STORE+56	:STORE R0 FOR USE LATER IN THIS ROUTINE.
7389	041532	005067	137500			CLR	\$TMP2	:MOVE A '0' IN 'AC CHANGED' LOCATION.
7390	041536	012767	041602	137500		MOV	#101\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7391	041544	173467	000630			CMPF	STORE,ACO	:SEE IF ACO WAS CHANGED.
7392	041550	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7393	041552	001413				BEQ	101\$:BRANCH TO NEXT TEST IF OK.
7394	041554	174067	000666			STF	ACO,STORE+46	:STORE ACTUAL ACO FOR ERROR PRINTING.
7395	041560	012700	042400			MOV	#STORE,R0	:MOVE ADDRESS OF EXPECTED ACO TO R0.
7396						:THE NEXT TWO INSTRUCTIONS TRY TO RESTORE THE ACCUMULATOR AND CHECK THE ACCUMULATOR		
7397						:TO MAKE SURE IT WAS RESTORED PROPERLY FOR THE NEXT RUN THROUGH THIS TRAP HANDLER.		
7398						:IT IS *IMPORTANT* TO REALIZE THAT IF THE 'CMPF' FINDS A DIFFERENCE, THAT THE		
7399						:*FLOATING*POINT*STATUS* IS BEING CHANGED MISTAKENLY. AN ERROR IN THE MICROCODE		
7400						:HAS BEEN FOUND TO CAUSE THIS, SO CHECK THE REVISION OF THE ROM/PROM SET IN THE		
7401						:FPP YOU HAVE. IF YOU DO HAVE WHAT *SEEMS* TO BE THE LATEST REV, A NEW REV WILL		
7402						:BE COMMING OUT TO CORRECT THIS PROBLEM. THIS SAME 'LDF/CMPF' SET OF RESTORE/		
7403						:CHECK INSTRUCTIONS IS ACCOMPLISHED FOR EACH ACCUMULATOR CHECK. IT IS ALSO		
7404						:IMPORTANT TO NOTE THAT IF AN ACCUMULATOR FAILS TO RESTORE PROPERLY, SUBSEQUENT		
7405						:PASSES THROUGH THE TRAP HANDLER WILL SKIP THE ACCUMULATOR CHECKS DUE TO THE		
7406						:BIT TEST #400 ABOVE. FOR EXAMPLE, IF ACO FAILS TO LOAD PROPERLY, AC1 THROUGH		
7407						:AC3 WILL STILL BE CHECKED. AS SOON AS ANOTHER FPP INSTRUCTION TRAPS IN THE		
7408						:MAIN TEST, ALL *FURTHER* ACO-AC3 CHECKS WILL BE SKIPPED.		
7409	041564	172467	000610			LDF	STORE,ACO	:RESTORE ACO.
7410	041570	173467	000604			CMPF	STORE,ACO	:SEE IF IT WAS RESTORED PROPERLY.
7411	041574	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7412	041576	001567				BEQ	7\$:BRANCH TO ERROR CALL IF OK.
7413	041600	000476				BR	113\$:BRANCH TO ERROR SETUP ROUTINE.
7414	041602	012767	000001	137426	101\$:	MOV	#1,\$TMP2	:PUT A '1' IN 'AC CHANGED' LOCATION.
7415	041610	012767	041654	137426		MOV	#102\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7416	041616	173567	000566			CMPF	STORE+10,AC1	:SEE IF AC1 WAS CHANGED.
7417	041622	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7418	041624	001413				BEQ	102\$:BRANCH TO NEXT TEST IF OK.
7419	041626	174167	000614			STF	AC1,STORE+46	:STORE ACTUAL AC1 FOR ERROR PRINTING.
7420	041632	012700	042410			MOV	#STORE+10,R0	:MOVE ADDRESS OF EXPECTED AC1 TO R0.
7421	041636	172567	000546			LDF	STORE+10,AC1	:RESTORE AC1.
7422	041642	173567	000542			CMPF	STORE+10,AC1	:SEE IF IT WAS RESTORED PROPERLY.
7423	041646	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7424	041650	001542				BEQ	7\$:BRANCH TO ERROR CALL IF OK.
7425	041652	000451				BR	113\$:BRANCH TO ERROR SETUP ROUTINE.
7426	041654	012767	000002	137354	102\$:	MOV	#2,\$TMP2	:PUT A '2' IN 'AC CHANGED' LOCATION.
7427	041662	012767	041726	137354		MOV	#103\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7428	041670	173667	000524			CMPF	STORE+20,AC2	:SEE IF AC2 WAS CHANGED.
7429	041674	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7430	041676	001413				BEQ	103\$:BRANCH TO NEXT TEST IF OK.
7431	041700	174267	000542			STF	AC2,STORE+46	:STORE ACTUAL AC2 FOR ERROR PRINTING.

7432	041704	012700	042420			MOV	#STORE+20,R0	:MOVE ADDRESS OF EXPECTED AC2 TO R0.
7433	041710	172667	000504			LDF	STORE+20,AC2	:RESTORE AC2.
7434	041714	173667	000500			CMPF	STORE+20,AC2	:SEE IF IT WAS RESTORED PROPERLY.
7435	041720	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7436	041722	001515				BEQ	7\$:BRANCH TO ERROR CALL IF OK.
7437	041724	000424				BR	113\$:BRANCH TO ERROR SETUP ROUTINE.
7438	041726	012767	000003	137302	103\$:	MOV	#3,\$TMP2	:PUT A '3' IN 'AC CHANGED' LOCATION.
7439	041734	012767	042006	137302		MOV	#100\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7440	041742	173767	000462			CMPF	STORE+30,AC3	:SEE IF AC3 WAS CHANGED.
7441	041746	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7442	041750	001416				BEQ	100\$:BRANCH TO NEXT TEST IF OK.
7443	041752	174367	000470			STF	AC3,STORE+46	:STORE ACTUAL AC3 FOR ERROR PRINTING.
7444	041756	012700	042430			MOV	#STORE+30,R0	:MOVE ADDRESS OF EXPECTED AC3 TO R0.
7445	041762	172767	000442			LDF	STORE+30,AC3	:RESTORE AC3.
7446	041766	173767	000436			CMPF	STORE+30,AC3	:SEE IF IT WAS RESTORED PROPERLY.
7447	041772	170000				CFCC		:COPY FPP CONDITION CODES TO CPU CODES.
7448	041774	001470				BEQ	7\$:BRANCH TO ERROR CALL IF OK.
7449	041776	012767	000770	137266	113\$:	MOV	#770,\$TMP20	:MOVE 370 FOR AC LOAD FAILURE, & SET BIT 8 OF ERROR POINTER.
7450	042004	000464				BR	7\$:BRANCH TO ERROR CALL.
7451	042006	005067	137224		100\$:	CLR	\$TMP2	:CLEAR 'REGISTER CHANGED' LOCATION.
7452	042012	112767	000367	137252		MOVB	#367,\$TMP20	:MOVE 367, GENERAL REGISTER CHANGED ERROR, TO POINTER.
7453	042020	012767	042054	137216		MOV	#120\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7454	042026	026700	000406			CMP	STORE+40,R0	:SEE IF R0 WAS CHANGED.
7455	042032	001410				BEQ	120\$:BRANCH TO NEXT TEST IF OK.
7456	042034	010067	137206			MOV	R0,\$TMP6	:MOVE ACTUAL R0 TO LOCATION FOR ERROR PRINTING.
7457	042040	016767	000374	137172		MOV	STORE+40,\$TMP3	:MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7458	042046	016700	000366			MOV	STORE+40,R0	:RESTORE R0.
7459	042052	000441				BR	7\$:BRANCH TO ERROR CALL.
7460	042054	012767	000002	137154	120\$:	MOV	#2,\$TMP2	:PUT A '2' IN 'REGISTER CHANGED' LOCATION.
7461	042062	012767	042116	137154		MOV	#130\$,\$TMP5	:MOVE RETURN TO \$TMP5.
7462	042070	026702	000346			CMP	STORE+42,R2	:SEE IF R2 WAS CHANGED.
7463	042074	001410				BEQ	130\$:BRANCH TO NEXT TEST IF OK.
7464	042076	010267	137144			MOV	R2,\$TMP6	:MOVE ACTUAL R2 TO LOCATION FOR ERROR PRINTING.
7465	042102	016767	000334	137130		MOV	STORE+42,\$TMP3	:MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7466	042110	016702	000326			MOV	STORE+42,R2	:RESTORE R2.
7467	042114	000420				BR	7\$:BRANCH TO ERROR CALL.
7468	042116	012767	000003	137134	130\$:	MOV	#3,\$TMP13	:PUT A '3' IN 'REGISTER CHANGED' LOCATION.
7469	042124	012767	042334	137112		MOV	#RETURN,\$TMP5	:MOVE RETURN TO \$TMP5.
7470	042132	026703	000306			CMP	STORE+44,R3	:SEE IF R3 WAS CHANGED.
7471	042136	001476				BEQ	RETURN	:BRANCH TO RETURN IF OK.
7472	042140	010367	137102			MOV	R3,\$TMP6	:MOVE ACTUAL R3 TO LOCATION FOR ERROR PRINTING.
7473	042144	016767	000274	137066		MOV	STORE+44,\$TMP3	:MOVE EXPECTED TO LOCATION FOR ERROR PRINTING.
7474	042152	016703	000266			MOV	STORE+44,R3	:RESTORE R3.
7475	042156	122767	000370	137106	7\$:	CMPB	#370,\$TMP20	:TEST TO SEE IF AC INFO NEEDS TO BE STORED.
7476	042164	001404				BEQ	71\$:BRANCH TO INFO STORE ROUTINE IF SO.
7477	042166	122767	000366	137076		CMPB	#366,\$TMP20	:TEST TO SEE IF AC INFO NEEDS TO BE STORED.
7478	042174	001026				BNE	MULTER	:SKIP AC INFO ROUTINE IF NOT.
7479	042176	012067	137036		71\$:	MOV	(R0)+,\$TMP3	:MOVE 1ST WORD OF ACTUAL AC DATA TO \$TMP3.
7480	042202	012067	137034			MOV	(R0)+,\$TMP4	:MOVE 2ND WORD OF ACTUAL AC DATA TO \$TMP4.
7481	042206	012067	137034			MOV	(R0)+,\$TMP6	:MOVE 3RD WORD OF ACTUAL AC DATA TO \$TMP6.
7482	042212	012067	137032			MOV	(R0)+,\$TMP7	:MOVE 4TH WORD OF ACTUAL AC DATA TO \$TMP7.
7483	042216	016700	000234			MOV	STORE+56,R0	:RESTORE R0 TO WHAT IT HAD AT BEGINNING OF TRAP.
7484	042222	016767	000220	137022		MOV	STORE+46,\$TMP10	:MOVE 1ST WORD OF EXPECTED AC DATA TO \$TMP10.
7485	042230	016767	000214	137016		MOV	STORE+50,\$TMP11	:MOVE 2ND WORD OF EXPECTED AC DATA TO \$TMP11.
7486	042236	016767	000210	137012		MOV	STORE+52,\$TMP12	:MOVE 3RD WORD OF EXPECTED AC DATA TO \$TMP12.
7487	042244	016767	000204	137022		MOV	STORE+54,\$TMP21	:MOVE 4TH WORD OF EXPECTED AC DATA TO \$TMP21.
7488	042252	012667	137010		MULTER:	MOV	(SP)+,\$TMP16	:SAVE 1ST CONTENTS OF STACK AND POP IT ONCE.

```

7489 042256 012667 137006      MOV      (SP)+,$TMP17      ;SAVE 2ND CONTENTS OF STACK AND POP IT AGAIN.
7490 042262 142767 000377 000006  BICB    #377,74$         ;CLEAR OUT LAST ERROR OFFSET FROM ERROR INSTRUCTION.
7491 042270 153767 001272 000000  BISB    @#$TMP20,74$     ;PUT ERROR NUMBER TO BE ACCOMPLISHED IN ERROR INSTRUCTION.
7492                                     ;THIS ERROR IS DEFINED BY THE CONTENTS OF THE LOWER BYTE OF LOCATION [1272]
7493 042276 104000      74$:   ERROR      +0
7494 042300 016746 136764      MOV      $TMP17,-(SP)     ;PUSH 2ND CONTENTS BACK ON THE STACK.
7495 042304 016746 136756      MOV      $TMP16,-(SP)     ;PUSH 1ST CONTENTS BACK ON THE STACK.
7496 042310 022767 000364 136754  CMP      #364,$TMP20      ;SEE IF RETURN ROUTINE IS TO BE SKIPPED.
7497 042316 001417      BEQ      RTI              ;BRANCH TO RTI IF SO.
7498 042320 022767 000365 136744  CMP      #365,$TMP20      ;SEE IF RETURN ROUTINE IS TO BE SKIPPED.
7499 042326 001413      BEQ      RTI              ;BRANCH TO RTI IF SO.
7500 042330 000177 136710      JMP      @$TMP5           ;JUMP TO CONTINUE CHECKING.
7501 042334 022776 000004 000000  RETURN:  CMP      #4,@0(SP) ;SEE IF INSTRUCTION IS THE IOT.
7502 042342 001403      BEQ      9$              ;BRANCH IF THE IOT HAS BEEN FOUND.
7503 042344 062716 000002      ADD      #2,(SP)         ;CORRECT PC RETURN.
7504 042350 000771      BR      RETURN           ;BRANCH BACK FOR ANOTHER TRY.
7505 042352 062716 000002  9$:     ADD      #2,(SP)         ;CORRECT PC RETURN TO POINT AFTER IOT FOUND.
7506 042356 005267 135210  RTI:    INC      MMR0      ;TURN ON MEMORY MANAGEMENT, AND
7507 042362 000002      RTI                    ;RETURN FROM INTERRUPT.
  
```

```

7508
7509 042364  NODAT: .BLKW 6          ;LOCATION IN NON-RES. D-SPACE USED TO FORCE A TRAP.
7510                                     ;THE 'STORE' LOCATION BELOW IS PARTITIONED TO RESERVE *4* WORDS FOR EACH FP
7511                                     ;ACCUMULATOR, EVEN THOUGH ONLY 2 ARE REQUIRED FOR STORING A FLOATING NUMBER.
7512                                     ;THIS IS BECAUSE *IF* THE FPS IS CHANGED BY A PROBLEM IN THE FPP, SO THAT A
7513                                     ;*DOUBLE* IS STORED, *4* WORDS RESERVED WILL GUARANTEE THAT THE NEXT DATA BLOCK
7514                                     ;WILL NOT BE DISTURBED. PARTITIONING IS AS FOLLOWS:
  
```

WORD(S)	USE
1 - 4	STORE AC0
5 -10	STORE AC1
11-14	STORE AC2
15-20	STORE AC3
21	STORE R0
22	STORE R2
23	STORE R3
24-27	STORE ACTUAL AC
30	STORE ACTUAL R0 SO R0 CAN BE USED IN AC ERROR CALLS

```

7525
7526 042400  STORE: .BLKW 30        ;STORAGE LOCATIONS FOR THE FLOATING ACCUMULATORS & DATA.
7527                                     ;THE FOLLOWING BYTE TABLE WILL BE USED TO CHECK THE VALUES OF SR1 AND THE CALCULATED
7528                                     ;VALUES. SR1 MAY TRACK THE ACTIVE REGISTER PROPERLY, BUT IF THE *VALUE* OF THE
7529                                     ;INCREMENT/DECREMENT IS WRONG, AND ERROR STILL EXISTS.
  
```

```

7530 042460      000      000      000  BYTABL: .BYTE 0,0,0,0,0,0,41,21,41,41,0,0,0,27,0,27,0,27,0,27,0,341,361,341
      042463      000      000      000
      042466      041      021      041
      042471      041      000      000
      042474      000      027      000
      042477      027      000      027
      042502      000      027      000
      042505      341      361      341
7531 042510      341      000      361      .BYTE 341,0,361,361,361,361,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
      042513      361      361      361
      042516      000      000      000
      042521      000      000      000
      042524      000      000      000
      042527      000      000      000
      042532      000      000      000
      042535      000      000      000
  
```

```

042540      000      000
7532
7533 042542 005067 135024      ENDTES: CLR      MMRO      ;TURN OFF MEMORY MANAGEMENT.
7534 042546 016767 136510 135474      MOV      $TMP14,MMVECT ;RESTORE MMVECT CONTENTS.
7535 042554 016767 136504 135236      MOV      $TMP15,IOTRAP ;RESTORE IOTRAP CONTENTS..
7536 042562
      042562 104412      DIDONE:      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?).
  
```

```

7537
7538 042564      TST100:
7539
7540
7541
7542
  
```

```

.SBTTL END OF PASS ROUTINE
;*****
;*INCREMENT THE PASS NUMBER ($PASS)
;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
;*IF SW12=1 INHIBIT TRACE TRAP
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO LOOP
$EOP:
  
```

```

042564      000004      SCOPE
042564 005067 136310      CLR      $TSTNM      ;;ZERO THE TEST NUMBER
042572 005067 136504      CLR      $TIMES      ;;ZERO THE NUMBER OF ITERATIONS
042576 005267 136522      INC      $PASS      ;;INCREMENT THE PASS NUMBER
042602 042767 100000 136514      BIC      #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
042610 005327      DEC      (PC)+      ;;LOOP?
042612 000001      $EOPCT: .WORD      1
042614 003074      BGT      $DOAGN      ;;YES
042616 012737      MOV      (PC)+,@(PC)+ ;;RESTORE COUNTER
042620 000001      $ENDCT: .WORD      1
042622 042612      $EOPCT
042624 104401 042632      TYPE      ,65$      ;;TYPE ASCIZ STRING
042630 000407      BR      64$      ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <12><15>/END PASS #/
64$:
042650      016746 136450      MOV      $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
042650      104403      TYPOS      ;;TYPE PASS NUMBER IN OCTAL
042656      006      .BYTE      6      ;;GO TYPE--OCTAL ASCII
042657      000      .BYTE      0      ;;TYPE 6 DIGITS
042660 104401 042666      TYPE      ,67$      ;;SUPPRESS LEADING ZEROS
042664 000421      BR      66$      ;;TYPE ASCIZ STRING
;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
042730      016746 136156      MOV      $ERTTL,-(SP) ;;SAVE $ERTTL FOR TYPEOUT
042730      TYPOS      ;;TOTAL NUMBER OF ERRORS IN OCTAL
042734 104403      .BYTE      6      ;;GO TYPE--OCTAL ASCII
042736      006      .BYTE      0      ;;TYPE 6 DIGITS
042737      000      .BYTE      0      ;;SUPPRESS LEADING ZEROS
042740 104401 001313      TYPE      ,$CRLF      ;;TYPE CARRIAGE RETURN, LINE FEED
042744 005067 136142      CLR      $ERTTL      ;;CLEAR ERROR TOTAL
042750 013700 000042      $GET42: MOV      @#42,R0      ;;GET MONITOR ADDRESS
042754 001414      BEQ      $DOAGN      ;;BRANCH IF NO MONITOR
  
```



```

042756 005046          CLR      -(SP)          ;;INSURE THE 'T' BIT IS CLEAR
042760 012746 042766  MOV      #$CLR.T,-(SP)  ;;SETUP FOR AN RTI OR RTT
042764 000426          BR       $RTRN          ;;GO DO AN RTI OR RTT TO LOAD THE PSW
                                ;;WITH A CLEARED 'T' BIT

042766          $CLR.T:
042766 013700 000042  MOV      @#42,R0          ;;INSURE R0 CONTAINS THE MONITORS
042772 001405          BEQ      $DOAGN          ;;RETURN ADDRESS
042774 000005          RESET          ;;CLEAR THE WORLD
042776 004710          $ENDAD: JSR     PC,(R0)  ;;GO TO MONITOR
043000 000240          NOP          ;;SAVE ROOM
043002 000240          NOP          ;;FOR
043004 000240          NOP          ;;ACT11
043006          $DOAGN:
043006 104400          TRAP          ;;PUSH OLD PSW AND PC ON STACK
043010 042716 000020  BIC      #20,(SP)        ;;CLEAR THE 'T' BIT
043014 032777 010000 136116 BIT      #BIT12,@SWR     ;;RUN WITH TRACE TRAP?
043022 001005          BNE      1$              ;;BR IF NO
043024 005167 000020  COM      $TBIT          ;;IS IT TIME FOR TRACE TRAP
043030 100402          BMI      1$              ;;BR IF NO
043032 052716 000020  BIS      #20,(SP)        ;;SET TRACE TRAP
043036 012746 043044  1$:      MOV      #$LOOP,-(SP)  ;;JUMP TO START OF TEST
043042 000002  $RTRN: RTI          ;;RETURN--THIS IS CHANGED TO
                                ;;AN 'RTT' IF 'RTT' IS A LEGAL
                                ;;INSTRUCTION

043044          $LOOP:
043044 000137          JMP      @(PC)+          ;;RETURN
043046 006570          $RTNAD: .WORD   LOOP
043050 000000          $TBIT:  .WORD   0          ;;'T' BIT STATE INDICATOR
043052 377 377 000  $ENULL: .BYTE  -1,-1,0  ;;NULL CHARACTER STRING
                                .EVEN
    
```

7543
7544

```

.SBTTL SCOPE HANDLER ROUTINE
;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;*SW14=1      LOOP ON TEST
;*SW11=1      INHIBIT ITERATIONS
;*SW09=1      LOOP ON ERROR
;*SW08=1      LOOP ON TEST IN SWR<7:0>
;*CALL
;*          SCOPE          ;;SCOPE=IOT
$SCOPE:
043056 104406          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
043060 032777 040000 136052 1$:      BIT      #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
043066 001114          BNE      $OVER          ;;YES IF SW14=1
;#####START OF CODE FOR THE XOR TESTER#####
043070 000416          $XTSTR: BR      6$          ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
                                ;;THIS INSTRUCTION TO A 'NOP' (NOP=240)
043072 013746 000004          MOV      @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
043076 012737 043116 000004  MOV      #5$,@#ERRVEC     ;;SET FOR TIMEOUT
043104 005737 177060          TST      @#177060        ;;TIME OUT ON XOR?
043110 012637 000004          MOV      (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
043114 000463          BR       $SVLAD        ;;GO TO THE NEXT TEST
043116 022626          5$:      CMP      (SP)+,(SP)+  ;;CLEAR THE STACK AFTER A TIME OUT
043120 012637 000004          MOV      (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
    
```

```

043124 000423          BR      7$          ;;LOOP ON THE PRESENT TEST
043126 032777 000400 136004 6$:;#####END OF CODE FOR THE XOR TESTER#####
043134 001404          BEQ      2$          ;;LOOP ON SPEC. TEST?
043136 127767 135776 135736          CMPB   @SWR,$STNM   ;;BR IF NO
043144 001465          BEQ      $OVER       ;;ON THE RIGHT TEST? SWR<7:0>
043146 105767 135731          TSTB   $ERFLG      ;;BR IF YES
043152 001421          BEQ      3$          ;;HAS AN ERROR OCCURRED?
043154 126767 135735 135721          CMPB   $ERMAX,$ERFLG ;;BR IF NO
043162 101015          BHI     3$          ;;MAX. ERRORS FOR THIS TEST OCCURRED?
043164 032777 001000 135746          BIT    #BIT09,@SWR  ;;BR IF NO
043172 001404          BEQ      4$          ;;LOOP ON ERROR?
043174 016767 135710 135704 7$:    MOV    $LPERR,$LPADR ;;BR IF NO
043202 000446          BR      $OVER       ;;SET LOOP ADDRESS TO LAST SCOPE
043204 105067 135673          CLRB   $ERFLG      ;;ZERO THE ERROR FLAG
043210 005067 136066          CLR    $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
043214 000415          BR      1$          ;;ESCAPE TO THE NEXT TEST
043216 032777 004000 135714 3$:    BIT    #BIT11,@SWR  ;;INHIBIT ITERATIONS?
043224 001011          BNE     1$          ;;BR IF YES
043226 005767 136072          TST    $PASS       ;;IF FIRST PASS OF PROGRAM
043232 001406          BEQ      1$          ;;INHIBIT ITERATIONS
043234 005267 135644          INC    $ICNT       ;;INCREMENT ITERATION COUNT
043240 026767 136036 135636          CMP    $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
043246 002024          BGE     $OVER       ;;BR IF MORE ITERATION REQUIRED
043250 012767 000001 135626 1$:    MOV    #1,$ICNT     ;;REINITIALIZE THE ITERATION COUNTER
043256 016767 000052 136016          MOV    $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
043264 105267 135612          $SVLAD: INCB   $STNM   ;;COUNT TEST NUMBERS
043270 116767 135606 136024          MOVB   $STNM,$STNM  ;;SET TEST NUMBER IN APT MAILBOX
043276 011667 135604          MOV    (SP),$LPADR  ;;SAVE SCOPE LOOP ADDRESS
043302 011667 135602          MOV    (SP),$LPERR  ;;SAVE ERROR LOOP ADDRESS
043306 005067 135772          CLR    $ESCAPE     ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
043312 112767 000001 135575          MOVB   #1,$ERMAX   ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
043320 016777 135556 135614 $OVER: MOV    $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
043326 016716 135554          MOV    $LPADR,(SP) ;;FUDGE RETURN ADDRESS
043332 000002          RTI             ;;FIXES PS
043334 000001          $MXCNT: 1       ;;MAX. NUMBER OF ITERATIONS

```

7545
7546

```

.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

```

```

043336 104406          $ERROR: CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
043340 105267 135537          7$:    INCB   $ERFLG      ;;SET THE ERROR FLAG
043344 001775          BEQ      7$          ;;DON'T LET THE FLAG GO TO ZERO
043346 016777 135530 135566          MOV    $STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
043354 032777 002000 135556          BIT    #BIT10,@SWR  ;;BELL ON ERROR?
043362 001402          BEQ      1$          ;;NO - SKIP
043364 104401 001306          TYPE   , $BELL     ;;RING BELL

```

```

043370 005267 135516      1$:  INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
043374 011667 135516      MOV      (SP), $ERRPC  ;;GET ADDRESS OF ERROR INSTRUCTION
043400 162767 000002 135510  SUB      #2, $ERRPC
043406 117767 135504 135500  MOVB    @ $ERRPC, $ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
043414 032777 020000 135516  BIT      #BIT13, @SWR   ;;SKIP TYPEOUT IF SET
043422 001004      BNE      20$          ;;SKIP TYPEOUTS
043424 004767 002174      JSR      PC, ERTYPE    ;;GO TO USER ERROR ROUTINE
043430 104401 001313      TYPE      , $CRLF
043434      20$:
043434 122767 000001 135674  CMPB    #APTENV, $ENV   ;;RUNNING IN APT MODE
043442 001007      BNE      2$          ;;NO, SKIP APT ERROR REPORT
043444 116767 135444 000004  MOVB    $ITEMB, 21$    ;;SET ITEM NUMBER AS ERROR NUMBER
043452 004767 001010      JSR      PC, $ATY4     ;;REPORT FATAL ERROR TO APT
043456      000      21$:  .BYTE    0
043457      000      .BYTE    0
043460 000777      22$:  BR      22$          ;;APT ERROR LOOP
043462 005777 135452      2$:  TST     @SWR          ;;HALT ON ERROR
043466 100002      BPL     3$          ;;SKIP IF CONTINUE
043470 000000      HALT                    ;;HALT ON ERROR!
043472 104406      CKSWR                    ;;TEST FOR CHANGE IN SOFT-SWR
043474 032777 001000 135436  3$:  BIT      #BIT09, @SWR  ;;LOOP ON ERROR SWITCH SET?
043502 001402      BEQ     4$          ;;BR IF NO
043504 016716 135400      MOV     $LPERR, (SP)   ;;FUDGE RETURN FOR LOOPING
043510 005767 135570      4$:  TST     $ESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS
043514 001402      BEQ     5$          ;;BR IF NONE
043516 016716 135562      MOV     $ESCAPE, (SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
043522      5$:
043522 022737 042776 000042  5$:  CMP     # $ENDAD, @#42 ;;ACT-11 AUTO-ACCEPT?
043530 001001      BNE     6$          ;;BRANCH IF NO
043532 000000      HALT                    ;;YES
043534      6$:
043534 032777 001000 135376  6$:  BIT      #BIT09, @SWR
043542 001013      BNE     ERM10
043544 011637 001162      MOV     (SP), @#$REGO  ;SEE IF ERROR #377
043550 062737 177776 001162  ADD     #-2, @#$REGO
043556 122777 000377 135376  CMPB    #377, @ $REGO
043564 001002      BNE     ERM10
043566 062716 000002      ADD     #2, (SP)
043572 000002      ERM10: RTI
    
```

7547
7548

```

.SBTTL SAVE AND RESTORE R0-R5 ROUTINES
;*****
;SAVE R0-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
    
```

043574
043574 010046

```

043576 010146      MOV     R1,-(SP)      ;;PUSH R1 ON STACK
043600 010246      MOV     R2,-(SP)      ;;PUSH R2 ON STACK
043602 010346      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
043604 010446      MOV     R4,-(SP)      ;;PUSH R4 ON STACK
043606 010546      MOV     R5,-(SP)      ;;PUSH R5 ON STACK
043610 016646 000022  MOV     22(SP),-(SP)  ;;SAVE PS OF MAIN FLOW
043614 016646 000022  MOV     22(SP),-(SP)  ;;SAVE PC OF MAIN FLOW
043620 016646 000022  MOV     22(SP),-(SP)  ;;SAVE PS OF CALL
043624 016646 000022  MOV     22(SP),-(SP)  ;;SAVE PC OF CALL
043630 000002      RTI

```

```

;*RESTORE R0-R5
;*CALL:
;*
;* RESREG
$RESREG:

```

```

043632 012666 000022  MOV     (SP)+,22(SP)  ;;RESTORE PC OF CALL
043636 012666 000022  MOV     (SP)+,22(SP)  ;;RESTORE PS OF CALL
043642 012666 000022  MOV     (SP)+,22(SP)  ;;RESTORE PC OF MAIN FLOW
043646 012666 000022  MOV     (SP)+,22(SP)  ;;RESTORE PS OF MAIN FLOW
043652 012605      MOV     (SP)+,R5      ;;POP STACK INTO R5
043654 012604      MOV     (SP)+,R4      ;;POP STACK INTO R4
043656 012603      MOV     (SP)+,R3      ;;POP STACK INTO R3
043660 012602      MOV     (SP)+,R2      ;;POP STACK INTO R2
043662 012601      MOV     (SP)+,R1      ;;POP STACK INTO R1
043664 012600      MOV     (SP)+,R0      ;;POP STACK INTO R0
043666 000002      RTI

```

7549
7550

.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

```

```

043670 105767 135263 $TYPE: TSTB $TPFLG ;;IS THERE A TERMINAL?
043674 100002      BPL 1$ ;;BR IF YES
043676 000000      HALT ;;HALT HERE IF NO TERMINAL
043700 000430      BR 3$ ;;LEAVE
043702 010046      1$: MOV R0,-(SP) ;;SAVE R0
043704 017600 000002  MOV @2(SP),R0 ;;GET ADDRESS OF ASCIZ STRING
043710 122767 000001 135420  CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
043716 001011      BNE 62$ ;;NO,GO CHECK FOR APT CONSOLE
043720 132767 000100 135411  BITB #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
043726 001405      BEQ 62$ ;;NO,GO CHECK FOR CONSOLE
043730 010067 000004      MOV R0,61$ ;;SETUP MESSAGE ADDRESS FOR APT
043734 004767 000516      JSR PC,$ATY3 ;;SPOOL MESSAGE TO APT
043740 000000      61$: .WORD 0 ;;MESSAGE ADDRESS
043742 132767 000040 135367 62$: BITB #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
043750 001003      BNE 60$ ;;YES,SKIP TYPE OUT
043752 112046      2$: MOVB (R0)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK

```

```

043754 001005          BNE      4$          ;;BR IF IT ISN'T THE TERMINATOR
043756 005726          TST      (SP)+        ;;IF TERMINATOR POP IT OFF THE STACK
043760 012600          60$:    MOV      (SP)+,R0    ;;RESTORE R0
043762 062716 000002  3$:      ADD      #2,(SP)    ;;ADJUST RETURN PC
043766 000002          RTI                    ;;RETURN
043770 122716 000011  4$:      CMPB     #HT,(SP)    ;;BRANCH IF <HT>
043774 001430          BEQ      8$
043776 122716 000200  CMPB     #CRLF,(SP)    ;;BRANCH IF NOT <CRLF>
044002 001006          BNE      5$
044004 005726          TST      (SP)+        ;;POP <CR><LF> EQUIV
044006 104401          TYPE                    ;;TYPE A CR AND LF
044010 001313          $CRLF
044012 105067 000200  CLRB     $CHARCNT    ;;CLEAR CHARACTER COUNT
044016 000755          BR       2$          ;;GET NEXT CHARACTER
044020 004767 000056  5$:      JSR      PC,$TYPEC    ;;GO TYPE THIS CHARACTER
044024 126726 135126  6$:      CMPB     $FILLC,(SP)+  ;;IS IT TIME FOR FILLER CHARS.?
044030 001350          BNE      2$          ;;IF NO GO GET NEXT CHAR.
044032 016746 135116  MOV      $NULL,-(SP)  ;;GET # OF FILLER CHARS. NEEDED
                                ;;AND THE NULL CHAR.
044036 105366 000001  7$:      DECB     1(SP)    ;;DOES A NULL NEED TO BE TYPED?
044042 002770          BLT      6$          ;;BR IF NO--GO POP THE NULL OFF OF STACK
044044 004767 000032  JSR      PC,$TYPEC    ;;GO TYPE A NULL
044050 105367 000142  DECB     $CHARCNT    ;;DO NOT COUNT AS A COUNT
044054 000770          BR       7$          ;;LOOP
                                :HORIZONTAL TAB PROCESSOR
044056 112716 000040  8$:      MOVVB   #' ,(SP)    ;;REPLACE TAB WITH SPACE
044062 004767 000014  9$:      JSR      PC,$TYPEC    ;;TYPE A SPACE
044066 132767 000007 000122  BITB     #7,$CHARCNT  ;;BRANCH IF NOT AT
044074 001372          BNE      9$          ;;TAB STOP
044076 005726          TST      (SP)+        ;;POP SPACE OFF STACK
044100 000724          BR       2$          ;;GET NEXT CHARACTER
044102 105777 135042  $TYPEC:  TSTB     @STPS    ;;WAIT UNTIL PRINTER IS READY
044106 100375          BPL     $TYPEC
044110 116677 000002 135034  MOVVB   2(SP),@STPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
044116 105777 135022  TSTB     @STKS        ;;SEE IF KEYBOARD IS TALKING.
044122 100021          BPL     2$          ;;BRANCH IF IT ISN'T.
044124 017746 135016  MOV      @STKB,-(SP)  ;;PUSH CHARACTER ONTO STACK.
044130 042716 177600  BIC     #177600,(SP) ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044134 022726 000023  CMP      #23,(SP)+   ;;SEE IF THIS IS A ^S.
044140 001012          BNE      2$          ;;BRANCH TO CONTINUE IF IT ISN'T.
044142 105777 134776  3$:      TSTB     @STKS        ;;WAIT FOR ANOTHER INPUT.
044146 100375          BPL     3$          ;;BRANCH BACK IF NOT READY.
044150 017746 134772  MOV      @STKB,-(SP)  ;;PUSH NEXT CHARACTER ON STACK.
044154 042716 177600  BIC     #177600,(SP) ;;BIT CLEAR TOP BYTE AND PARITY BIT.
044160 022726 000021  CMP      #21,(SP)+   ;;SEE IF THIS IS A ^Q.
044164 001366          BNE      3$          ;;BRANCH BACK FOR MORE WAIT IF NOT.
044166 122766 000015 000002  2$:      CMPB     #CR,2(SP)    ;;IS CHARACTER A CARRIAGE RETURN?
044174 001003          BNE      1$          ;;BRANCH IF NO
044176 105067 000014  CLRB     $CHARCNT    ;;YES--CLEAR CHARACTER COUNT
044202 000406          BR       $TYPEX     ;;EXIT
044204 122766 000012 000002  1$:      CMPB     #LF,2(SP)    ;;IS CHARACTER A LINE FEED?
044212 001402          BEQ      $TYPEX     ;;BRANCH IF YES
044214 105227          INCB     (PC)+     ;;COUNT THE CHARACTER
044216 000000          $CHARCNT: .WORD  0  ;;CHARACTER COUNT STORAGE
044220 000207          $TYPEX:  RTS      PC
  
```

7551
7552

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
044222 017646 000000      $TYPOS: MOV      @(SP),-(SP)      ;;PICKUP THE MODE
044226 116667 000001 000211  MOVB     1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
044234 112667 000207      MOVB     (SP)+, $OMODE+1    ;;NUMBER OF DIGITS TO TYPE
044240 062716 000002      ADD      #2,(SP)          ;;ADJUST RETURN ADDRESS
044244 000406      BR      $TYPON
044246 112767 000001 000171 $TYPOC: MOVB     #1, $OFILL      ;;SET THE ZERO FILL SWITCH
044254 112767 000006 000165  MOVB     #6, $OMODE+1    ;;SET FOR SIX(6) DIGITS
044262 112767 000005 000154 $TYPON: MOVB     #5, $OCNT      ;;SET THE ITERATION COUNT
044270 010346      MOV      R3,-(SP)        ;;SAVE R3
044272 010446      MOV      R4,-(SP)        ;;SAVE R4
044274 010546      MOV      R5,-(SP)        ;;SAVE R5
044276 116704 000145      MOVB     $OMODE+1,R4    ;;GET THE NUMBER OF DIGITS TO TYPE
044302 005404      NEG      R4
044304 062704 000006      ADD      #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
044310 110467 000132      MOVB     R4, $OMODE      ;;SAVE IT FOR USE
044314 116704 000125      MOVB     $OFILL,R4      ;;GET THE ZERO FILL SWITCH
044320 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
044324 005003      CLR      R3            ;;CLEAR THE OUTPUT WORD
044326 006105      1$:     ROL      R5      ;;ROTATE MSB INTO 'C'
044330 000404      BR      3$            ;;GO DO MSB
044332 006105      2$:     ROL      R5      ;;FORM THIS DIGIT
044334 006105      ROL      R5
044336 006105      ROL      R5
044340 010503      MOV      R5,R3
044342 006103      3$:     ROL      R3      ;;GET LSB OF THIS DIGIT
044344 105367 000076      DECB     $OMODE        ;;TYPE THIS DIGIT?
044350 100016      BPL      7$            ;;BR IF NO
044352 042703 177770      BIC      #177770,R3    ;;GET RID OF JUNK
044356 001002      BNE      4$            ;;TEST FOR 0
044360 005704      TST      R4            ;;SUPPRESS THIS 0?
044362 001403      BEQ     5$            ;;BR IF YES
044364 005204      4$:     INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
044366 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
044372 052703 000040      5$:     BIS      #' ,R3  ;;MAKE ASCII IF NOT ALREADY
044376 110367 000040      MOVB     R3,8$        ;;SAVE FOR TYPING
    
```

044402	104401	044442			TYPE	.8\$::GO TYPE THIS DIGIT
044406	105367	000032	7\$:		DECB	\$OCNT	::COUNT BY 1
044412	003347				BGT	2\$::BR IF MORE TO DO
044414	002402				BLT	6\$::BR IF DONE
044416	005204				INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
044420	000744				BR	2\$::GO DO THE LAST DIGIT
044422	012605		6\$:		MOV	(SP)+,R5	::RESTORE R5
044424	012604				MOV	(SP)+,R4	::RESTORE R4
044426	012603				MOV	(SP)+,R3	::RESTORE R3
044430	016666	000002	000004		MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
044436	012616				MOV	(SP)+,(SP)	
044440	000002				RTI		::RETURN
044442	000			8\$:	.BYTE	0	::STORAGE FOR ASCII DIGIT
044443	000				.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
044444	000			\$OCNT:	.BYTE	0	::OCTAL DIGIT COUNTER
044445	000			\$OFILL:	.BYTE	0	::ZERO FILL SWITCH
044446	000000			\$OMODE:	.WORD	0	::NUMBER OF DIGITS TO TYPE

7553
7554

.SBTTL APT COMMUNICATIONS ROUTINE

```

*****
044450 112767 000001 000236 $ATY1: MOVB #1,$FFLG ::TO REPORT FATAL ERROR
044456 112767 000001 000226 $ATY3: MOVB #1,$MFLG ::TO TYPE A MESSAGE
044464 000403 BR $ATYC
044466 112767 000001 000220 $ATY4: MOVB #1,$FFLG ::TO ONLY REPORT FATAL ERROR
044474 $ATYC:
044474 010046 MOV R0,-(SP) ::PUSH R0 ON STACK
044476 010146 MOV R1,-(SP) ::PUSH R1 ON STACK
044500 105767 000206 TSTB $MFLG ::SHOULD TYPE A MESSAGE?
044504 001450 BEQ 5$ ::IF NOT: BR
044506 122767 000001 134622 CMPB #APTENV,$ENV ::OPERATING UNDER APT?
044514 001031 BNE 3$ ::IF NOT: BR
044516 132767 000100 134613 BITB #APTSPOOL,$ENVM ::SHOULD SPOOL MESSAGES?
044524 001425 BEQ 3$ ::IF NOT: BR
044526 017600 000004 MOV @4(SP),R0 ::GET MESSAGE ADDR.
044532 062766 000002 000004 ADD #2,4(SP) ::BUMP RETURN ADDR.
044540 005767 134552 1$: TST $MSGTYPE ::SEE IF DONE W/ LAST XMISSION?
044544 001375 BNE 1$ ::IF NOT: WAIT
044546 010067 134560 MOV R0,$MSGAD ::PUT ADDR IN MAILBOX
044552 105720 2$: TSTB (R0)+ ::FIND END OF MESSAGE
044554 001376 BNE 2$
044556 166700 134550 SUB $MSGAD,R0 ::SUB START OF MESSAGE
044562 006200 ASR R0 ::GET MESSAGE LNTH IN WORDS
044564 010067 134544 MOV R0,$MSGGLT ::PUT LENGTH IN MAILBOX
044570 012767 000004 134520 MOV #4,$MSGTYPE ::TELL APT TO TAKE MSG.
044576 000413 BR 5$
044600 017667 000004 000016 3$: MOV @4(SP),4$ ::PUT MSG ADDR IN JSR LINKAGE
044606 062766 000002 000004 ADD #2,4(SP) ::BUMP RETURN ADDRESS
044614 016746 133156 MOV 177776,-(SP) ::PUSH 177776 ON STACK
044620 004767 177044 JSR PC,$TYPE ::CALL TYPE MACRO
044624 000000 4$: .WORD 0
044626 5$:
044626 105767 000062 10$: TSTB $FFLG ::SHOULD REPORT FATAL ERROR?
044632 001416 BEQ 12$ ::IF NOT: BR
044634 005767 134476 TST $ENV ::RUNNING UNDER APT?
044640 001413 BEQ 12$ ::IF NOT: BR
044642 005767 134450 11$: TST $MSGTYPE ::FINISHED LAST MESSAGE?
044646 001375 BNE 11$ ::IF NOT: WAIT
    
```

```

044650 017667 000004 134442      MOV    @4(SP), $FATAL    ;;GET ERROR #
044656 062766 000002 000004      ADD    #2,4(SP)         ;;BUMP RETURN ADDR.
044664 005267 134426                INC    $MSGTYPE         ;;TELL APT TO TAKE ERROR
044670 105067 000020      12$:  CLRB   $FFLG         ;;CLEAR FATAL FLAG
044674 105067 000013                CLRB   $LFLG         ;;CLEAR LOG FLAG
044700 105067 000006                CLRB   $MFLG         ;;CLEAR MESSAGE FLAG
044704 012601                MOV    (SP)+,R1        ;;POP STACK INTO R1
044706 012600                MOV    (SP)+,R0        ;;POP STACK INTO R0
044710 000207                RTS    PC              ;;RETURN
044712      000      $MFLG: .BYTE 0         ;;MESSG. FLAG
044713      000      $LFLG: .BYTE 0         ;;LOG FLAG
044714      000      $FFLG: .BYTE 0         ;;FATAL FLAG
                .EVEN

```

000200
 000001
 000100
 000040
 APTSIZE=200
 APTENV=001
 APTSPool=100
 APTCSUP=040

7555
7556

```

.SBTTL TTY INPUT ROUTINE
;*****
.ENABL LSB
;*****
;*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.
044716 022767 000176 134214  $CKSWR: CMP    #SWREG,SWR    ;;IS THE SOFT-SWR SELECTED?
044724 001074                BNE    15$            ;;BRANCH IF NO
044726 105777 134212                TSTB   @$TKS         ;;CHAR THERE?
044732 100071                BPL    15$            ;;IF NO, DON'T WAIT AROUND
044734 117746 134206                MOVB   @$TKB,-(SP)    ;;SAVE THE CHAR
044740 042716 177600                BIC    #^C177,(SP)   ;;STRIP-OFF THE ASCII
044744 022726 000007                CMP    #7,(SP)+      ;;IS IT A CONTROL G?
044750 001062                BNE    15$            ;;NO, RETURN TO USER
044752 126727 134156 000001  $AUTOB,#1  CMPB   $AUTOB,#1     ;;ARE WE RUNNING IN AUTO-MODE?
044760 001456                BEQ    15$            ;;BRANCH IF YES
044762 104401 045325                TYPE   , $CNTLG      ;;ECHO THE CONTROL-G (^G)
044766 104401 045332  $GTSWR: TYPE   , $MSWR    ;;TYPE CURRENT CONTENTS
044772 016746 133200                MOV    SWREG,-(SP)   ;;SAVE SWREG FOR TYPEOUT
044776 104402                TYPOC                ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
045000 104401 045343                TYPE   , $MNEW       ;;PROMPT FOR NEW SWR
045004 005046      19$:  CLR    -(SP)         ;;CLEAR COUNTER
045006 005046                CLR    -(SP)         ;;THE NEW SWR
045010 105777 134130      7$:  TSTB   @$TKS         ;;CHAR THERE?
045014 100375                BPL    7$            ;;IF NOT TRY AGAIN
045016 117746 134124                MOVB   @$TKB,-(SP)   ;;PICK UP CHAR
045022 042716 177600                BIC    #^C177,(SP)   ;;MAKE IT 7-BIT ASCII
045026 021627 000025      9$:  CMP    (SP),#25      ;;IS IT A CONTROL-U?
045032 001005                BNE    10$           ;;BRANCH IF NOT
045034 104401 045320                TYPE   , $CNTLU      ;;YES, ECHO CONTROL-U (^U)
045040 062706 000006      20$: ADD    #6,SP         ;;IGNORE PREVIOUS INPUT
045044 000757                BR     19$           ;;LET'S TRY IT AGAIN
045046 021627 000015      10$: CMP    (SP),#15      ;;IS IT A <CR>?
045052 001022                BNE    16$           ;;BRANCH IF NO
045054 005766 000004                TST    4(SP)         ;;YES, IS IT THE FIRST CHAR?
045060 001403                BEQ    11$           ;;BRANCH IF YES
045062 016677 000002 134050  MOV    2(SP),@SWR    ;;SAVE NEW SWR

```



```

045070 052706 000006      11$:  ADD    #6,SP      ;;CLEAR UP STACK
045074 104401 001313      14$:  TYPE   $CRLF    ;;ECHO <CR> AND <LF>
045100 126727 134031 000001  CMPB   $INTAG,#1  ;;RE-ENABLE TTY KBD INTERRUPTS?
045106 001003      15$:  BNE    -15$     ;;BRANCH IF NOT
045110 012777 000100 134026  MOV    #100,@$TKS ;;RE-ENABLE TTY KBD INTERRUPTS
045116 000002      16$:  RTI                    ;;RETURN
045120 004767 176756      15$:  JSR    PC,$TYPEC  ;;ECHO CHAR
045124 021627 000060      16$:  CMP    (SP),#60   ;;CHAR < 0?
045130 002420      18$:  BLT    18$       ;;BRANCH IF YES
045132 021627 000067      18$:  CMP    (SP),#67   ;;CHAR > 7?
045136 003015      18$:  BGT    18$       ;;BRANCH IF YES
045140 042726 000060      BIC    #60,(SP)+  ;;STRIP-OFF ASCII
045144 005766 000002      TST    2(SP)     ;;IS THIS THE FIRST CHAR
045150 001403      BEQ    17$       ;;BRANCH IF YES
045152 006316      ASL    (SP)     ;;NO, SHIFT PRESENT
045154 006316      ASL    (SP)     ;;CHAR OVER TO MAKE
045156 006316      ASL    (SP)     ;;ROOM FOR NEW ONE.
045160 005266 000002      17$:  INC    2(SP)     ;;KEEP COUNT OF CHAR
045164 056616 177776      BIS    -2(SP),(SP) ;;SET IN NEW CHAR
045170 000707      BR     7$       ;;GET THE NEXT ONE
045172 104401 001312      18$:  TYPE   $QUES    ;;TYPE ?<CR><LF>
045176 000720      BR     20$     ;;SIMULATE CONTROL-U
      .DSABL  LSB
      *****
      *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
      *CALL:
      *      RDCHR      ;;INPUT A SINGLE CHARACTER FROM THE TTY
      *      RETURN HERE ;;CHARACTER IS ON THE STACK
      *                      ;;WITH PARITY BIT STRIPPED OFF
045200 011646      $RDCHR: MOV    (SP),-(SP) ;;PUSH DOWN THE PC
045202 016666 000004 000002  MOV    4(SP),2(SP) ;;SAVE THE PS
045210 105777 133730      1$:  TSTB   @$TKS    ;;WAIT FOR
045214 100375      BPL    1$       ;;A CHARACTER
045216 117766 133724 000004  MOVB   @$TKB,4(SP) ;;READ THE TTY
045224 042766 177600 000004  BIC    #^C<177>,4(SP) ;;GET RID OF JUNK IF ANY
045232 026627 000004 000023  CMP    4(SP),#23  ;;IS IT A CONTROL-S?
045240 001013      BNE    3$       ;;BRANCH IF NO
045242 105777 133676      2$:  TSTB   @$TKS    ;;WAIT FOR A CHARACTER
045246 100375      BPL    2$       ;;LOOP UNTIL ITS THERE
045250 117746 133672  MOVB   @$TKB,-(SP) ;;GET CHARACTER
045254 042716 177600      BIC    #^C177,(SP) ;;MAKE IT 7-BIT ASCII
045260 022627 000021  CMP    (SP)+,#21  ;;IS IT A CONTROL-Q?
045264 001366      BNE    2$       ;;IF NOT DISCARD IT
045266 000750      BR     1$       ;;YES, RESUME
045270 026627 000004 000140  3$:  CMP    4(SP),#140 ;;IS IT UPPER CASE?
045276 002407      BLT    4$       ;;BRANCH IF YES
045300 026627 000004 000175  CMP    4(SP),#175 ;;IS IT A SPECIAL CHAR?
045306 003003      BGT    4$       ;;BRANCH IF YES
045310 042766 000040 000004  BIC    #40,4(SP)  ;;MAKE IT UPPER CASE
045316 000002      4$:  RTI                    ;;GO BACK TO USER
045320      136      125      015  $CNTLU: .ASCIZ /^U/<15><12> ;;CONTROL 'U'
045323      012      000      015  $CNTLG: .ASCIZ /^G/<15><12> ;;CONTROL 'G'
045325      136      107      015  $MSWR: .ASCIZ <15><12>/SWR = /
045330      012      000      123  $MSWR: .ASCIZ <15><12>/SWR = /
045332      015      012      040
045335      127      122
  
```

045340 075 040 000
 045343 040 040 116
 045346 105 127 040
 045351 075 040 000

7557
 7558

```

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

```

045354 010046
 045356 016600 000002
 045362 005740
 045364 111000
 045366 006300
 045370 016000 045410
 045374 000200

```

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

```

045376 011646
 045400 016666 000004 000002
 045406 000002

```

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
$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.
:ROUTINE
:-----

```

045410 045376
 045412 043670
 045414 044246
 045416 044222
 045420 044262
 045422 044766
 045424 044716
 045426 045200
 045430 043574
 045432 043632
 7559 045434 046356
 7560 045436 046350
 7561 000030
 7562
 7563

```

$TRPAD: .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 TEST
        .LPER      ;;CALL=LPER      TRAP+13(104413) ROUTINE TO SET UP LOOP ON ERROR ADDRESS
$TERM=-.$TRPAD

```

```

.SBTTL POWER DOWN AND UP ROUTINES
:*****

```

045440 012737 045616 000024
 045446 012737 000340 000026
 045454 010046
 045456 010146
 045460 010246
 045462 010346
 045464 010446
 045466 010546
 045470 017746 133444
 045474 010667 000122
 045500 012737 045512 000024
 045506 000000
 045510 000776

```

:POWER DOWN ROUTINE
$PWRDN: MOV #SILLUP,@#PWRVEC ;;SET FOR FAST UP
        MOV #340,@#PWRVEC+2 ;;PRIO:7
        MOV R0,-(SP)        ;;PUSH R0 ON STACK
        MOV R1,-(SP)        ;;PUSH R1 ON STACK
        MOV R2,-(SP)        ;;PUSH R2 ON STACK
        MOV R3,-(SP)        ;;PUSH R3 ON STACK
        MOV R4,-(SP)        ;;PUSH R4 ON STACK
        MOV R5,-(SP)        ;;PUSH R5 ON STACK
        MOV @SWR,-(SP)      ;;PUSH @SWR ON STACK
        MOV SP,$SAVR6      ;;SAVE SP
        MOV #PWRUP,@#PWRVEC ;;SET UP VECTOR
        HALT
        BR -.2             ;;HANG UP

```

```

*****
:POWER UP ROUTINE
045512 012737 045616 000024 $PWRUP: MOV    # $ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
045520 016706 000076          MOV    $SAVR6,SP          ;;GET SP
045524 005067 000072          CLR    $SAVR6          ;;WAIT LOOP FOR THE TTY
045530 005267 000066          1$:  INC    $SAVR6          ;;WAIT FOR THE INC
045534 001375          BNE    1$              ;;OF WORD
045536 012677 133376          MOV    (SP)+,@SWR      ;;POP STACK INTO @SWR
045542 012605          MOV    (SP)+,R5        ;;POP STACK INTO R5
045544 012604          MOV    (SP)+,R4        ;;POP STACK INTO R4
045546 012603          MOV    (SP)+,R3        ;;POP STACK INTO R3
045550 012602          MOV    (SP)+,R2        ;;POP STACK INTO R2
045552 012601          MOV    (SP)+,R1        ;;POP STACK INTO R1
045554 012600          MOV    (SP)+,R0        ;;POP STACK INTO R0
045556 012737 045440 000024  MOV    # $PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
045564 012737 000340 000026  MOV    #340,@#PWRVEC+2  ;;PRIO:7
045572 104401          TYPE                                ;;REPORT THE POWER FAILURE
045574 046426          $PWRMG: .WORD  POWERM      ;;POWER FAIL MESSAGE POINTER
045576 012716          MOV    (PC)+,(SP)      ;;RESTART AT START
045600 006106          $PWRAD: .WORD  START      ;;RESTART ADDRESS
045602 042766 000020 000002  BIC    #20,2(SP)        ;;CLEAR 'T' BIT
045610 005067 175234          CLR    $TBIT          ;;CLEAR THE 'T' BIT FLAG
045614 000002          RTI
045616 000000          $ILLUP: HALT          ;;THE POWER UP SEQUENCE WAS STARTED
045620 000776          BR    .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
045622 000000          $SAVR6: 0             ;;PUT THE SP HERE

```

7564
7565
7566
7567

.SBTTL ERROR TYPE OUT ROUTINE

7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593

```

*****
:THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
:IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE
:OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
:OUT AND THEN EXECUTING A:
:*
:*      JSR    PC,ERTYPE
:*
ERTYPE: TYPE                                ;TYPE A CRLF
        .WORD  $CRLF
7574 045624 104401          MOV    @#$STSTNM,@#$TMP0
7575 045626 001313          BIC    #177400,@#$TMP0
7576 045630 113737 001102 001232  MOV    @#$ERRPC,@#$TMP1          ;GET PC OF CALL
7577 045636 042737 177400 001232  MOV    R0,-(SP)                ;SAVE R0
7578 045644 013737 001116 001234
7579 045652 010046
7580
7581 045654 113700 001114          MOV    @#$ITEMB,R0            ;GET THE ITEM NUMBER.
7582 045660 042700 177400          BIC    #177400,R0
7583 045664 001005          BNE    1$
7584
7585 045666 013746 001116          MOV    @#$ERRPC,-(SP)        ;IF ZERO THEN JUST
7586 045672 104402          TYPOC                        ;PRINT THE PC
7587 045674 000137 046250          JMP    @#ERT5
7588
7589 045700 022700 000377          1$:  CMP    #377,R0
7590 045704 001005          SNE    20$
7591 045706 016600 000004          MOV    4(SP),R0
7592 045712 011000          MOV    (R0),R0
7593 045714 062700 000400          ADD    #400,R0

```

```

7594 045720 005300          20$:  DEC      R0          ;OTHERWISE MAKE R0 AN
7595 045722 006300          ASL      R0          ;INDEX FOR THE TABLE.
7596 045724 006300          ASL      R0
7597 045726 006300          ASL      R0
7598 045730 062700 001442  ADD      #$ERRTB,R0
7599
7600 045734 012037 045744  MOV      (R0)+,@#2$    ;PICK UP THE ADDRESS
7601 045740 001404          BEQ      3$           ;OF THE EM, ERROR MESSAGE
7602 045742 104401          TYPE
7603 045744 000000          2$:  .WORD    0
7604 045746 104401          TYPE
7605 045750 001313          .WORD    $CRLF
7606
7607 045752 012037 045762  3$:  MOV      (R0)+,@#4$    ;GET THE DH,DATA HEADER
7608 045756 001404          BEQ      5$
7609 045760 104401          TYPE
7610 045762 000000          4$:  .WORD    0
7611 045764 104401          TYPE
7612 045766 001313          .WORD    $CRLF
7613
7614 045770 010146          5$:  MOV      R1,-(SP)     ;SAVE R1,R2 AND R3
7615 045772 010246          MOV      R2,-(SP)
7616 045774 010346          MOV      R3,-(SP)
7617
7618 045776 012001          MOV      (R0)+,R1     ;GET THE ADDRESS OF THE
7619                                ;DATA TABLE.
7620 046000 001516          BEQ      ERT4        ;RETURN IF NO DATA.
7621
7622 046002 011000          MOV      (R0),R0     ;GET A POINTER TO THE DATA
7623                                ;FORMAT TABLE.
7624 046004 105710          ERT1: TSTB   (R0)    ;FORMAT ZERO?
  
```

ERROR	TYPE	OUT	ROUTINE						
7626	046006	001003			BNE	7\$			
7627									
7628	046010	013146			MOV	@(R1)+,-(SP)			;FORMAT ZERO SO TYPE
7629	046012	104402			TYPOC				;AN OCTAL NUMBER.
7630	046014	000502			BR	ERT2			
7631									
7632	046016			7\$:					
7633	046016	122710	000002	8\$:	CMPB	#2,(R0)			;FORMAT TWO?
7634	046022	001010			BNE	9\$			
7635									
7636	046024	013102			MOV	@(R1)+,R2			;FORMAT TWO SO TYPE TWO
7637	046026	012246			MOV	(R2)+,-(SP)			;OCTAL NUMBERS.
7638	046030	104402			TYPOC				
7639	046032	104401			TYPE				
7640	046034	046472			.WORD	SPACE			
7641	046036	011246			MOV	(R2)+,-(SP)			
7642	046040	104402			TYPOC				
7643	046042	000467			BR	ERT2			
7644									
7645	046044	122710	000003	9\$:	CMPB	#3,(R0)			;FORMAT THREE?
7646	046050	001020			BNE	10\$			
7647									
7648	046052	013102			MOV	@(R1)+,R2			;FORMAT THREE SO TYPE
7649	046054	012246			MOV	(R2)+,-(SP)			;FOUR OCTAL NUMBERS.
7650	046056	104402			TYPOC				
7651	046060	104401			TYPE				
7652	046062	046472			.WORD	SPACE			
7653	046064	012246			MOV	(R2)+,-(SP)			
7654	046066	104402			TYPOC				
7655	046070	104401			TYPE				
7656	046072	046472			.WORD	SPACE			
7657	046074	012246			MOV	(R2)+,-(SP)			
7658	046076	104402			TYPOC				
7659	046100	104401			TYPE				
7660	046102	046472			.WORD	SPACE			
7661	046104	011246			MOV	(R2)+,-(SP)			
7662	046106	104402			TYPOC				
7663	046110	000444			BR	ERT2			
7664									
7665	046112	122710	000004	10\$:	CMPB	#4,(R0)			;FORMAT FOUR?
7666	046116	001004			BNE	11\$			
7667									
7668	046120	013146			MOV	@(R1)+,-(SP)			;FORMAR FOUR SO TYPE
7669	046122	104403			TYPOS				;AN OCTAL NUMBER
7670	046124	016			.BYTE	16			;SUPPRESSING LEADING ZEROES.
7671	046125	000			.BYTE	0			
7672	046126	000435			BR	ERT2			
7673									
7674	046130	122710	000005	11\$:	CMPB	#5,(R0)			;FORMAT FIVE?
7675	046134	001005			BNE	13\$			
7676									
7677	046136	012137	046144		MOV	(R1)+,@#12\$;FORMAT FIVE SO TYPE AN
7678	046142	104401			TYPE				;ASCIZ STRING.
7679	046144	000000		12\$:	.WORD	0			
7680	046146	000427			BR	ERT3			
7681									
7682	046150	122710	000011	13\$:	CMPB	#11,(R0)			;FORMAT ELEVEN?

```

7683 046154 001005          BNE      15$
7684
7685 046156 013137 046164    MOV      @ (R1)+, @ #14$      ;FORMAT ELEVEN SO PICK
7686 046162 104401          TYPE                                ;A POINTER TO AN ASCIZ
7687 046164 000000          14$:   .WORD      0          ;STRING.
7688 046164 000417          BR       ERT3
7689
7690 046170 122710 000012    15$:   CMPB     #12, (R0)      ;FORMAT TWELVE?
7691 046174 001011          BNE     17$
7692
7693 046176 013102          MOV      @ (R1)+, R2        ;FORMAT TWELVE SO TYPE
7694 046200 012703 000006    16$:   MOV      #6, R3        ;TYPE SIX OCTAL NUMBERS
7695 046204 012246          MOV      (R2)+, -(SP)
7696 046206 104402          TYPOC
7697 046210 104401          TYPE
7698 046212 046472          .WORD   SPACE
7699 046214 077305          SOB     R3, 16$
7700 046216 000401          BR       ERT2
7701
7702 046220 000000          17$:   HALT                ;UNDEFINED FORMAT FOR DATA?????
7703
7704 046222 104401          ERT2:  TYPE                                ;PRINT A TAB AFTER TYPING
7705 046224 046475          .WORD   $TAB                ;AN DATA TABLE ENTRY
7706                                     ;OF ALL FORMATS EXCEPT
7707                                     ;ASCIZ, FORMATS 5 OR 11
7708
7709 046226 005200          ERT3:  INC      R0            ;POINT TO THE NEXT FORMAT
7710 046230 005711          TST     (R1)                ;END OF DATA TABLE.
7711 046232 001401          BEQ     ERT4
7712 046234 000663          BR      ERT1
7713
7714 046236 104401          ERT4:  TYPE                                ;DONE.
7715 046240 001313          .WORD   $CRLF
7716 046242 012603          MOV     (SP)+, R3           ;RESTORE R1, R2 AND R3
7717 046244 012602          MOV     (SP)+, R2
7718 046246 012601          MOV     (SP)+, R1
7719 046250 012600          ERT5:  MOV     (SP)+, R0     ;RESTORE R0.
7720 046252 000207          RTS     PC                  ;AND RETURN.
7721
7722
7723
7724
7725
7726

```

.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.
*

```

```

7727
7728
7729
7730
7731 046254 011637 001236    FPSPUR: MOV     (SP), @#$TMP2   ;SAVE PC OF TRAP.
7732 046260 022626          CMP     (SP)+, (SP)+        ;RESTORE SP.
7733 046262 170200          STFPS  R0                   ;GET FPS
7734 046264 010037 001240    MOV     R0, @#$TMP3
7735 046270 170300          STST   R0                   ;GET FEC
7736 046272 010037 001242    MOV     R0, @#$TMP4
7737 046276 104377          1$:   ERROR  +377
7738 046300 000441          .WORD  441

```

7739 046302 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?).

7740 046304 000137 042564 JMP @#\$EOP

7741
7742
7743
7744

.SBTTL CPU SPURIOUS TRAP TO 4 HANDLER

::*****
:*****
:*****

*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.

7745
7746
7747 046310 011637 001236
7748 046314 022626
7749 046316 104377
7750 046320 000442
7751 046322 104412

CPSPUR: MOV (SP),@#\$TMP2 ;SAVE PC OF TRAP.

CMP (SP)+,(SP)+

1\$: 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
;THE USER TYPED CONTROL G?).

7752 046324 000137 042564 JMP @#\$EOP

7753
7754
7755
7756

.SBTTL CPU SPURIOUS TRAP TO 10 HANDLER

::*****
:*****
:*****

*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.

7757
7758
7759 046330 011637 001236
7760 046334 022626
7761 046336 104377
7762 046340 000443
7763 046342 104412

CPTWO: MOV (SP),@#\$TMP2 ;SAVE PC OF TRAP.

CMP (SP)+,(SP)+

1\$: ERROR +377

.WORD 443

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?).

7764 046344 000137 042564 JMP @#\$EOP

7765
7766
7767
7768
7769
7770
7771

.SBTTL SET LOOP ON ERROR ADDRESS ROUTINE

::*****
:*****
:*****

7772
7773 046350 011637 001110
7774 046354 000002

.LPER: MOV (SP),@#\$LPERR

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

7778
7779

```

7730
7781
7782
7783
7784
7785 046356 023727 001140 177570 .RSET: CMP @#SWR,#177570 ;SEE IF THERE IS A PHYSICAL
7786 ;CONSOLE SWITCH REGISTER.
7787 046364 001001 BNE 1$ ;BRANCH IF NO.
7788 046366 104406 CKSWR ;OTHERWISE TYPE THE CONTENTS
7789 ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
7790 ;AND GIVE THE USER A CHANCE TO
7791 ;MODIFY IT.
7792 046370 012737 046254 000244 1$: MOV #FPSPUR,@#FPVECT
7793 046376 012737 046310 000004 MOV #CPSPUR,@#ERRVECT
7794 046404 012737 046330 000010 MOV #CPTWO,@#10
7795 046412 011600 (SP),R0 ;SAVE RETURN ADDRESS.
7796 046414 012706 001100 MOV #STACK,SP ;RESET THE STACK POINTER.
7797 046420 005004 CLR R4 ;CLEAR THE FPS.
7798 046422 170104 LDFPS R4
7799 046424 000110 JMP (R0) ;RETURN.
  
```

```

7800
7801
7802 .NLIST BEX
7803
7804 ;THESE ARE SPECIAL MESSAGES:
7805
  
```

```

7806 046426 200 120 117 POWERM: .ASCIZ <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'
7807 046472 040 040 000 SPACE: .ASCIZ ' '
7808 046475 011 000 $TAB: .ASCIZ <TAB>
7809
7810 046477 107 117 124 MS1: .ASCIZ 'GOT RESULT:'<TAB><TAB>
7811 046515 105 130 120 MS2: .ASCIZ 'EXPECTED RESULT:'<TAB>
7812 046537 101 103 040 MS3: .ASCIZ 'AC OPERAND:'<TAB><TAB>
7813 046555 123 117 125 MS4: .ASCIZ 'SOURCE OPERAND:'<TAB>
7814 046537 046537 MS10=MS3
7815 046577 105 130 120 MS11: .ASCIZ 'EXPONENT OPERAND:'<TAB>
7816 046622 114 117 101 MS20: .ASCIZ 'LOADED:'<TAB><TAB>
7817 046634 124 122 111 MS21: .ASCIZ 'TRIED TO LOAD:'<TAB>
7818
7819
  
```

```

7820 ;THESE ARE ERROR MESSAGES:
7824 046654 123 124 106 EM1: .ASCIZ 'STF A,AC7 DID NOT TRAP. FID=0.'
7825 046713 123 124 106 EM2: .ASCIZ 'STF A,AC7. FPS BAD. FID=0.'
7826 046746 123 124 106 EM3: .ASCIZ 'STF A,AC7. FEC BAD. FID=0.'
7830 047001 EM4:
7834 047001 123 124 106 .ASCIZ \STF A,(R). RO BAD. FDST FAILED.\
7835 047041 EM5:
7839 047041 123 124 106 .ASCII \STF A,(R) FAILED.\
7840 047062 000 .BYTE 0
7839 047063 EM6:
7840 047063 123 124 106 .ASCII \STF A,(R). FDST FAILED.\
7841 047112 200 050 102 .ASCIZ <CRLF>\(BUT FD) ST 707 WENT TO 245 INSTEAD OF 244.\
7842 047167 EM7:
7843 047167 123 124 106 .ASCIZ \STF A,(R)+. RO BAD. FDST FAILED.\
7842 047230 EM10:
7843 047230 123 124 106 .ASCII \STF A,(R)+ FAILED.\
7843 047252 000 .BYTE 0
  
```


7844	047253				EM11:		
	047253	123	124	104		.ASCIZ	\STD A,(R)+. RO BAD. FDST FAILED.\
7845	047314				EM12:		
	047314	123	124	104		.ASCII	\STD A,(R)+ FAILED.\
7846	047336	000				.BYTE	0
7847	047337	123	124	104	EM13:	.ASCIZ	'STD A,#N TRAP TO 4 IN FDST.'
7848		047337			EM14=EM13		
7849	047373				EM15:		
	047373	123	124	104		.ASCII	\STD A,#N FAILED.\
7850	047413	000				.BYTE	0
7851	047414	120	103	040	EM16:	.ASCIZ	'PC BAD AFTER STD A,#N.'
7855	047443				EM17:		
	047443	123	124	104		.ASCIZ	\STD A,-(R) TRAP TO 4 IN FDST.\
7856	047501				EM20:		
	047501	123	124	104		.ASCIZ	\STD A,-(R). RO BAD. FDST FAILED.\
7857	047542				EM21:		
	047542	123	124	104		.ASCII	\STD A,-(R) FAILED.\
7858	047564	000				.BYTE	0
7859		047542			EM22=EM21		
7860	047565				EM23:		
	047565	123	124	104		.ASCIZ	\STD A,@(R)+ TRAP TO 4 IN FDST.\
7861	047624				EM24:		
	047624	123	124	104		.ASCIZ	\STD A,@(R)+. RO BAD. FDST FAILED.\
7862	047666				EM25:		
	047666	123	124	104		.ASCII	\STD A,@(R)+ FAILED.\
7863	047711	000				.BYTE	0
7864	047712				EM26:		
	047712	123	124	104		.ASCIZ	\STD A,@-(R) TRAP TO 4 IN FDST.\
7865	047751				EM27:		
	047751	123	124	104		.ASCIZ	\STD A,@-(R). RO BAD. FDST FAILED.\
7866	050013				EM30:		
	050013	123	124	104		.ASCII	\STD A,@-(R) FAILED.\
7867	050036	000				.BYTE	0
7868	050037				EM31:		
	050037	123	124	104		.ASCIZ	\STD A,N(R) TRAP TO 4 IN FDST.\
7869	050075				EM32:		
	050075	123	124	104		.ASCIZ	\STD A,N(R). RO BAD. FDST FAILED.\
7870	050136				EM33:		
	050136	123	124	104		.ASCII	\STD A,N(R) FAILED.\
7871	050160	000				.BYTE	0
7872	050161				EM34:		
	050161	123	124	104		.ASCIZ	\STD A,@N(R) TRAP TO 4 IN FDST.\
7873	050220				EM35:		
	050220	123	124	104		.ASCIZ	\STD A,@N(R). RO BAD. FDST FAILED.\
7874	050262				EM36:		
	050262	123	124	104		.ASCII	\STD A,@N(R) FAILED.\
7875	050305	000				.BYTE	0
7882	050306				EM37:		
	050306	123	124	103		.ASCII	'STCFD A,(R) FAILED.'
7883	050331	000				.BYTE	0
7884	050332				EM40:		
	050332	123	124	103		.ASCII	\STCFD A,(R). FPS BAD.\
7885	050357	000				.BYTE	0
7886	050360				EM41:		
	050360	123	124	103		.ASCII	\STCFD A,(R). FEC BAD.\
7887	050405	000				.BYTE	0
7888	050406				EM42:		

7889	050406	123	124	103		.ASCII	'STCFD A,(R) FAILED.'
7890	050431	200	111	116		.ASCIIZ	<CRLF>'INVERT FDFL ST 767 FAILED.'
7891	050465	123	124	103	EM43:	.ASCII	\STCFD A,(R). FPS BAD.\
7892	050512	200	050	102		.ASCIIZ	<CRLF>\(BUT EZBT) ST 560 WENT TO 061 INSTEAD OF 261.\
7893	050571	123	124	103	EM44:	.ASCII	'STCFD A,(R) FAILED.'
7894	050614	200	114	117		.ASCIIZ	<CRLF>'LOW ORDER BITS OF X11 DID NOT GET 0 ST 766.'
7895	050671	123	124	103	EM45:	.ASCII	'STCFD A,(R) FAILED.'
7896	050714	200	050	102		.ASCIIZ	<CRLF>'(BUT OP1C) ST 251 FAILED.'
7897	050747	123	124	103	EM46:	.ASCII	\STCFD A,(R). FPS BAD.\
7898	050774	200	050	102		.ASCIIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 262 INSTEAD OF 062.\
7899	051053	123	124	103	EM47:	.ASCII	'STCFD A,(R) FAILED.'
7900	051076	200	050	102		.ASCIIZ	<CRLF>\(BUT FD) ST 113 WENT TO 415 INSTEAD OF 414.\
7901	051153	123	124	103	EM50:	.ASCII	'STCFD A,(R) FAILED.'
7902	051176	040	123	111		.ASCII	' SIGN BAD.'
7903	051210	200	050	102		.ASCIIZ	<CRLF>\(BUT ENBT) ST 567 WENT TO 060 INSTEAD OF 460.\
7904	051267	123	124	103	EM51:	.ASCII	'STCDF A,(R) FAILED.'
7905	051312	000				.BYTE	0
7906	051313	123	124	104	EM52:	.ASCII	\STD A,(R). FPS BAD.\
7907	051336	000				.BYTE	0
7908	051337	123	124	104	EM53:	.ASCII	\STD A,(R). FEC BAD.\
7909	051362	000				.BYTE	0
7910	051363	123	124	103	EM54:	.ASCII	'STCDF A,(R) FAILED.'
7911	051406	200	111	116		.ASCIIZ	<CRLF>'INVERT FDFL ST 767 FAILED.'
7912	051442	123	124	103	EM55:	.ASCII	'STCDF A,(R) FAILED.'
7913	051465	200	122	117		.ASCII	<CRLF>'ROUND ERROR, OR'
7914	051505	200	050	102		.ASCIIZ	<CRLF>\(BUT BREAKOUT) ST 400 WENT TO 766 INSTEAD OF 767.\
7915	051570	123	124	104	EM56:	.ASCII	\STD A,(R). FPS BAD.\
7916	051613	200	050	102		.ASCIIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 062 INSTEAD OF 262.\
7917	051672	123	124	104	EM57:	.ASCII	\STD A,(R). FPS BAD.\
7918	051715	040	106	111		.ASCII	' FIV=0.'
7919	051724	200	050	102		.ASCIIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 123 INSTEAD OF 103.\
7920	052002	123	124	103	EM60:	.ASCII	'STCDF A,(R) FAILED.'
7921	052025	040	106	111		.ASCII	' FIV=1.'
7922	052034	200	050	102		.ASCIIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 103 INSTEAD OF 123.\
7923	052112	123	124	104	EM61:	.ASCII	\STD A,(R). FPS BAD.\
7924	052135	200	050	102		.ASCIIZ	<CRLF>\(BUT FLAG) ST 147 WENT TO 361 INSTEAD OF 365.\
7925	052214	123	124	103	EM62:	.ASCII	'STCFD A,AC6. FPS BAD.'
7926	052241	200	050	102		.ASCIIZ	<CRLF>\(BUT FDST) ST 767 WENT TO 567 INSTEAD OF 577.\
7927	052320	123	124	103	EM63:	.ASCIIZ	'STCFD A,AC6. FEC BAD.'
7928	052346	103	114	122	EM64:	.ASCII	\CLRD (R) FAILED.\
7929	052366	200	132	105		.ASCIIZ	<CRLF>'ZERO X11 AT ST 770 FAILED.'

7938	052422				EM65:	.ASCII	\CLRD (R). FPS BAD.\
	052422	103	114	122		.BYTE	0
7939	052444	000					
7940	052445				EM66:	.ASCIZ	\CLRD (R). RO BAD. FDST FAILED.\
	052445	103	114	122			
7941/	052504				EM67:	.ASCII	\CLRD AC7. FPS BAD.\
	052504	103	114	122		.ASCIZ	<CRLF>\(BUT FDST) ST 770 WENT TO 607 INSTEAD OF 617.\
7942	052526	200	050	102			
7943	052605				EM70:	.ASCII	\CLRD AC7. FEC BAD.\
	052605	103	114	122		.BYTE	0
7944	052627	000					
7945	052630	116	105	107	EM176:	.ASCIZ	'NEGF AC7. FPS BAD.'
7946	052653	116	105	107	EM177:	.ASCIZ	'NEGF AC7. FEC BAD.'
7959	052676				EM71:	.ASCIZ	\NEGF A FAILED.\
	052676	116	105	107			
7960	052715				EM72:	.ASCIZ	\NEGF A. FPS BAD.\
	052715	116	105	107			
7961	052736				EM107:	.ASCIZ	\NEGD (R) TRAP TO 4 IN SRC MODE.\
	052736	116	105	107			
7962	052776				EM73:	.ASCIZ	\NEGD (R) FAILED.\
	052776	116	105	107			
7963	053017				EM74:	.ASCIZ	\NEGD (R). RO BAD.\
	053017	116	105	107			
7964	053041				EM75:	.ASCIZ	\NEGD (R). FPS BAD.\
	053041	116	105	107			
7965	053064				EM76:	.ASCIZ	\ABSD (R)+ TRAP TO 4 IN SRC MODE.\
	053064	101	102	123			
7966	053125				EM77:	.ASCIZ	\ABSD (R)+ FAILED.\
	053125	101	102	123			
7967	053147				EM100:	.ASCIZ	\ABSD (R)+. RO BAD.\
	053147	101	102	123			
7968	053172				EM101:	.ASCIZ	\ABSD (R)+. FPS BAD.\
	053172	101	102	123			
7969	053216				EM102:	.ASCIZ	\ABSD -(R) TRAP TO 4 IN SRC MODE.\
	053216	101	102	123			
7970	053257				EM103:	.ASCIZ	\ABSD -(R) FAILED.\
	053257	101	102	123			
7971	053301				EM104:	.ASCIZ	\ABSD -(R). RO BAD.\
	053301	101	102	123			
7972	053324				EM105:	.ASCIZ	\ABSD -(R). FPS BAD.\
	053324	101	102	123			
7973	053350				EM106:	.ASCIZ	\ABSD @ (R)+ TRAP TO 4 IN SRC MODE.\
	053350	101	102	123			
7974	053412				EM110:	.ASCIZ	\ABSD @ (R)+ FAILED.\
	053412	101	102	123			
7975	053435				EM111:	.ASCIZ	\ABSD @ (R)+. RO BAD.\
	053435	101	102	123			
7976	053461				EM112:	.ASCIZ	\ABSD @ (R)+. FPS BAD.\
	053461	101	102	123			
7977	053506				EM113:	.ASCIZ	\NEGD @-(R) TRAP TO 4 IN SRC MODE.\
	053506	116	105	107			
7978	053550				EM114:	.ASCIZ	\NEGD @-(R) FAILED.\
	053550	116	105	107			
7979	053573				EM115:	.ASCIZ	\NEGD @-(R). RO BAD.\
	053573	116	105	107			
7980	053617				EM116:	.ASCIZ	\NEGD @-(R). FPS BAD.\
	053617	116	105	107			

7981	053644				EM117:	.ASCIZ \ABSD N(R) TRAP TO 4 IN SRC MODE.\
	053644	101	102	123		
7982	053705				EM120:	.ASCIZ \ABSD N(R) FAILED.\
	053705	101	102	123		
7983	053727				EM121:	.ASCIZ \ABSD N(R). RO BAD.\
	053727	101	102	123		
7984	053752				EM122:	.ASCIZ \ABSD N(R). FPS BAD.\
	053752	101	102	123		
7985	053776				EM123:	.ASCIZ \NEGD @N(R) TRAP TO 4 IN SRC MODE.\
	053776	116	105	107		
7986	054040				EM124:	.ASCIZ \NEGD @N(R) FAILED.\
	054040	116	105	107		
7987	054063				EM125:	.ASCIZ \NEGD @N(R). RO BAD.\
	054063	116	105	107		
7988	054107				EM126:	.ASCIZ \NEGD @N(R). FPS BAD.\
	054107	116	105	107		
7989	054134				EM127:	.ASCIZ \NEGD N(R7) TRAP TO 4 IN SRC MODE.\
	054134	116	105	107		
7990	054176				EM130:	.ASCIZ \NEGD N(R7) FAILED.\
	054176	116	105	107		
7991	054221				EM131:	.ASCIZ \NEGD N(R7). FPS BAD.\
	054221	116	105	107		
7992	054246				EM132:	.ASCIZ \ABSD @N(R7) TRAP TO 4 IN SRC MODE.\
	054246	101	102	123		
7993	054311				EM133:	.ASCIZ \ABSD @N(R7) FAILED.\
	054311	101	102	123		
7994	054335				EM134:	.ASCIZ \ABSD @N(R7). FPS BAD.\
	054335	101	102	123		
8004	054363				EM135:	.ASCIZ 'NEGD A FAILED.'
	054363	116	105	107		
8005	054401					.ASCIZ <CRLF>'XOR SIGN BIT ST 336 FAILED.'
	054401	200	130	117		
8006	054436				EM136:	.ASCIZ \NEGD A FAILED.\
	054436	116	105	107		
8007	054455				EM137:	.ASCIZ \NEGD A. FPS BAD.\
	054455	116	105	107		
8008	054476				EM140:	.ASCIZ \NEGD (R) FAILED.\
	054476	116	105	107		
8009	054517				EM141:	.ASCIZ \NEGD (R). RO BAD. SPECIAL DEST FAILED.\
	054517	116	105	107		
8010	054566				EM142:	.ASCIZ \NEGD (R). FPS BAD.\
	054566	116	105	107		
8011	054611				EM143:	.ASCIZ \NEGD (R)+ FAILED.\
	054611	116	105	107		
8012	054633				EM144:	.ASCIZ \NEGD (R)+. RO BAD. SPECIAL DEST FAILED.\
	054633	116	105	107		
8013	054703				EM145:	.ASCIZ \NEGD (R)+. FPS BAD.\
	054703	116	105	107		
8014	054727				EM146:	.ASCIZ \NEGD -(R) FAILED.\
	054727	116	105	107		
8015	054751				EM147:	.ASCIZ \NEGD -(R). RO BAD. SPECIAL DEST FAILED.\
	054751	116	105	107		
8016	055021				EM150:	.ASCIZ \NEGD -(R). FPS BAD.\
	055021	116	105	107		
8017	055045				EM151:	.ASCIZ \NEGD @N(R)+ FAILED.\
	055045	116	105	107		
8018	055070				EM152:	.ASCIZ \NEGD @N(R)+. RO BAD. SPECIAL DEST FAILED.\
	055070	116	105	107		
8019	055141				EM153:	

	055141	116	105	107		.ASCIZ \NEGD @ (R)+. FPS BAD.\
8020	055166				EM154:	
	055166	116	105	107		.ASCIZ \NEGD @-(R) FAILED.\
8021	055211				EM155:	
	055211	116	105	107		.ASCIZ \NEGD @-(R). RO BAD. SPECIAL DEST FAILED.\
8022	055262				EM156:	
	055262	116	105	107		.ASCIZ \NEGD @-(R). FPS BAD.\
8023	055307				EM157:	
	055307	116	105	107		.ASCIZ \NEGF (R)+ FAILED.\
8024	055331	116	105	107	EM160:	.ASCII 'NEGF (R)+. RO BAD.'
8025	055353	102	101	104		.ASCIZ 'BAD CONSTANT USED. SPECIAL DEST FAILED.'
8026	055423				EM161:	
	055423	116	105	107		.ASCIZ \NEGF (R)+. FPS BAD.\
8027	055447				EM162:	
	055447	116	105	107		.ASCIZ \NEGD (R7)+ FAILED.\
8028	055472				EM163:	
	055472	116	105	107		.ASCIZ \NEGD (R7)+. FPS BAD.\
8029	055517	120	103	040	EM164:	.ASCIZ 'PC BAD AFTER NEGD (R7)+. BAD CONSTANT USED.'
8034	055573				EM215:	
	055573	120	103	040		.ASCII \PC BAD AFTER NEGD N(R). BAD CONSTANT USED 746 746.\
	055655	200	117	122		.ASCIZ <CRLF>'OR (BUT FDST) IN SPECIAL DEST FAILED.'
8035	055724				EM216:	
	055724	116	105	107		.ASCIZ \NEGD N(R) FAILED.\
8036	055746				EM217:	
	055746	116	105	107		.ASCIZ \NEGD N(R). RO BAD. SPECIAL DEST FAILED.\
8037	056016				EM220:	
	056016	116	105	107		.ASCIZ \NEGD N(R). FPS BAD.\
8038	056042				EM221:	
	056042	120	103	040		.ASCII \PC BAD AFTER NEGD @N(R). BAD CONSTANT USED 747 747.\
	056125	200	117	122		.ASCIZ <CRLF>'OR (BUT FDST) IN SPECIAL DEST FAILED.'
8039	056174				EM222:	
	056174	116	105	107		.ASCIZ \NEGD @N(R) FAILED.\
8040	056217				EM223:	
	056217	116	105	107		.ASCIZ \NEGD @N(R). RO BAD. SPECIAL DEST FAILED.\
8041	056270				EM224:	
	056270	116	105	107		.ASCIZ \NEGD @N(R). FPS BAD.\
8057	056315				EM165:	
	056315	116	105	107		.ASCIZ \NEGD (R) FAILED.\
8058	056336				EM166:	
	056336	101	102	123		.ASCIZ \ABSD (R) FAILED.\
8059	056357				EM167:	
	056357	124	123	124		.ASCIZ \TSTD (R) FAILED.\
8060	056400				EM170:	
	056400	116	105	107		.ASCIZ \NEGD (R). FPS BAD.\
8061	056423				EM171:	
	056423	101	102	123		.ASCIZ \ABSD (R). FPS BAD.\
8062	056446				EM172:	
	056446	124	123	124		.ASCIZ \TSTD (R). FPS BAD.\
8063	056471				EM173:	
	056471	116	105	107		.ASCIZ \NEGD (R). FEC BAD.\
8064	056514				EM174:	
	056514	101	102	123		.ASCIZ \ABSD (R). FEC BAD.\
8065	056537				EM175:	
	056537	124	123	124		.ASCIZ \TSTD (R). FEC BAD.\
8066	056562				EM200:	
	056562	116	105	107		.ASCII \NEGD (R) FAILED.\
8067	056602	200	130	117		.ASCIZ <CRLF>'XOR SIGN BIT FAILED ST 336.'

8068	056637				EM201:	.ASCII \WEGD (R). FPS BAD.\
	056637	116	105	107		.ASCIIZ <CRLF>\(BUT ENBT) ST 336 WENT TO 053 INSTEAD OF 453.\
8069	056661	200	050	102		
8070	056740				EM202:	.ASCII \WEGD (R). FPS BAD.\
	056740	116	105	107		.ASCIIZ <CRLF>\(BUT ENBT) ST 336 WENT TO 453 INSTEAD OF 053.\
8071	056762	200	050	102		
8072	057041				EM203:	.ASCII \ABSD (R) FAILED.\
	057041	101	102	123		.ASCII <CRLF>'(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 335, OR'
8073	057061	200	050	102		.ASCIIZ <CRLF>\(BUT ENBT) ST 335 WENT TO 452 INSTEAD OF 052.\
8074	057142	200	050	102		
8075	057221				EM204:	.ASCII \ABSD (R) FAILED.\
	057221	101	102	123		.ASCIIZ <CRLF>'XOR SIGN BIT FAILED ST 452.'
8076	057241	200	130	117		
8077	057276				EM205:	.ASCII \TSTD (R) FAILED.\
	057276	124	123	124		.ASCIIZ <CRLF>\(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 334.\
8078	057316	200	050	102		
8079	057375				EM206:	.ASCII \TSTD (R). FPS BAD.\
	057375	124	123	124		.ASCIIZ <CRLF>\(BUT ENBT) ST 334 WENT TO 453 INSTEAD OF 053.\
8080	057417	200	050	102		
8081	057476				EM207:	.ASCII \TSTD (R) FAILED.\
	057476	124	123	124		.ASCIIZ <CRLF>\(BUT OP1B) ST 057 WENT TO 335 INSTEAD OF 334.\
8082	057516	200	050	102		
8083	057575				EM210:	.ASCII \TSTD (R) FAILED.\
	057575	124	123	124		.ASCIIZ <CRLF>\(BUT ENBT) ST 334 WENT TO 053 INSTEAD OF 453.\
8084	057615	200	050	102		
8085	057674				EM211:	.ASCII \TSTD (R) FAILED.\
	057674	124	123	124		.ASCIIZ <CRLF>\(BUT OP1B) ST 255 WENT TO 311 OR 312 INSTEAD OF 310.\
8086	057714	200	050	102		
8087	060002				EM212:	.ASCII \TSTD (R). FPS BAD.\
	060002	124	123	124		.ASCIIZ <CRLF>\(BUT ENBT) ST 310 WENT TO 402 INSTEAD OF 002.\
8088	060024	200	050	102		
8089	060103				EM213:	.ASCII \TSTD (R). FPS BAD.\
	060103	124	123	124		.ASCII ' FIUV=0, OPERAND=-0.'
8090	060125	040	106	111		.ASCIIZ <CRLF>\(BUT FIUV) ST 257 WENT TO 355 INSTEAD OF 255.\
8091	060151	200	050	102		
8092	060230				EM214:	.ASCII \TSTD (R). FPS BAD.\
	060230	124	123	124		.ASCII ' FIUV=1, OPERAND=-0.'
8093	060252	040	106	111		.ASCIIZ <CRLF>\(BUT FIUV) ST 257 WENT TO 255 INSTEAD OF 355.\
8094	060276	200	050	102		
8095						
8105						
8106	060355				EM225:	.ASCIIZ \LDFPS (R). RO BAD.\
	060355	114	104	106		
8107	060400				EM226:	.ASCIIZ \LDFPS (R). FPS BAD.\
	060400	114	104	106		
8108	060424				EM227:	.ASCIIZ \LDFPS (R) TRAPPED TO 4.\
	060424	114	104	106		
8109						
8110	060454				EM230:	.ASCIIZ \LDFPS (R)+. RO BAD.\
	060454	114	104	106		
8111	060500				EM231:	.ASCIIZ \LDFPS (R)+. FPS BAD.\
	060500	114	104	106		
8112	060525				EM232:	.ASCIIZ \LDFPS (R)+ TRAPPED TO 4.\
	060525	114	104	106		
8113						
8114	060556				EM233:	.ASCIIZ \LDFPS -(R). RO BAD.\
	060556	114	104	106		

8115	060602				EM234:	.ASCIZ \LDFPS -(R). FPS BAD.\
8116	060602	114	104	106	EM235:	.ASCIZ \LDFPS -(R) TRAPPED TO 4.\
8117	060627	114	104	106		
8118	060660				EM236:	.ASCIZ \LDFPS @(R)+. RO BAD.\
8119	060660	114	104	106	EM237:	.ASCIZ \LDFPS @(R)+. FPS BAD.\
8120	060705	114	104	106	EM240:	.ASCIZ \LDFPS @(R)+ TRAPPED TO 4.\
8121	060733	114	104	106		
8122	060765				EM241:	.ASCIZ \LDFPS @-(R). RO BAD.\
8123	060765	114	104	106	EM242:	.ASCIZ \LDFPS @-(R). FPS BAD.\
8124	061012	114	104	106	EM243:	.ASCIZ \LDFPS @-(R) TRAPPED TO 4.\
8125	061040	114	104	106		
8129	061040					
8130	061072				EM244:	.ASCIZ \LDFPS N(R). RO BAD.\
8131	061072	114	104	106	EM245:	.ASCIZ \LDFPS N(R). FPS BAD.\
8132	061116	114	104	106	EM246:	.ASCIZ \PC BAD AFTER LDFPS N(R).\
8133	061143	120	103	040	EM247:	.ASCIZ \LDFPS N(R) TRAPPED TO 4.\
8134	061174	114	104	106		
8135	061225				EM250:	.ASCIZ \LDFPS @N(R). RO BAD.\
8136	061225	114	104	106	EM251:	.ASCIZ \LDFPS @N(R). FPS BAD.\
8137	061252	114	104	106	EM252:	.ASCIZ \PC BAD AFTER LDFPS @N(R).\
8138	061300	120	103	040	EM253:	.ASCIZ \LDFPS @N(R) TRAPPED TO 4.\
8139	061332	114	104	106		
8140	061364				EM254:	.ASCIZ \PC BAD AFTER LDCLD (R7)+,A.\
8141	061364	120	103	040	EM255:	.ASCIZ \LDCLD (R7)+,A TRAPPED TO 4.\
8142	061420	114	104	103		
8143	061454				EM256:	.ASCIZ \LDCLD (R)+,A. RO BAD.\
8144	061454	114	104	103	EM257:	.ASCIZ \LDCLD (R)+,A. FPS BAD.\
8145	061502	114	104	103		
8152	061502					
8153	061531				EM260:	.ASCII \LDCIF OR LDCLF (R),A FAILED.\
8154	061531	114	104	103		.BYTE 0
8155	061565	000			EM261:	.ASCII \LDCIF OR LDCLF (R),A. FPS BAD.\
8156	061566	114	104	103		.BYTE 0
8157	061624	000				
8158	061625				EM262:	

8159	061625	114	104	103		.ASCII \LDCIF (R),A FAILED.\
8160	061650	200	050	102		.ASCIZ <CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
8161	061725				EM263:	
8162	061725	114	104	103		.ASCII \LDCLF (R),A. FPS BAD.\
8163	061752	000				.BYTE 0
8164	061753				EM264:	
8165	061753	114	104	103		.ASCII \LDCIF (R),A FAILED.\
8166	061776	200	125	123		.ASCIZ <CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
8167	062050				EM265:	
8168	062050	114	104	103		.ASCII \LDCIF OR LDCLF (R),A FAILED.\
8169	062104	200	123	105		.ASCIZ <CRLF>'SET SIGN BIT FAILED ST 146.'
8170	062141				EM266:	
8171	062141	114	104	103		.ASCII \LDCIF OR LDCLF (R),A FAILED.\
8172	062175	200	050	102		.ASCIZ <CRLF>\(BUT XNBT) ST 372 WENT TO 152 INSTEAD OF 112.\
8173	062254				EM267:	
8174	062254	114	104	103		.ASCII \LDCLF (R),A FAILED.\
8175	062277	200	125	123		.ASCIZ <CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'
8176	062351				EM270:	
8177	062351	114	104	103		.ASCII \LDCLF (R),A FAILED.\
8178	062374	040	122	117		.ASCIZ ' ROUND ERROR.'
8179	062412				EM271:	
8180	062412	114	104	103		.ASCII \LDCLF (R),A FAILED.\
8181	062435	040	124	122		.ASCIZ ' TRUNCATION ERROR.'
8182	062460				EM272:	
8183	062460	114	104	103		.ASCII \LDCIF OR LDCLF (R),A FAILED.\
8184	062514	200	122	061		.ASCIZ <CRLF>'R14 NOT INCREMENTED ST 630.'
8185	062551				EM273:	
8186	062551	114	104	103		.ASCII \LDCID OR LDCLD (R),A FAILED.\
8187	062605	000				.BYTE 0
8188	062606				EM274:	
8189	062606	114	104	103		.ASCII \LDCID OR LDCLD (R),A. FPS BAD.\
8190	062644	000				.BYTE 0
8191	062645				EM275:	
8192	062645	114	104	103		.ASCII \LDCID (R),A FAILED.\
8193	062670	200	050	102		.ASCIZ <CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
8194	062745				EM276:	
8195	062745	114	104	103		.ASCII \LDCID (R),A FAILED.\
8196	062770	200	125	123		.ASCIZ <CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
8197	063042				EM277:	
8198	063042	114	104	103		.ASCII \LDCID (R),A FAILED.\
8199	063065	200	123	105		.ASCIZ <CRLF>'SET SIGN FAILED ST 146.'
8200	063116				EM300:	
8200	063116	114	104	103		.ASCII \LDCLD (R),A FAILED.\
8200	063141	200	125	123		.ASCIZ <CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'

8201							
8202	063213				EM301:	.ASCII	\LDEXP (R),A FAILED.\
	063213	114	104	105		.BYTE	0
8203	063236	000					
8204	063237				EM302:	.ASCII	\LDEXP (R),A. FPS BAD.\
	063237	114	104	105		.BYTE	0
8205	063264	000					
8206	063265	114	104	105	EM303:	.ASCIZ	'LDEXP (R),A. FEC BAD.'
8207							
8208	063313				EM304:	.ASCII	\LDEXP (R),A FAILED.\
	063313	114	104	105		.ASCIZ	<CRLF>'EXCESS 200 CALCULATION ST 624 BAD.'
8209	063336	200	105	130			
8210							
8211	063402				EM305:	.ASCII	\LDEXP (R),A. FPS BAD.\
	063402	114	104	105		.ASCII	'(BUT ENBT,EZBT,XNBT) ST 625 DID NOT GO TO 304.'
8212	063427	050	102	125			
8213							
8214	063505				EM306:	.ASCII	\LDEXP (R),A. FPS BAD.\
	063505	114	104	105		.ASCII	<CRLF>'(BUT EZBT) ST 544 WENT TO 504 INSTEAD OF 704, OR'
8215	063532	200	050	102		.ASCIZ	<CRLF>\(BUT EZBT) ST 704 WENT TO 264 INSTEAD OF 064.\
8216	063613	200	050	102			
8217							
8218	063672				EM307:	.ASCII	\LDEXP (R),A FAILED.\
	063672	114	104	105		.ASCIZ	<CRLF>\(BUT EZBT) ST 704 WENT TO 064 INSTEAD OF 264.\
8219	063715	200	050	102			
8220							
8221	063774				EM310:	.ASCII	\LDEXP (R),A. FPS BAD.\
	063774	114	104	105		.ASCIZ	<CRLF>\(BUT FIU) ST 264 WENT TO 115 INSTEAD OF 155.\
8222	064021	200	050	102			
8223							
8224	064077				EM311:	.ASCII	\LDEXP (R),A FAILED.\
	064077	114	104	105		.ASCIZ	<CRLF>\(BUT FIU) ST 264 WENT TO 155 INSTEAD OF 115.\
8225	064122	200	050	102			
8226							
8227	064200				EM312:	.ASCII	\LDEXP (R),A FAILED.\
	064200	114	104	105		.ASCIZ	<CRLF>\(BUT EZBT) ST 544 WENT TO 704 INSTEAD OF 504.\
8228	064223	200	050	102			
8229							
8230	064302				EM313:	.ASCII	\LDEXP (R),A FAILED.\
	064302	114	104	105		.ASCIZ	<CRLF>\(BUT FIU) ST 504 WENT TO 155 INSTEAD OF 115.\
8231	064325	200	050	102			
8232							
8233	064403				EM314:	.ASCII	\LDEXP (R),A FAILED.\
	064403	114	104	105		.ASCIZ	<CRLF>\(BUT FIV) ST 104 WENT TO 116 INSTEAD OF 136.\
8234	064426	200	050	102			
8235							
8236	064504				EM315:	.ASCII	\LDEXP (R),A FAILED.\
	064504	114	104	105		.ASCIZ	<CRLF>\(BUT FIV) ST 104 WENT TO 136 INSTEAD OF 116.\
8237	064527	200	050	102			
8238							
8239	064605				EM316:	.ASCII	\LDEXP (R),A FAILED.\
	064605	114	104	105		.ASCIZ	<CRLF>\(BUT FIV) ST 144 WENT TO 116 INSTEAD OF 136.\
8240	064630	200	050	102			
8241							
8242	064706				EM317:	.ASCII	\LDEXP (R),A FAILED.\
	064706	114	104	105		.ASCIZ	<CRLF>\(BUT FIV) ST 144 WENT TO 136 INSTEAD OF 116.\
8243	064731	200	050	102			

8244						
8245	065007				EM320:	.ASCII \LDEXP (R),A FAILED.\
	065007	114	104	105		.ASCIIZ <CRLF>\(BUT FIV) ST 344 WENT TO 116 INSTEAD OF 136.\
8246	065032	200	050	102		
8247						
8248	065110				EM321:	.ASCII \LDEXP (R),A FAILED.\
	065110	114	104	105		.ASCIIZ <CRLF>\(BUT FIV) ST 344 WENT TO 136 INSTEAD OF 116.\
8249	065133	200	050	102		
8250						
8251	065211				EM322:	.ASCII \STCDI OR STCDL (R),A FAILED.\
	065211	123	124	103		.BYTE 0
8252	065245	000				
8253						
8254	065246				EM323:	.ASCII \STCDI OR STCDL (R),A. FPS BAD.\
	065246	123	124	103		.BYTE 0
8255	065304	000				
8256						
8257	065305	123	124	103	EM324:	.ASCIIZ 'STCDI OR STCDL (R),A. FEC BAD.'
8258						
8259	065344				EM325:	.ASCII \STCDL (R),A. FPS BAD.\
	065344	123	124	103		.ASCII <CRLF>'CLEAR FLAG ST 774 FAILED, OR'
8260	065371	200	103	114		.ASCIIZ <CRLF>\(BUT FLAG) ST 662 WENT TO 365 INSTEAD OF 361.\
8261	065426	200	050	102		
8262						
8263	065344				EM326=EM325	
8264						
8265	065505				EM327:	.ASCII \STCDL (R),A FAILED.\
	065505	123	124	103		.ASCIIZ <CRLF>\(BUT ENBT) ST 632 WENT TO 473 INSTEAD OF 073.\
8266	065530	200	050	102		
8267						
8268	065607				EM330:	.ASCII \STCDL (R),A. FPS BAD.\
	065607	123	124	103		.ASCIIZ <CRLF>\(BUT FIC) ST 004 WENT TO 305 INSTEAD OF 315.\
8269	065634	200	050	102		
8270						
8271	065712				EM331:	.ASCII \STCDL (R),A. FPS BAD.\
	065712	123	124	103		.ASCIIZ <CRLF>\(BUT FIC) ST 004 WENT TO 315 INSTEAD OF 305.\
8272	065737	200	050	102		
8273						
8274	065246				EM333=EM323	
8275						
8276	066015				EM334:	.ASCII \STCDI (R),A. FPS BAD.\
	066015	123	124	103		.ASCIIZ <CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66.'
8277	066042	200	125	123		
8278						
8279	066111				EM335:	.ASCII \STCDI (R),A FAILED.\
	066111	123	124	103		.ASCIIZ <CRLF>\(BUT ENBT) ST 632 WENT TO 073 INSTEAD OF 473.\
8280	066134	200	050	102		
8281						
8282	066213				EM336:	.ASCII \STCDI (R),A. FPS BAD.\
	066213	123	124	103		

FLAG RESET AND CONSOLE TEST ROUTINE

8284	066240	200	123	105		.ASCIZ	<CRLF>'SET FN ST 473 FAILED.'
8285							
8286	066267				EM337:	.ASCII	\STCDL (R),A FAILED.\
8287	066312	123	124	103		.ASCIZ	<CRLF>\(BUT COUT) ST 275 WENT TO 074 INSTEAD OF 274.\
8288		200	050	102			
8289	066371				EM340:	.ASCII	\STCDL (R),A FAILED.\
8290	066414	123	124	103		.ASCIZ	<CRLF>\(BUT COUT) ST 275 WENT TO 274 INSTEAD OF 074.\
8291		200	050	102			
8292	066473				EM341:	.ASCII	\STCDL (R),A. FPS BAD.\
8293	066520	123	124	103		.ASCIZ	<CRLF>\(BUT EZBT) ST 377 WENT TO 633 INSTEAD OF 433.\
8294		200	050	102			
8295	066577				EM342:	.ASCII	\STCDL (R),A FAILED.\
8296	066622	123	124	103		.ASCIZ	<CRLF>\(BUT COUT) ST 360 WENT TO 654 INSTEAD OF 454.\
8297		200	050	102			
8298	066701				EM343:	.ASCII	\STCDL (R),A FAILED.\
8299	066724	123	124	103		.ASCIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 531 INSTEAD OF 431.\
8300		200	050	102			
8301	067003				EM344:	.ASCII	\STCDL (R),A FAILED.\
8302	067026	123	124	103		.ASCIZ	<CRLF>'(BUT COUT) ST 360 WENT TO 454 INSTEAD OF 654, OR'
8303	067107	200	050	102		.ASCIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 431 INSTEAD OF 531.\
8304		200	050	102			
8305	067166				EM332:	.ASCII	\STCDI (R),A FAILED.\
8306	067211	123	124	103		.ASCIZ	<CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66.'
8307		200	125	123			
8308	067260				EM345:	.ASCII	\STCDI (R),A FAILED.\
8309	067303	123	124	103		.ASCIZ	<CRLF>\(BUT FL) ST 633 WENT TO 655 INSTEAD OF 654.\
8310		200	050	102			
8311	067360				EM346:	.ASCII	\STCFL (R),A FAILED.\
8312	067403	123	124	103		.ASCIZ	<CRLF>'ZERO LOW ORDER PART OF X11 FAILED ST 773.'
8313		200	132	105			
8314							
8315	067456				EM347:	.ASCII	\STEXP A,(R) FAILED.\
8316	067501	123	124	105		.BYTE	0
8317		000					
8318	067502				EM350:	.ASCII	\STEXP A,(R). FPS BAD.\
8319	067527	123	124	105		.BYTE	0
8320		000					
8321	067530	115	117	122	EM351:	.ASCII	'MORE THAN ONE WORD '
8322	067553	127	122	111		.ASCIZ	'WRITTEN BY STEXP A,(R).<CRLF>'ZERO FDFL ST 347 FAILED.'
8323							
8324	067634				EM352:	.ASCII	\STEXP A,(R). FPS BAD.\
8325	067661	123	124	105		.ASCIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 071 INSTEAD OF 471.\
8326		200	050	102			
8327	067740				EM353:	.ASCII	\STEXP A,(R). FPS BAD.\
	067740	123	124	105			

8328	067765	200	050	102		.ASCIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 072 INSTEAD OF 272.\
8329							
8330	070044				EM354:	.ASCII	\STEXP A,(R). FPS BAD.\
	070044	123	124	105		.ASCIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 272 INSTEAD OF 072.\
8331	070071	200	050	102			
8332							
8333	070150				EM355:	.ASCII	\STEXP A,(R). FPS BAD.\
	070150	123	124	105		.ASCIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 471 INSTEAD OF 071.\
8334	070175	200	050	102			
8335							
8336	070254	123	124	123	EM356:	.ASCII	'STST (R) GOT BAD FEC.'<CRLF>
8337	070302	101	106	124		.ASCIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
8338							
8339	070352	123	124	123	EM357:	.ASCII	'STST (R) GOT BAD FEA.'<CRLF>
8340	070400	101	106	124		.ASCIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
8341							
8342	070450	117	116	114	EM360:	.ASCII	'ONLY ONE WORD WRITTEN BY STST (R). '
8343	070513	123	105	124		.ASCIZ	'SET FDFL ST 636 FAILED.'
8344							

8358							
8359	070543				EM401:		
	070543	123	124	106		.ASCIZ	\STFPS (R). RO BAD.\
8360	070566				EM402:		
	070566	123	124	106		.ASCIZ	\STFPS (R) FAILED.\
8361	070610	115	117	122	EM403:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R).'
8362	070660	200	050	102		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8363	070742				EM404:		
	070742	123	124	106		.ASCIZ	\STFPS (R) TRAPPED TO 4.\
8364							
8365	070772				EM405:		
	070772	123	124	106		.ASCIZ	\STFPS (R)+. RO BAD.\
8366	071016				EM406:		
	071016	123	124	106		.ASCIZ	\STFPS (R)+ FAILED.\
8367	071041	115	117	122	EM407:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R)+.'
8368	071112	200	050	102		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8369	071174				EM410:		
	071174	123	124	106		.ASCIZ	\STFPS (R)+ TRAPPED TO 4.\
8370							
8371	071225				EM411:		
	071225	123	124	106		.ASCIZ	\STFPS -(R). RO BAD.\
8372	071251				EM412:		
	071251	123	124	106		.ASCIZ	\STFPS -(R) FAILED.\
8373	071274	115	117	122	EM413:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS -(R).'
8374	071345	200	050	102		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8375	071427				EM414:		
	071427	123	124	106		.ASCIZ	\STFPS -(R) TRAPPED TO 4.\
8376							
8377	071460				EM415:		
	071460	123	124	106		.ASCIZ	\STFPS @ (R)+. RO BAD.\
8378	071505				EM416:		
	071505	123	124	106		.ASCIZ	\STFPS @ (R)+ FAILED.\
8379	071531	123	124	106	EM417:	.ASCIZ	'STFPS @ (R)+ DID NOT DEFFER THE WRITE.'
8380	071577				EM420:		
	071577	123	124	106		.ASCIZ	\STFPS @ (R)+ TRAPPED TO 4.\
8381							
8382	071631				EM421:		
	071631	123	124	106		.ASCIZ	\STFPS @-(R). RO BAD.\
8383	071656				EM422:		
	071656	123	124	106		.ASCIZ	\STFPS @-(R) FAILED.\
8384	071702	123	124	106	EM423:	.ASCIZ	'STFPS @-(R) DID NOT DEFFER THE WRITE.'
8385	071750				EM424:		
	071750	123	124	106		.ASCIZ	\STFPS @-(R) TRAPPED TO 4.\
8386							
8387	072002				EM425:		
	072002	123	124	106		.ASCIZ	\STFPS N(R). RO BAD.\
8388	072026				EM426:		
	072026	123	124	106		.ASCIZ	\STFPS N(R) FAILED.\
8389	072051	115	117	122	EM427:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS N(R).'
8390	072122	200	050	102		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8391	072204				EM430:		
	072204	123	124	106		.ASCIZ	\STFPS N(R) TRAPPED TO 4.\
8392	072235	120	103	040	EM431:	.ASCII	'PC BAD AFTER STFPS N(R). BAD CONSTANT USED.'
8393							
8394	072310				EM432:		
	072310	123	124	106		.ASCIZ	\STFPS @N(R). RO BAD.\
8395	072335				EM433:		

	072335	123	124	106		.ASCIZ	\STFPS @N(R) FAILED.\
8396	072361	115	117	122	EM434:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS @N(R).'
8397	072433	200	050	102		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
8398	072515				EM435:		
	072515	123	124	106		.ASCIZ	\STFPS @N(R) TRAPPED TO 4.\
8399	072547	120	103	040	EM436:	.ASCIZ	'PC BAD AFTER STFPS @N(R). BAD CONSTANT USED.'
8400							
8401	072624				EM437:		
	072624	123	124	103		.ASCIZ	\STCDL A,(R)+. R0 BAD.\
8402							
8403	072652				EM440:		
	072652	123	124	103		.ASCIZ	\STCDL A,-(R). R0 BAD.\
8404							
8405	072700	123	124	123	EM361:	.ASCIZ	'STST (R). FPS BAD.'
8406							
8407	072723	116	117	116	EM362:	.ASCII	'NON-RESIDENT MEMORY MANAGEMENT TRAP - '
8408	072771	111	115	120		.ASCIZ	'IMPROPER D-SPACE ACCESS ATTEMPTED'
8409	073033	104	111	106	EM363:	.ASCIZ	'DIFFERENCE BETWEEN SR1 AND CALCULATED'
8410	073101	106	120	120	EM364:	.ASCII	'FPP INSTRUCTION FAILED TO ABORT, NOT '
8411	073146	101	114	114		.ASCIZ	'ALLOWING EXAMINATION OF SR1'
8412	073201	115	117	104	EM365:	.ASCIZ	'MODE 0 INSTRUCTION ABORTED WHEN IT SHOULD NOT HAVE'
8413	073264	106	120	120	EM366:	.ASCIZ	'FPP ACCUMULATOR WAS CHANGED IN THE EXPECTED ABORT.'
8414	073347	107	105	116	EM367:	.ASCIZ	'GENERAL REGISTER WAS CHANGED IN THE EXPECTED ABORT'
8415	073432	106	120	120	EM370:	.ASCIZ	'FPP UNABLE TO RESTORE AN AC'
8416		000000			EM371=0		
8417		000000			EM372=0		
8418		000000			EM373=0		
8419		000000			EM374=0		
8420		000000			EM375=0		
8421		000000			EM376=0		
8422		000000			EM377=0		
8423		000000			EM400=0		
8424							
8425	073466	125	116	105	EM441:	.ASCIZ	'UNEXPECTED FPP TRAP TO 244.'
8426	073522	125	116	105	EM442:	.ASCIZ	'UNEXPECTED CPU TRAP TO 4.'
8427	073554	125	116	105	EM443:	.ASCIZ	'UNEXPECTED CPU TRAP TO 10.'
8428						.EVEN	

```

8430 ;THESE ARE DATA TABLE HEADERS:
8431
8432 073610 040 040 124 DH1: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8433 073650 011 106 120 .ASCIZ <TAB>'FPS.<TAB>'FEC.'
8437 073663
      073663 040 040 124 DH2: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8438 073723 011 107 117 .ASCIZ <TAB>'GOT FPS.<TAB>'EXPECTED FPS.'
8439 073753
      073753 040 040 124 DH3: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8440 074013 011 107 117 .ASCIZ <TAB>'GOT FEC.<TAB>'EXPECTED FEC.'
8441 074043
      074043 040 040 124 DH4: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8442 074103 011 107 117 .ASCIZ <TAB>'GOT RO. <TAB>'EXPECTED RO.'
8443 074132
      074132 040 040 124 DH5: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8444 074172 000 .BYTE 0
8445 074132 DH6=DH5
8446 074043 DH7=DH4
8447 074132 DH10=DH5
8448 074043 DH11=DH4
8449 074132 DH12=DH5
8450 074173 040 040 124 DH13: .ASCIZ ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF TRAP.'
8451 074173 DH14=DH13
8452 074132 DH15=DH5
8453 074233
      074233 040 040 124 DH16: .ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
8454 074273 011 107 117 .ASCIZ <TAB>'GOT PC. <TAB>'EXPECTED PC.'
8455 074173 DH17=DH13
8456 074043 DH20=DH4
8457 074132 DH21=DH5
8458 074132 DH22=DH5
8459 074173 DH23=DH13
8460 074043 DH24=DH4
8461 074132 DH25=DH5
8462 074173 DH26=DH13
8463 074043 DH27=DH4
8464 074132 DH30=DH5
8465 074173 DH31=DH13
8466 074043 DH32=DH4
8467 074132 DH33=DH5
8468 074173 DH34=DH13
8469 074043 DH35=DH4
8470 074132 DH36=DH5
8471 074322 040 040 124 DH37: .ASCIZ ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.<TAB>'GOT FPS.<TAB>'EXPECTED
8472 074322 DH40=DH37
8473 074412 040 040 124 DH41: .ASCIZ ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.<TAB>'FPS.<TAB>'GOT FEC. EXPE
8474 074322 DH42=DH37
8475 074322 DH43=DH37
8476 074322 DH44=DH37
8477 074322 DH45=DH37
8478 074322 DH46=DH37
8479 074322 DH47=DH37
8480 074322 DH50=DH37
8481 074322 DH51=DH37
8482 074322 DH52=DH37
8483 074412 DH53=DH41
8484 074322 DH54=DH37
  
```

8485	074322	DH55=DH37
8486	074322	DH56=DH37
8487	074322	DH57=DH37
8488	074322	DH60=DH37
8489	074322	DH61=DH37
8490	073663	DH62=DH2
8491	073753	DH63=DH3
8492	074132	DH64=DH5
8493	073663	DH65=DH2
8494	074043	DH66=DH4
8495	073663	DH67=DH2
8496	073753	DH70=DH3
8497	073663	DH176=DH2
8498	073753	DH177=DH3
8499	074132	DH71=DH5
8500	073663	DH72=DH2
8501	074173	DH107=DH13
8502	074132	DH73=DH5
8503	074043	DH74=DH4
8504	073663	DH75=DH2
8505	074173	DH76=DH107
8506	074132	DH77=DH5
8507	074043	DH100=DH4
8508	073663	DH101=DH2
8509	074173	DH102=DH107
8510	074132	DH103=DH5
8511	074043	DH104=DH4
8512	073663	DH105=DH2
8513	074173	DH106=DH107
8514	074132	DH110=DH5
8515	074043	DH111=DH4
8516	073663	DH112=DH2
8517	074173	DH113=DH107
8518	074132	DH114=DH5
8519	074043	DH115=DH4
8520	073663	DH116=DH2
8521	074173	DH117=DH107
8522	074132	DH120=DH5
8523	074043	DH121=DH4
8524	073663	DH122=DH2
8525	074173	DH123=DH107
8526	074132	DH124=DH5
8527	074043	DH125=DH4
8528	073663	DH126=DH2
8529	074173	DH127=DH107
8530	074132	DH130=DH5
8531	073663	DH131=DH2
8532	074173	DH132=DH107
8533	074132	DH133=DH5
8534	073663	DH134=DH2
8535	074132	DH135=DH5
8536	074132	DH136=DH5
8537	073663	DH137=DH2
8538	074132	DH140=DH5
8539	074043	DH141=DH4
8540	073663	DH142=DH2
8541	074132	DH143=DH5

8542	074043			DH144=DH4
8543	073663			DH145=DH2
8544	074132			DH146=DH5
8545	074043			DH147=DH4
8546	073663			DH150=DH2
8547	074132			DH151=DH5
8548	074043			DH152=DH4
8549	073663			DH153=DH2
8550	074132			DH154=DH5
8551	074043			DH155=DH4
8552	073663			DH156=DH2
8553	074132			DH157=DH5
8554	074043			DH160=DH4
8555	073663			DH161=DH2
8556	074132			DH162=DH5
8557	073663			DH163=DH2
8558	074233			DH164=DH16
8559	074233			DH215=DH16
8560	074132			DH216=DH5
8561	074043			DH217=DH4
8562	073663			DH220=DH2
8563	074233			DH221=DH16
8564	074132			DH222=DH5
8565	074043			DH223=DH4
8566	073663			DH224=DH2
8567	074322			DH165=DH37
8568	074322			DH166=DH37
8569	074322			DH167=DH37
8570	074322			DH170=DH37
8571	074322			DH171=DH37
8572	074322			DH172=DH37
8573	074412			DH173=DH41
8574	074412			DH174=DH41
8575	074412			DH175=DH41
8576	074322			DH200=DH37
8577	074322			DH201=DH37
8578	074322			DH202=DH37
8579	074322			DH203=DH37
8580	074322			DH204=DH37
8581	074322			DH205=DH37
8582	074322			DH206=DH37
8583	074322			DH207=DH37
8584	074322			DH210=DH37
8585	074322			DH211=DH37
8586	074322			DH212=DH37
8587	074322			DH213=DH37
8588	074322			DH214=DH37
8589				
8590	074043			DH225=DH4
8591	073663			DH226=DH2
8592	074507	040	124	DH227: .ASCIZ ' TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
8593	074043			DH230=DH4
8594	073663			DH231=DH2
8595	074507			DH232=DH227
8596	074043			DH233=DH4
8597	073663			DH234=DH2
8598	074507			DH235=DH227

8599	074043	DH236=DH4
8600	073663	DH237=DH2
8601	074507	DH240=DH227
8602	074043	DH241=DH4
8603	073663	DH242=DH2
8604	074507	DH243=DH227
8605	074043	DH244=DH4
8606	073663	DH245=DH2
8607	074233	DH246=DH16
8608	074507	DH247=DH227
8609	074043	DH250=DH4
8610	073663	DH251=DH2
8611	074233	DH252=DH16
8612	074507	DH253=DH227
8613	074233	DH254=DH16
8614	074507	DH255=DH227
8615	074043	DH256=DH4
8616	073663	DH257=DH2
8617	074322	DH260=DH37
8618	074322	DH261=DH37
8619	074322	DH262=DH37
8620	074322	DH263=DH37
8621	074322	DH264=DH37
8622	074322	DH265=DH37
8623	074322	DH266=DH37
8624	074322	DH267=DH37
8625	074322	DH270=DH37
8626	074322	DH271=DH37
8627	074322	DH272=DH37
8628	074322	DH273=DH37
8629	074322	DH274=DH37
8630	074322	DH275=DH37
8631	074322	DH276=DH37
8632	074322	DH277=DH37
8633	074322	DH300=DH37
8634	074322	DH301=DH37
8635	074322	DH302=DH37
8636	074412	DH303=DH41
8637	074322	DH304=DH37
8638	074322	DH305=DH37
8639	074322	DH306=DH37
8640	074322	DH307=DH37
8641	074322	DH310=DH37
8642	074322	DH311=DH37
8643	074322	DH312=DH37
8644	074322	DH313=DH37
8645	074322	DH314=DH37
8646	074322	DH315=DH37
8647	074322	DH316=DH37
8648	074322	DH317=DH37
8649	074322	DH320=DH37
8650	074322	DH321=DH37
8651	074322	DH322=DH37
8652	074322	DH323=DH37
8653	074412	DH324=DH41
8654	074322	DH325=DH37
8655	074322	DH326=DH37

8656	074322				DH327=DH37	
8657	074322				DH330=DH37	
8658	074322				DH331=DH37	
8659	074322				DH332=DH37	
8660	074322				DH333=DH37	
8661	074322				DH334=DH37	
8662	074322				DH335=DH37	
8663	074322				DH336=DH37	
8664	074322				DH337=DH37	
8665	074322				DH340=DH37	
8666	074322				DH341=DH37	
8667	074322				DH342=DH37	
8668	074322				DH343=DH37	
8669	074322				DH344=DH37	
8670	074322				DH345=DH37	
8671	074322				DH346=DH37	
8672	074322				DH347=DH37	
8673	074322				DH350=DH37	
8674	074173				DH351=DH13	
8675	074322				DH352=DH37	
8676	074322				DH353=DH37	
8677	074322				DH354=DH37	
8678	074322				DH355=DH37	
8679	074043				DH356=DH11	
8680	074547	040	040	124	DH357: .ASCII	' TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
8681	074607	011	107	117	.ASCIZ	<TAB>'GOT FEA.'<TAB>'EXPECTED FEA.'
8682	074173				DH360=DH13	
8683	073663				DH361=DH2	
8684						
8685	074637	040	040	124	DH362: .ASCIZ	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'MMRO'
8686	074702	040	040	124	DH363: .ASCII	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>
8687	074740	123	122	061	.ASCIZ	'SR1'<TAB>'CALCD'<TAB>'EXPECTED'
8688	074763	040	040	124	DH364: .ASCII	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF INSTRUCTION FAILING'
8689	075036	040	124	117	.ASCIZ	' TO ABORT'
8690	075050	040	040	124	DH365: .ASCIZ	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'
8691	075106	040	040	124	DH366: .ASCII	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'AC #'
8692	075150	040	103	110	.ASCIZ	' CHANGED'
8693	075161	040	040	124	DH367: .ASCII	' TEST'<TAB>'PC OF CALL'<TAB>'PC OF ERROR'<TAB>'REG #'
8694	075224	011	122	105	.ASCIZ	<TAB>'RECEIVED'<TAB>'EXPECTED'
8695	075247	040	040	124	DH370: .ASCIZ	' TEST'<TAB>'PC OF CALL'<TAB>'AC #'<TAB>'PC OF ERROR'
8696	000000				DH371=0	
8697	000000				DH372=0	
8698	000000				DH373=0	
8699	000000				DH374=0	
8700	000000				DH375=0	
8701	000000				DH376=0	
8702	000000				DH377=0	
8703	000000				DH400=0	
8704						

8706	074043			DH401=DH4
8707	073663			DH402=DH2
8708	074173			DH403=DH13
8709	074507			DH404=DH227
8710	074043			DH405=DH4
8711	073663			DH406=DH2
8712	074173			DH407=DH13
8713	074507			DH410=DH227
8714	074043			DH411=DH4
8715	073663			DH412=DH2
8716	074173			DH413=DH13
8717	074507			DH414=DH227
8718	074043			DH415=DH4
8719	073663			DH416=DH2
8720	074173			DH417=DH13
8721	074507			DH420=DH227
8722	074043			DH421=DH4
8723	073663			DH422=DH2
8724	074173			DH423=DH13
8725	074507			DH424=DH227
8726	074043			DH425=DH4
8727	073663			DH426=DH2
8728	074173			DH427=DH13
8729	074507			DH430=DH227
8730	074173			DH431=DH13
8731	074043			DH432=DH4
8732	073663			DH433=DH2
8733	074173			DH434=DH13
8734	074507			DH435=DH227
8735	074173			DH436=DH13
8736	074043			DH437=DH4
8737	074043			DH440=DH4
8738	075312	040	124	DH441: .ASCIZ ' TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'<TAB>'FEC.'
8739	075360	040	124	DH442: .ASCIZ ' TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF ERROR.'
8740	075360			DH443=DH442
8741				

```

8743      ;THESE ARE FORMAT SPECIFICATIONS FOR THE DATA TABLES:
8744 075421      004      000      005 DF1:      .BYTE      4,0,5,0,5,0,0
8745 075430      004      000      005 DF2:      .BYTE      4,0,5,0,5,0,5,0
8746      075430      DF3=DF2
8747      075430      DF4=DF2
8748 075440      004      000      005 DF5:      .BYTE      4,0,5,0,5,5,2,5,5,2
8749 075452      004      000      005 DF6:      .BYTE      4,0,5,0
8750      075430      DF7=DF4
8751      075440      DF10=DF5
8752      075430      DF11=DF4
8753 075456      004      000      005 DF12:     .BYTE      4,0,5,0,5,5,3,5,5,3
8754      075452      DF13=DF6
8755      075452      DF14=DF6
8756      075456      DF15=DF12
8757      075430      DF16=DF2
8758      075452      DF17=DF6
8759      075430      DF20=DF2
8760      075456      DF21=DF12
8761      075456      DF22=DF12
8762      075452      DF23=DF6
8763      075430      DF24=DF2
8764      075456      DF25=DF12
8765      075452      DF26=DF6
8766      075430      DF27=DF2
8767      075456      DF30=DF12
8768      075452      DF31=DF6
8769      075430      DF32=DF2
8770      075456      DF33=DF12
8771      075452      DF34=DF6
8772      075430      DF35=DF2
8773      075456      DF36=DF12
8774 075470      004      000      005 DF37:     .BYTE      4,0,5,0,5,0,5,0,5,5,3,5,5,3,5,5,3
8775      075470      DF40=DF37
8776 075511      004      000      005 DF41:     .BYTE      4,0,5,0,5,0,0,0,5,5,3,5,5,3,5,5,3
8777      075470      DF42=DF37
8778      075470      DF43=DF37
8779      075470      DF44=DF37
8780      075470      DF45=DF37
8781      075470      DF46=DF37
8782      075470      DF47=DF37
8783      075470      DF50=DF37
8784      075470      DF51=DF37
8785      075470      DF52=DF37
8786      075470      DF53=DF37
8787      075470      DF54=DF37
8788      075470      DF55=DF37
8789      075470      DF56=DF37
8790      075470      DF57=DF37
8791      075470      DF60=DF37
8792      075470      DF61=DF37
8793      075430      DF62=DF2
8794      075430      DF63=DF2
8795      075440      DF64=DF5
8796      075430      DF65=DF2
8797      075430      DF66=DF2
8798      075430      DF67=DF2
8799      075430      DF70=DF2
  
```

8800	075430			DF176=DF2
8801	075430			DF177=DF2
8802	075532	004	000 005	DF71: .BYTE 4,0,5,0,5,5,3,5,5,3,5,5,3
8803	075430			DF72=DF2
8804	075452			DF107=DF6
8805	075532			DF73=DF71
8806	075430			DF74=DF2
8807	075430			DF75=DF2
8808	075452			DF76=DF6
8809	075532			DF77=DF71
8810	075430			DF100=DF2
8811	075430			DF101=DF2
8812	075452			DF102=DF6
8813	075532			DF103=DF71
8814	075430			DF104=DF2
8815	075430			DF105=DF2
8816	075452			DF106=DF6
8817	075532			DF110=DF71
8818	075430			DF111=DF2
8819	075430			DF112=DF2
8820	075452			DF113=DF6
8821	075532			DF114=DF71
8822	075430			DF115=DF2
8823	075430			DF116=DF2
8824	075452			DF117=DF6
8825	075532			DF120=DF71
8826	075430			DF121=DF2
8827	075430			DF122=DF2
8828	075452			DF123=DF6
8829	075532			DF124=DF71
8830	075430			DF125=DF2
8831	075430			DF126=DF2
8832	075452			DF127=DF6
8833	075532			DF130=DF71
8834	075430			DF131=DF2
8835	075452			DF132=DF6
8836	075532			DF133=DF71
8837	075430			DF134=DF2
8838	075456			DF135=DF12
8839	075456			DF136=DF12
8840	075430			DF137=DF2
8841	075456			DF140=DF12
8842	075430			DF141=DF2
8843	075430			DF142=DF2
8844	075456			DF143=DF12
8845	075430			DF144=DF2
8846	075430			DF145=DF2
8847	075456			DF146=DF12
8848	075430			DF147=DF2
8849	075430			DF150=DF2
8850	075456			DF151=DF12
8851	075430			DF152=DF2
8852	075430			DF153=DF2
8853	075456			DF154=DF12
8854	075430			DF155=DF2
8855	075430			DF156=DF2
8856	075456			DF157=DF12

8857	075430			DF160=DF2
8858	075430			DF161=DF2
8859	075456			DF162=DF12
8860	075430			DF163=DF2
8861	075430			DF164=DF2
8862	075430			DF215=DF2
8863	075456			DF216=DF12
8864	075430			DF217=DF2
8865	075430			DF220=DF2
8866	075430			DF221=DF2
8867	075456			DF222=DF12
8868	075430			DF223=DF2
8869	075430			DF224=DF2
8870	075470			DF165=DF37
8871	075470			DF166=DF37
8872	075470			DF167=DF37
8873	075470			DF170=DF37
8874	075470			DF171=DF37
8875	075470			DF172=DF37
8876	075511			DF173=DF41
8877	075511			DF174=DF41
8878	075511			DF175=DF41
8879	075470			DF200=DF37
8880	075470			DF201=DF37
8881	075470			DF202=DF37
8882	075470			DF203=DF37
8883	075470			DF204=DF37
8884	075470			DF205=DF37
8885	075470			DF206=DF37
8886	075470			DF207=DF37
8887	075470			DF210=DF37
8888	075470			DF211=DF37
8889	075470			DF212=DF37
8890	075470			DF213=DF37
8891	075470			DF214=DF37
8892	075547	000	005	DF225: .BYTE 4,0,5,0,5,0,5,0
8893	075547			DF226=DF225
8894	075557	000	005	DF227: .BYTE 4,0,5,0
8895	075547			DF230=DF225
8896	075547			DF231=DF225
8897	075557			DF232=DF227
8898	075547			DF233=DF225
8899	075547			DF234=DF225
8900	075557			DF235=DF227
8901	075547			DF236=DF225
8902	075547			DF237=DF225
8903	075557			DF240=DF227
8904	075547			DF241=DF225
8905	075547			DF242=DF225
8906	075557			DF243=DF227
8907	075547			DF244=DF225
8908	075547			DF245=DF225
8909	075547			DF246=DF225
8910	075557			DF247=DF227
8911	075547			DF250=DF225
8912	075547			DF251=DF225
8913	075547			DF252=DF225

8914	075557			DF253=DF227	
8915	075547			DF254=DF225	
8916	075557			DF255=DF227	
8917	075547			DF256=DF225	
8918	075547			DF257=DF225	
8919					
8920	075563	004	000	005	DF260: .BYTE 4,0,5,0,5,0,5,0,5,5,2,5,5,2,5,5,2
8921	075563				DF261=DF260
8922	075563				DF262=DF260
8923	075563				DF263=DF260
8924	075563				DF264=DF260
8925	075563				DF265=DF260
8926	075563				DF266=DF260
8927	075563				DF267=DF260
8928	075563				DF270=DF260
8929	075563				DF271=DF260
8930	075563				DF272=DF260
8931					
8932	075604	004	000	005	DF273: .BYTE 4,0,5,0,5,0,5,0,5,5,2,5,5,3,5,5,3
8933	075604				DF274=DF273
8934	075604				DF275=DF273
8935	075604				DF276=DF273
8936	075604				DF277=DF273
8937	075604				DF300=DF273
8938	075625	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
8939	075625				DF302=DF301
8940	075651	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
8941	075625				DF304=DF301
8942	075625				DF305=DF301
8943	075625				DF306=DF301
8944	075625				DF307=DF301
8945	075625				DF310=DF301
8946	075625				DF311=DF301
8947	075625				DF312=DF301
8948	075625				DF313=DF301
8949	075625				DF314=DF301
8950	075625				DF315=DF301
8951	075625				DF316=DF301
8952	075625				DF317=DF301
8953	075625				DF320=DF301
8954	075625				DF321=DF301
8955					
8956	075675	004	000	005	DF322: .BYTE 4,0,5,0,5,0,5,0,5,5,3,5,5,2,5,5,2
8957	075675				DF323=DF322
8958	075716	004	000	005	DF324: .BYTE 4,0,5,0,5,0,0,0,5,5,3,5,5,2,5,5,2
8959	075675				DF325=DF322
8960	075675				DF326=DF322
8961	075675				DF327=DF322
8962	075675				DF330=DF322
8963	075675				DF331=DF322
8964	075675				DF332=DF322
8965	075675				DF333=DF322
8966	075675				DF334=DF322
8967	075675				DF335=DF322
8968	075675				DF336=DF322
8969	075675				DF337=DF322
8970	075675				DF340=DF322

8971	075675			DF341=DF322	
8972	075675			DF342=DF322	
8973	075675			DF343=DF322	
8974	075675			DF344=DF322	
8975	075675			DF345=DF322	
8976	075675			DF346=DF322	
8977					
8978	075737	004	000	005	DF347: .BYTE 4,0,5,0,5,0,5,0,5,5,3,5,5,0,5,5,0
8979	075737				DF350=DF347
8980	075557				DF351=DF227
8981	075737				DF352=DF347
8982	075737				DF353=DF347
8983	075737				DF354=DF347
8984	075737				DF355=DF347
8985	075547				DF356=DF225
8986	075547				DF357=DF225
8987	075557				DF360=DF227
8988	075547				DF361=DF225
8989					
8990	075760	004	000	005	DF362: .BYTE 4,0,5,0,5,0
8991	075766	004	000	005	DF363: .BYTE 4,0,5,0,5,0,0
8992	075452				DF364=DF6
8993	075452				DF365=DF6
8994	075775	004	000	005	DF366: .BYTE 4,0,5,0,5,0,5,5,0,0,0,0,5,5,0,0,0,0
8995	076017	004	000	005	DF367: .BYTE 4,0,5,0,5,0,5,0
8996	076027	004	000	005	DF370: .BYTE 4,0,5,0,0,5,5,0,0,0,0,0,5,5,0,0,0,0
8997	000000				DF371=0
8998	000000				DF372=0
8999	000000				DF373=0
9000	000000				DF374=0
9001	000000				DF375=0
9002	000000				DF376=0
9003	000000				DF377=0
9004	000000				DF400=0
9005					
9006	075547				DF401=DF225
9007	075547				DF402=DF225
9008	075557				DF403=DF227
9009	075557				DF404=DF227
9010	075547				DF405=DF225
9011	075547				DF406=DF225
9012	075557				DF407=DF227
9013	075557				DF410=DF227
9014	075547				DF411=DF225
9015	075547				DF412=DF225
9016	075557				DF413=DF227
9017	075557				DF414=DF227
9018	075547				DF415=DF225
9019	075547				DF416=DF225
9020	075557				DF417=DF227
9021	075557				DF420=DF227
9022	075547				DF421=DF225
9023	075547				DF422=DF225
9024	075557				DF423=DF227
9025	075557				DF424=DF227
9026	075547				DF425=DF225
9027	075547				DF426=DF225

9028	075557			DF427=DF227	
9029	075557			DF430=DF227	
9030	075557			DF431=DF227	
9031	075547			DF432=DF225	
9032	075547			DF433=DF225	
9033	075557			DF434=DF227	
9034	075557			DF435=DF227	
9035	075557			DF436=DF227	
9036	075547			DF437=DF225	
9037	075547			DF440=DF225	
9038	076050	004	000	005	DF441: .BYTE 4,0,5,0,5,0
9039	076050				DF442=DF441
9040	076050				DF443=DF441
9041					
9042					

```

9044 .EVEN
9045
9046 ;THESE ARE THE ERROR MESSAGE DATA TABLES:
9047
9048 076056 001232 001234 046475 DT1: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TMP4,0
9049 076076 001232 001234 046475 DT2: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,$TAB,$TMP5,0
9050 076120 001232 001234 046475 DT3: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP6,0
9051 076142 001232 001234 046475 DT4: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP3,0
9052 076164 001232 001234 046475 DT5: .WORD $TMP0,$TMP1,$TAB,$TMP2,$CRLF,$MS1,$TMP3
9053 076202 001313 046515 001242 .WORD $CRLF,$MS2,$TMP4,0
9054 076212 001232 001234 046475 DT6: .WORD $TMP0,$TMP1,$TAB,$TMP2,0
9055 076142 DT7=DT4
9056 076164 DT10=DT5
9057 076142 DT11=DT4
9058 076164 DT12=DT5
9059 076212 DT13=DT6
9060 076212 DT14=DT6
9061 076164 DT15=DT5
9062 076224 001232 001234 046475 DT16: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP4,$TAB,$TMP3,0
9063 076212 DT17=DT6
9064 076224 DT20=DT16
9065 076164 DT21=DT5
9066 076164 DT22=DT5
9067 076212 DT23=DT6
9068 076224 DT24=DT16
9069 076164 DT25=DT5
9070 076212 DT26=DT6
9071 076224 DT27=DT16
9072 076164 DT30=DT5
9073 076212 DT31=DT6
9074 076224 DT32=DT16
9075 076164 DT33=DT5
9076 076212 DT34=DT6
9077 076224 DT35=DT16
9078 076164 DT36=DT5
9079 076246 001232 001234 046475 DT37: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10,$CRLF
9080 076270 046555 001240 001313 .WORD $MS4,$TMP3,$CRLF,$MS1,$TMP4,$CRLF,$MS2,$TMP5,0
9081 076246 DT40=DT37
9082 076312 001232 001234 046475 DT41: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TMP11,$TMP12
9083 076332 001313 046555 001240 .WORD $CRLF,$MS4,$TMP3,$CRLF,$MS1,$TMP4,$CRLF,$MS2,$TMP5,0
9084 076246 DT42=DT37
9085 076246 DT43=DT37
9086 076246 DT44=DT37
9087 076246 DT45=DT37
9088 076246 DT46=DT37
9089 076246 DT47=DT37
9090 076246 DT50=DT37
9091 076246 DT51=DT37
9092 076246 DT52=DT37
9093 076312 DT53=DT41
9094 076246 DT54=DT37
9095 076246 DT55=DT37
9096 076246 DT56=DT37
9097 076246 DT57=DT37
9098 076246 DT60=DT37
9099 076246 DT61=DT37
9100 076224 DT62=DT16
  
```

```

9101          076224          DT63=DT16
9102          076164          DT64=DT5
9103          076224          DT65=DT16
9104          076142          DT66=DT4
9105          076142          DT67=DT4
9106          076142          DT70=DT4
9107          076142          DT176=DT4
9108          076142          DT177=DT4
9109  076356  001232  001234  046475  DT71:  .WORD  $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS3,$TMP3,$CRLF,MS1
9110  076400  001244  001313  046515      .WORD  $TMP5,$CRLF,MS2,$TMP4,0
9111          076142          DT72=DT4
9112          076212          DT107=DT6
9113  076412  001232  001234  046475  DT73:  .WORD  $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS4,$TMP4
9114  076430  001313  046477  001244      .WORD  $CRLF,MS1,$TMP5,$CRLF,MS2,$TMP3,0
9115          076142          DT74=DT4
9116          076076          DT75=DT2
9117          076212          DT76=DT6
9118          076412          DT77=DT73
9119          076142          DT100=DT4
9120          076076          DT101=DT2
9121          076212          DT102=DT6
9122          076412          DT103=DT73
9123          076142          DT104=DT4
9124          076076          DT105=DT2
9125          076212          DT106=DT6
9126          076412          DT110=DT73
9127          076142          DT111=DT4
9128          076076          DT112=DT2
9129          076212          DT113=DT6
9130          076412          DT114=DT73
9131          076142          DT115=DT4
9132          076076          DT116=DT2
9133          076212          DT117=DT6
9134          076412          DT120=DT73
9135          076142          DT121=DT4
9136          076076          DT122=DT2
9137          076212          DT123=DT6
9138          076412          DT124=DT73
9139          076142          DT125=DT4
9140          076076          DT126=DT2
9141          076212          DT127=DT6
9142          076412          DT130=DT73
9143          076076          DT131=DT2
9144          076212          DT132=DT6
9145          076412          DT133=DT73
9146          076076          DT134=DT2
9147          076164          DT135=DT5
9148          076164          DT136=DT5
9149          076224          DT137=DT16
9150          076164          DT140=DT5
9151          076142          DT141=DT4
9152          076142          DT142=DT4
9153          076164          DT143=DT5
9154          076142          DT144=DT4
9155          076142          DT145=DT4
9156          076164          DT146=DT5
9157          076142          DT147=DT4
    
```

9158	076142	DT150=DT4
9159	076164	DT151=DT5
9160	076142	DT152=DT4
9161	076142	DT153=DT4
9162	076164	DT154=DT5
9163	076142	DT155=DT4
9164	076142	DT156=DT4
9165	076164	DT157=DT5
9166	076142	DT160=DT4
9167	076142	DT161=DT4
9168	076164	DT162=DT5
9169	076142	DT163=DT4
9170	076142	DT164=DT4
9171	076142	DT215=DT4
9172	076164	DT216=DT5
9173	076142	DT217=DT4
9174	076142	DT220=DT4
9175	076142	DT221=DT4
9176	076164	DT222=DT5
9177	076142	DT223=DT4
9178	076142	DT224=DT4
9179	076246	DT165=DT37
9180	076246	DT166=DT37
9181	076246	DT167=DT37
9182	076246	DT170=DT37
9183	076246	DT171=DT37
9184	076246	DT172=DT37
9185	076312	DT173=DT41
9186	076312	DT174=DT41
9187	076312	DT175=DT41
9188	076246	DT200=DT37
9189	076246	DT201=DT37
9190	076246	DT202=DT37
9191	076246	DT203=DT37
9192	076246	DT204=DT37
9193	076246	DT205=DT37
9194	076246	DT206=DT37
9195	076246	DT207=DT37
9196	076246	DT210=DT37
9197	076246	DT211=DT37
9198	076246	DT212=DT37
9199	076246	DT213=DT37
9200	076246	DT214=DT37
9201	076142	DT225=DT4
9202	076142	DT226=DT4
9203	076212	DT227=DT6
9204	076142	DT230=DT4
9205	076142	DT231=DT4
9206	076212	DT232=DT6
9207	076142	DT233=DT4
9208	076142	DT234=DT4
9209	076212	DT235=DT6
9210	076142	DT236=DT4
9211	076142	DT237=DT4
9212	076212	DT240=DT6
9213	076142	DT241=DT4
9214	076142	DT242=DT4

9215	076212			DT243=DT6	
9216	076142			DT244=DT4	
9217	076142			DT245=DT4	
9218	076142			DT246=DT4	
9219	076212			DT247=DT6	
9220	076142			DT250=DT4	
9221	076142			DT251=DT4	
9222	076142			DT252=DT4	
9223	076212			DT253=DT6	
9224	076142			DT254=DT4	
9225	076212			DT255=DT6	
9226	076142			DT256=DT4	
9227	076142			DT257=DT4	
9228	076246			DT260=DT37	
9229	076246			DT261=DT37	
9230	076246			DT262=DT37	
9231	076246			DT263=DT37	
9232	076246			DT264=DT37	
9233	076246			DT265=DT37	
9234	076246			DT266=DT37	
9235	076246			DT267=DT37	
9236	076246			DT270=DT37	
9237	076246			DT271=DT37	
9238	076246			DT272=DT37	
9239	076246			DT273=DT37	
9240	076246			DT274=DT37	
9241	076246			DT275=DT37	
9242	076246			DT276=DT37	
9243	076246			DT277=DT37	
9244	076246			DT300=DT37	
9245	076446	001232	001234	046475	DT301: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TAB,\$TMP10
9246	076466	001313	046537	001240	.WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4
9247	076502	001313	046477	001246	.WORD \$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0
9248	076446				DT302=DT301
9249	076520	001232	001234	046475	DT303: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TMP11,\$TMP12
9250	076540	001313	046537	001240	.WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4
9251	076554	001313	046477	001246	.WORD \$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0
9252	076446				DT304=DT301
9253	076446				DT305=DT301
9254	076446				DT306=DT301
9255	076446				DT307=DT301
9256	076446				DT310=DT301
9257	076446				DT311=DT301
9258	076446				DT312=DT301
9259	076446				DT313=DT301
9260	076446				DT314=DT301
9261	076446				DT315=DT301
9262	076446				DT316=DT301
9263	076446				DT317=DT301
9264	076446				DT320=DT301
9265	076446				DT321=DT301
9266	076572	001232	001234	046475	DT322: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TAB,\$TMP10
9267	076612	001313	046537	001240	.WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0
9268	076572				DT323=DT322
9269	076636	001232	001234	046475	DT324: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TMP11,\$TMP12
9270	076656	001313	046537	001240	.WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0
9271	076572				DT325=DT322

```

9272      076572      DT326=DT322
9273      076572      DT327=DT322
9274      076572      DT330=DT322
9275      076572      DT331=DT322
9276      076572      DT332=DT322
9277      076572      DT333=DT322
9278      076572      DT334=DT322
9279      076572      DT335=DT322
9280      076572      DT336=DT322
9281      076572      DT337=DT322
9282      076572      DT340=DT322
9283      076572      DT341=DT322
9284      076572      DT342=DT322
9285      076572      DT343=DT322
9286      076572      DT344=DT322
9287      076572      DT345=DT322
9288      076572      DT346=DT322
9289      076572      DT347=DT322
9290      076572      DT350=DT322
9291      076212      DT351=DT6
9292      076572      DT352=DT322
9293      076572      DT353=DT322
9294      076572      DT354=DT322
9295      076572      DT355=DT322
9296      076076      DT356=DT2
9297      076120      DT357=DT3
9298      076212      DT360=DT6
9299      076446      DT361=DT302
9300 076702 001232 001234 046475 DT362: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP3,0
9301 076720 001232 001234 046475 DT363: .WORD $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP2,$TMP3,0
9302 076740 001232 001234 046475 DT364: .WORD $TMP0,$TMP1,$TAB,$TMP13,0
9303      076212      DT365=DT6
9304 076752 001232 001234 046475 DT366: .WORD $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP2
9305 076766 001313 046515 001240 .WORD $CRLF,$MS2,$TMP3,$TMP4,$TMP6,$TMP7
9306 077002 001313 046477 001252 .WORD $CRLF,$MS1,$TMP10,$TMP11,$TMP12,$TMP21,0
9307 077020 001232 001234 046475 DT367: .WORD $TMP0,$TMP1,$TAB,$TMP13,$TAB,$TMP6,$TAB,$TMP3,0
9308 077042 001232 001234 046475 DT370: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TMP13
9309 077054 001313 046634 001240 .WORD $CRLF,$MS21,$TMP3,$TMP4,$TMP6,$TMP7
9310 077070 001313 046622 001252 .WORD $CRLF,$MS20,$TMP10,$TMP11,$TMP12,$TMP21,0
9311      000000      DT371=0
9312      000000      DT372=0
9313      000000      DT373=0
9314      000000      DT374=0
9315      000000      DT375=0
9316      000000      DT376=0
9317      000000      DT377=0
9318      000000      DT400=0
9319
9320      076142      DT401=DT4
9321      076142      DT402=DT4
9322      076212      DT403=DT6
9323      076212      DT404=DT6
9324      076142      DT405=DT4
9325      076142      DT406=DT4
9326      076212      DT407=DT6
9327      076212      DT410=DT6
9328      076142      DT411=DT4

```

9329	076142			DT412=DT4	
9330	076212			DT413=DT6	
9331	076212			DT414=DT6	
9332	076142			DT415=DT4	
9333	076142			DT416=DT4	
9334	076212			DT417=DT6	
9335	076212			DT420=DT6	
9336	076142			DT421=DT4	
9337	076142			DT422=DT4	
9338	076212			DT423=DT6	
9339	076212			DT424=DT6	
9340	076142			DT425=DT4	
9341	076142			DT426=DT4	
9342	076212			DT427=DT6	
9343	076212			DT430=DT6	
9344	076212			DT431=DT6	
9345	076142			DT432=DT4	
9346	076142			DT433=DT4	
9347	076212			DT434=DT6	
9348	076212			DT435=DT6	
9349	076212			DT436=DT6	
9350	076142			DT437=DT4	
9351	076142			DT440=DT4	
9352	077106	001232	001234 046475	DT441: .WORD	\$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP3,0
9353	077124	001232	001234 046475	DT442: .WORD	\$TMP0,\$TMP1,\$TAB,\$TMP2,0
9354		077124		DT443=DT442	
9355					
9356					
9357					
9358					

9360
9361 ;12345
9362 000001 .END

SYMBOL TABLE

AABBF0	013602	AMSGLG=	000000	CCB2	013654	DF133	= 075532	DF215	= 075430
AABDON	013632	AMSGTY=	000000	CKSWR =	104406	DF134	= 075430	DF216	= 075456
AABTP1	013612	AMTYP1=	000000	CNT =	000444	DF135	= 075456	DF217	= 075430
AABTP2	013622	AMTYP2=	000000	CPSPUR	046310	DF136	= 075456	DF22	= 075456
AAB1	013424	AMTYP3=	000000	CPTWO	046330	DF137	= 075430	DF220	= 075430
AAB2	013526	AMTYP4=	000000	CR =	J00015	DF14	= 075452	DF221	= 075430
AAB3	013546	APASS =	000000	CRLF =	000200	DF140	= 075456	DF222	= 075456
AAB4	013564	APRIOR=	000000	DATA =	117760	DF141	= 075430	DF223	= 075430
AACDON	023754	APTCSU=	000040	DDBBF0	014230	DF142	= 075430	DF224	= 075430
AACTP1	023660	APTENV=	000001	DDBDON	014250	DF143	= 075456	DF225	= 075547
AAC1	023600	APTSIZ=	0J0200	DDBTP1	014210	DF144	= 075430	DF226	= 075547
AAC10	023664	APTSPO=	000100	DDBTP2	014220	DF145	= 075430	DF227	= 075557
AAC11	023702	ASWREG=	000000	DDBTP3	014240	DF146	= 075456	DF23	= 075452
AAC2	023634	ATESTN=	000000	DDB1	014036	DF147	= 075430	DF230	= 075547
AAC20	023720	AUNIT =	000000	DDB2	014102	DF15	= 075456	DF231	= 075547
ABASE =	000000	AUSWR =	000000	DDB5	014144	DF150	= 075430	DF232	= 075557
ACDW1 =	000000	AVECT1=	000000	DDB6	014172	DF151	= 075456	DF233	= 075547
ACDW2 =	000000	AVECT2=	000000	DDCDON	024326	DF152	= 075430	DF234	= 075547
ACPUOP=	000000	BBBDON	024132	DDCTP1	024226	DF153	= 075430	DF235	= 075557
ACO =	%000000	BBCTP1	024040	DDC1	024136	DF154	= 075456	DF236	= 075547
AC1 =	%000001	BBC1	023760	DDC10	024240	DF155	= 075430	DF237	= 075547
AC2 =	%000002	BBC10	024044	DDC11	024256	DF156	= 075430	DF24	= 075430
AC3 =	%000003	BBC11	024062	DDC2	024174	DF157	= 075456	DF240	= 075557
AC4 =	%000004	BBC2	024014	DDC20	024274	DF16	= 075430	DF241	= 075547
AC5 =	%000005	BBC20	024100	DDISP =	177570	DF160	= 075430	DF242	= 075547
AC6 =	%000006	BIT0 =	000001	DF1	075421	DF161	= 075430	DF243	= 075557
AC7 =	%000007	BIT00 =	000001	DF10 =	075440	DF162	= 075456	DF244	= 075547
ADDW0 =	000000	BIT01 =	000002	DF100 =	075430	DF163	= 075430	DF245	= 075547
ADDW1 =	000000	BIT02 =	000004	DF101 =	075430	DF164	= 075430	DF246	= 075547
ADDW10=	000000	BIT03 =	000010	DF102 =	075452	DF165	= 075470	DF247	= 075557
ADDW11=	000000	BIT04 =	000020	DF103 =	075532	DF166	= 075470	DF25	= 075456
ADDW12=	000000	BIT05 =	000040	DF104 =	075430	DF167	= 075470	DF250	= 075547
ADDW13=	000000	BIT06 =	000100	DF105 =	075430	DF17	= 075452	DF251	= 075547
ADDW14=	000000	BIT07 =	000200	DF106 =	075452	DF170	= 075470	DF252	= 075547
ADDW15=	000000	BIT08 =	000400	DF107 =	075452	DF171	= 075470	DF253	= 075557
ADDW2 =	000000	BIT09 =	001000	DF11	= 075430	DF172	= 075470	DF254	= 075547
ADDW3 =	000000	BIT1 =	000002	DF110 =	075532	DF173	= 075511	DF255	= 075557
ADDW4 =	000000	BIT10 =	002000	DF111 =	075430	DF174	= 075511	DF256	= 075547
ADDW5 =	000000	BIT11 =	004000	DF112 =	075430	DF175	= 075511	DF257	= 075547
ADDW6 =	000000	BIT12 =	010000	DF113 =	075452	DF176	= 075430	DF26	= 075452
ADDW7 =	000000	BIT13 =	020000	DF114 =	075532	DF177	= 075430	DF260	= 075563
ADDW8 =	000000	BIT14 =	040000	DF115 =	075430	DF2	= 075430	DF261	= 075563
ADDW9 =	000000	BIT15 =	100000	DF116 =	075430	DF20	= 075430	DF262	= 075563
ADEVCT=	000000	BIT2 =	000004	DF117 =	075452	DF200	= 075470	DF263	= 075563
ADEVN =	000000	BIT3 =	000010	DF12	= 075456	DF201	= 075470	DF264	= 075563
AENV =	000000	BIT4 =	000020	DF120 =	075532	DF202 =	075470	DF265	= 075563
AENVN =	000000	BIT5 =	000040	DF121 =	075430	DF203 =	075470	DF266	= 075563
AFATAL=	000000	BIT6 =	000100	DF122 =	075430	DF204 =	075470	DF267	= 075563
AMADR1=	000000	BIT7 =	000200	DF123 =	075452	DF205 =	075470	DF27	= 075430
AMADR2=	000000	BIT8 =	000400	DF124 =	075532	DF206 =	075470	DF270	= 075563
AMADR3=	000000	BIT9 =	001000	DF125 =	075430	DF207 =	075470	DF271	= 075563
AMADR4=	000000	BPTVEC=	000014	DF126 =	075430	DF21	= 075456	DF272	= 075563
AMAMS1=	000000	BYTABL	042460	DF127 =	075452	DF210	= 075470	DF273	= 075604
AMAMS2=	000000	CCBDON	013732	DF13	= 075452	DF211	= 075470	DF274	= 075604
AMAMS3=	000000	CCB1	013636	DF130 =	075532	DF212	= 075470	DF275	= 075604
AMAMS4=	000000	CCB10	013700	DF131 =	075430	DF213	= 075470	DF276	= 075604
AMSGAD=	000000	CCB15	013716	DF132 =	075452	DF214	= 075470	DF277	= 075604

DF 3 = 075430	DF 361 = 075547	DF 443 = 076050	DH126 = 073663	DH21 = 074132
DF 30 = 075456	DF 362 = 075760	DF 45 = 075470	DH127 = 074173	DH210 = 074322
DF 300 = 075604	DF 363 = 075766	DF 46 = 075470	DH13 = 074173	DH211 = 074322
DF 301 = 075625	DF 364 = 075452	DF 47 = 075470	DH130 = 074132	DH212 = 074322
DF 302 = 075625	DF 365 = 075452	DF 5 = 075440	DH131 = 073663	DH213 = 074322
DF 303 = 075651	DF 366 = 075775	DF 50 = 075470	DH132 = 074173	DH214 = 074322
DF 304 = 075625	DF 367 = 076017	DF 51 = 075470	DH133 = 074132	DH215 = 074233
DF 305 = 075625	DF 37 = 075470	DF 52 = 075470	DH134 = 073663	DH216 = 074132
DF 306 = 075625	DF 370 = 076027	DF 53 = 075470	DH135 = 074132	DH217 = 074043
DF 307 = 075625	DF 371 = 000000	DF 54 = 075470	DH136 = 074132	DH22 = 074132
DF 31 = 075452	DF 372 = 000000	DF 55 = 075470	DH137 = 073663	DH220 = 073663
DF 310 = 075625	DF 373 = 000000	DF 56 = 075470	DH14 = 074173	DH221 = 074233
DF 311 = 075625	DF 374 = 000000	DF 57 = 075470	DH140 = 074132	DH222 = 074132
DF 312 = 075625	DF 375 = 000000	DF 6 = 075452	DH141 = 074043	DH223 = 074043
DF 313 = 075625	DF 376 = 000000	DF 60 = 075470	DH142 = 073663	DH224 = 073663
DF 314 = 075625	DF 377 = 000000	DF 61 = 075470	DH143 = 074132	DH225 = 074043
DF 315 = 075625	DF 4 = 075430	DF 62 = 075430	DH144 = 074043	DH226 = 073663
DF 316 = 075625	DF 40 = 075470	DF 63 = 075430	DH145 = 073663	DH227 = 074507
DF 317 = 075625	DF 400 = 000000	DF 64 = 075440	DH146 = 074132	DH23 = 074173
DF 32 = 075430	DF 401 = 075547	DF 65 = 075430	DH147 = 074043	DH230 = 074043
DF 320 = 075625	DF 402 = 075547	DF 66 = 075430	DH15 = 074132	DH231 = 073663
DF 321 = 075625	DF 403 = 075557	DF 67 = 075430	DH150 = 073663	DH232 = 074507
DF 322 = 075675	DF 404 = 075557	DF 7 = 075430	DH151 = 074132	DH233 = 074043
DF 323 = 075675	DF 405 = 075547	DF 70 = 075430	DH152 = 074043	DH234 = 073663
DF 324 = 075716	DF 406 = 075547	DF 71 = 075532	DH153 = 073663	DH235 = 074507
DF 325 = 075675	DF 407 = 075557	DF 72 = 075430	DH154 = 074132	DH236 = 074043
DF 326 = 075675	DF 41 = 075511	DF 73 = 075532	DH155 = 074043	DH237 = 073663
DF 327 = 075675	DF 410 = 075557	DF 74 = 075430	DH156 = 073663	DH24 = 074043
DF 33 = 075456	DF 411 = 075547	DF 75 = 075430	DH157 = 074132	DH240 = 074507
DF 330 = 075675	DF 412 = 075547	DF 76 = 075452	DH16 = 074233	DH241 = 074043
DF 331 = 075675	DF 413 = 075557	DF 77 = 075532	DH160 = 074043	DH242 = 073663
DF 332 = 075675	DF 414 = 075557	DH1 = 073610	DH161 = 073663	DH243 = 074507
DF 333 = 075675	DF 415 = 075547	DH10 = 074132	DH162 = 074132	DH244 = 074043
DF 334 = 075675	DF 416 = 075547	DH100 = 074043	DH163 = 073663	DH245 = 073663
DF 335 = 075675	DF 417 = 075557	DH101 = 073663	DH164 = 074233	DH246 = 074233
DF 336 = 075675	DF 42 = 075470	DH102 = 074173	DH165 = 074322	DH247 = 074507
DF 337 = 075675	DF 420 = 075557	DH103 = 074132	DH166 = 074322	DH25 = 074132
DF 34 = 075452	DF 421 = 075547	DH104 = 074043	DH167 = 074322	DH250 = 074043
DF 340 = 075675	DF 422 = 075547	DH105 = 073663	DH17 = 074173	DH251 = 073663
DF 341 = 075675	DF 423 = 075557	DH106 = 074173	DH170 = 074322	DH252 = 074233
DF 342 = 075675	DF 424 = 075557	DH107 = 074173	DH171 = 074322	DH253 = 074507
DF 343 = 075675	DF 425 = 075547	DH11 = 074043	DH172 = 074322	DH254 = 074233
DF 344 = 075675	DF 426 = 075547	DH110 = 074132	DH173 = 074412	DH255 = 074507
DF 345 = 075675	DF 427 = 075557	DH111 = 074043	DH174 = 074412	DH256 = 074043
DF 346 = 075675	DF 43 = 075470	DH112 = 073663	DH175 = 074412	DH257 = 073663
DF 347 = 075737	DF 430 = 075557	DH113 = 074173	DH176 = 073663	DH26 = 074173
DF 35 = 075430	DF 431 = 075557	DH114 = 074132	DH177 = 073753	DH260 = 074322
DF 350 = 075737	DF 432 = 075547	DH115 = 074043	DH2 = 073663	DH261 = 074322
DF 351 = 075557	DF 433 = 075547	DH116 = 073663	DH20 = 074043	DH262 = 074322
DF 352 = 075737	DF 434 = 075557	DH117 = 074173	DH200 = 074322	DH263 = 074322
DF 353 = 075737	DF 435 = 075557	DH12 = 074132	DH201 = 074322	DH264 = 074322
DF 354 = 075737	DF 436 = 075557	DH120 = 074132	DH202 = 074322	DH265 = 074322
DF 355 = 075737	DF 437 = 075547	DH121 = 074043	DH203 = 074322	DH266 = 074322
DF 356 = 075547	DF 44 = 075470	DH122 = 073663	DH204 = 074322	DH267 = 074322
DF 357 = 075547	DF 440 = 075547	DH123 = 074173	DH205 = 074322	DH27 = 074043
DF 36 = 075456	DF 441 = 076050	DH124 = 074132	DH206 = 074322	DH270 = 074322
DF 360 = 075557	DF 442 = 076050	DH125 = 074043	DH207 = 074322	DH271 = 074322

DH272 = 074322
 DH273 = 074322
 DH274 = 074322
 DH275 = 074322
 DH276 = 074322
 DH277 = 074322
 DH3 = 073753
 DH30 = 074132
 DH300 = 074322
 DH301 = 074322
 DH302 = 074322
 DH303 = 074412
 DH304 = 074322
 DH305 = 074322
 DH306 = 074322
 DH307 = 074322
 DH31 = 074173
 DH310 = 074322
 DH311 = 074322
 DH312 = 074322
 DH313 = 074322
 DH314 = 074322
 DH315 = 074322
 DH316 = 074322
 DH317 = 074322
 DH32 = 074043
 DH320 = 074322
 DH321 = 074322
 DH322 = 074322
 DH323 = 074322
 DH324 = 074412
 DH325 = 074322
 DH326 = 074322
 DH327 = 074322
 DH33 = 074132
 DH330 = 074322
 DH331 = 074322
 DH332 = 074322
 DH333 = 074322
 DH334 = 074322
 DH335 = 074322
 DH336 = 074322
 DH337 = 074322
 DH34 = 074173
 DH340 = 074322
 DH341 = 074322
 DH342 = 074322
 DH343 = 074322
 DH344 = 074322
 DH345 = 074322
 DH346 = 074322
 DH347 = 074322
 DH35 = 074043
 DH350 = 074322
 DH351 = 074173
 DH352 = 074322
 DH353 = 074322

DH354 = 074322
 DH355 = 074322
 DH356 = 074043
 DH357 = 074547
 DH36 = 074132
 DH360 = 074173
 DH361 = 073663
 DH362 = 074637
 DH363 = 074702
 DH364 = 074763
 DH365 = 075050
 DH366 = 075106
 DH367 = 075161
 DH37 = 074322
 DH370 = 075247
 DH371 = 000000
 DH372 = 000000
 DH373 = 000000
 DH374 = 000000
 DH375 = 000000
 DH376 = 000000
 DH377 = 000000
 DH4 = 074043
 DH40 = 074322
 DH400 = 000000
 DH401 = 074043
 DH402 = 073663
 DH403 = 074173
 DH404 = 074507
 DH405 = 074043
 DH406 = 073663
 DH407 = 074173
 DH41 = 074412
 DH410 = 074507
 DH411 = 074043
 DH412 = 073663
 DH413 = 074173
 DH414 = 074507
 DH415 = 074043
 DH416 = 073663
 DH417 = 074173
 DH42 = 074322
 DH420 = 074507
 DH421 = 074043
 DH422 = 073663
 DH423 = 074173
 DH424 = 074507
 DH425 = 074043
 DH426 = 073663
 DH427 = 074173
 DH43 = 074322
 DH430 = 074507
 DH431 = 074173
 DH432 = 074043
 DH433 = 073663
 DH434 = 074173
 DH435 = 074507

DH436 = 074173
 DH437 = 074043
 DH44 = 074322
 DH440 = 074043
 DH441 = 075312
 DH442 = 075360
 DH443 = 075360
 DH45 = 074322
 DH46 = 074322
 DH47 = 074322
 DH5 = 074132
 DH50 = 074322
 DH51 = 074322
 DH52 = 074322
 DH53 = 074412
 DH54 = 074322
 DH55 = 074322
 DH56 = 074322
 DH57 = 074322
 DH6 = 074132
 DH60 = 074322
 DH61 = 074322
 DH62 = 073663
 DH63 = 073753
 DH64 = 074132
 DH65 = 073663
 DH66 = 074043
 DH67 = 073663
 DH7 = 074043
 DH70 = 073753
 DH71 = 074132
 DH72 = 073663
 DH73 = 074132
 DH74 = 074043
 DH75 = 073663
 DH76 = 074173
 DH77 = 074132
 DIDONE = 042562
 DISPLA = 001142
 DISPRE = 000174
 DSWR = 177570
 DT1 = 076056
 DT10 = 076164
 DT100 = 076142
 DT101 = 076076
 DT102 = 076212
 DT103 = 076412
 DT104 = 076142
 DT105 = 076076
 DT106 = 076212
 DT107 = 076212
 DT11 = 076142
 DT110 = 076412
 DT111 = 076142
 DT112 = 076076
 DT113 = 076212
 DT114 = 076412

DT115 = 076142
 DT116 = 076076
 DT117 = 076212
 DT12 = 076164
 DT120 = 076412
 DT121 = 076142
 DT122 = 076076
 DT123 = 076212
 DT124 = 076412
 DT125 = 076142
 DT126 = 076076
 DT127 = 076212
 DT13 = 076212
 DT130 = 076412
 DT131 = 076076
 DT132 = 076212
 DT133 = 076412
 DT134 = 076076
 DT135 = 076164
 DT136 = 076164
 DT137 = 076224
 DT14 = 076212
 DT140 = 076164
 DT141 = 076142
 DT142 = 076142
 DT143 = 076164
 DT144 = 076142
 DT145 = 076142
 DT146 = 076164
 DT147 = 076142
 DT15 = 076164
 DT150 = 076142
 DT151 = 076164
 DT152 = 076142
 DT153 = 076142
 DT154 = 076164
 DT155 = 076142
 DT156 = 076142
 DT157 = 076164
 DT16 = 076224
 DT160 = 076142
 DT161 = 076142
 DT162 = 076164
 DT163 = 076142
 DT164 = 076142
 DT165 = 076246
 DT166 = 076246
 DT167 = 076246
 DT17 = 076212
 DT170 = 076246
 DT171 = 076246
 DT172 = 076246
 DT173 = 076312
 DT174 = 076312
 DT175 = 076312
 DT176 = 076142
 DT177 = 076142

DT2 = 076076
 DT20 = 076224
 DT200 = 076246
 DT201 = 076246
 DT202 = 076246
 DT203 = 076246
 DT204 = 076246
 DT205 = 076246
 DT206 = 076246
 DT207 = 076246
 DT21 = 076164
 DT210 = 076246
 DT211 = 076246
 DT212 = 076246
 DT213 = 076246
 DT214 = 076246
 DT215 = 076142
 DT216 = 076164
 DT217 = 076142
 DT22 = 076164
 DT220 = 076142
 DT221 = 076142
 DT222 = 076164
 DT223 = 076142
 DT224 = 076142
 DT225 = 076142
 DT226 = 076142
 DT227 = 076212
 DT23 = 076212
 DT230 = 076142
 DT231 = 076142
 DT232 = 076212
 DT233 = 076142
 DT234 = 076142
 DT235 = 076212
 DT236 = 076142
 DT237 = 076142
 DT24 = 076224
 DT240 = 076212
 DT241 = 076142
 DT242 = 076142
 DT243 = 076212
 DT244 = 076142
 DT245 = 076142
 DT246 = 076142
 DT247 = 076212
 DT25 = 076164
 DT250 = 076142
 DT251 = 076142
 DT252 = 076142
 DT253 = 076212
 DT254 = 076142
 DT255 = 076212
 DT256 = 076142
 DT257 = 076142
 DT26 = 076212
 DT260 = 076246

SYMBOL TABLE

DT261 = 076246	DT343 = 076572	DT425 = 076142	EEB25 014524	EM150 055021
DT262 = 076246	DT344 = 076572	DT426 = 076142	EECDON 024536	EM151 055045
DT263 = 076246	DT345 = 076572	DT427 = 076212	EECTP1 024426	EM152 055070
DT264 = 076246	DT346 = 076572	DT43 = 076246	EECTP2 024436	EM153 055141
DT265 = 076246	DT347 = 076572	DT430 = 076212	EEC1 024332	EM154 055166
DT266 = 076246	DT35 = 076224	DT431 = 076212	EEC10 024450	EM155 055211
DT267 = 076246	DT350 = 076572	DT432 = 076142	EEC11 024466	EM156 055262
DT27 = 076224	DT351 = 076212	DT433 = 076142	EEC2 024374	EM157 055307
DT270 = 076246	DT352 = 076572	DT434 = 076212	EEC20 024504	EM16 047414
DT271 = 076246	DT353 = 076572	DT435 = 076212	EMTVEC= 000030	EM160 055331
DT272 = 076246	DT354 = 076572	DT436 = 076212	EM1 046654	EM161 055423
DT273 = 076246	DT355 = 076572	DT437 = 076142	EM10 047230	EM162 055447
DT274 = 076246	DT356 = 076076	DT44 = 076246	EM100 053147	EM163 055472
DT275 = 076246	DT357 = 076120	DT440 = 076142	EM101 053172	EM164 055517
DT276 = 076246	DT36 = 076164	DT441 077106	EM102 053216	EM165 056315
DT277 = 076246	DT360 = 076212	DT442 077124	EM103 053257	EM166 056336
DT3 076120	DT361 = 076446	DT443 = 077124	EM104 053301	EM167 056357
DT30 = 076164	DT362 076702	DT45 = 076246	EM105 053324	EM17 047443
DT300 = 076246	DT363 076720	DT46 = 076246	EM106 053350	EM170 056400
DT301 076446	DT364 076740	DT47 = 076246	EM107 052736	EM171 056423
DT302 = 076446	DT365 = 076212	DT5 076164	EM11 047253	EM172 056446
DT303 076520	DT366 076752	DT50 = 076246	EM110 053412	EM173 056471
DT304 = 076446	DT367 077020	DT51 = 076246	EM111 053435	EM174 056514
DT305 = 076446	DT37 076246	DT52 = 076246	EM112 053461	EM175 056537
DT306 = 076446	DT370 077042	DT53 = 076312	EM113 053506	EM176 052630
DT307 = 076446	DT371 = 000000	DT54 = 076246	EM114 053550	EM177 052653
DT31 = 076212	DT372 = 000000	DT55 = 076246	EM115 053573	EM2 046713
DT310 = 076446	DT373 = 000000	DT56 = 076246	EM116 053617	EM20 047501
DT311 = 076446	DT374 = 000000	DT57 = 076246	EM117 053644	EM200 056562
DT312 = 076446	DT375 = 000000	DT6 076212	EM12 047314	EM201 056637
DT313 = 076446	DT376 = 000000	DT60 = 076246	EM120 053705	EM202 056740
DT314 = 076446	DT377 = 000000	DT61 = 076246	EM121 053727	EM203 057041
DT315 = 076446	DT4 076142	DT62 = 076224	EM122 053752	EM204 057221
DT316 = 076446	DT40 = 076246	DT63 = 076224	EM123 053776	EM205 057276
DT317 = 076446	DT400 = 000000	DT64 = 076164	EM124 054040	EM206 057375
DT32 = 076224	DT401 = 076142	DT65 = 076224	EM125 054063	EM207 057476
DT320 = 076446	DT402 = 076142	DT66 = 076142	EM126 054107	EM21 047542
DT321 = 076446	DT403 = 076212	DT67 = 076142	EM127 054134	EM210 057575
DT322 076572	DT404 = 076212	DT7 = 076142	EM13 047337	EM211 057674
DT323 = 076572	DT405 = 076142	DT70 = 076142	EM130 054176	EM212 060002
DT324 076636	DT406 = 076142	DT71 076356	EM131 054221	EM213 060103
DT325 = 076572	DT407 = 076212	DT72 = 076142	EM132 054246	EM214 060230
DT326 = 076572	DT41 076312	DT73 076412	EM133 054311	EM215 055573
DT327 = 076572	DT410 = 076212	DT74 = 076142	EM134 054335	EM216 055724
DT33 = 076164	DT411 = 076142	DT75 = 076076	EM135 054363	EM217 055746
DT330 = 076572	DT412 = 076142	DT76 = 076212	EM136 054436	EM22 = 047542
DT331 = 076572	DT413 = 076212	DT77 = 076412	EM137 054455	EM220 056016
DT332 = 076572	DT414 = 076212	EEBBF0 014404	EM14 = 047337	EM221 056042
DT333 = 076572	DT415 = 076142	EEBBF1 014414	EM140 054476	EM222 056174
DT334 = 076572	DT416 = 076142	EEBDON 014540	EM141 054517	EM223 056217
DT335 = 076572	DT417 = 076212	EEBTP1 014364	EM142 054566	EM224 056270
DT336 = 076572	DT42 = 076246	EEBTP2 014374	EM143 054611	EM225 060355
DT337 = 076572	DT420 = 076212	EEB1 014254	EM144 054633	EM226 060400
DT34 = 076212	DT421 = 076142	EEB10 014424	EM145 054703	EM227 060424
DT340 = 076572	DT422 = 076142	EEB15 014460	EM146 054727	EM23 047565
DT341 = 076572	DT423 = 076212	EEB2 014324	EM147 054751	EM230 060454
DT342 = 076572	DT424 = 076212	EEB20 014506	EM15 047373	EM231 060500

EM232	060525	EM314	064403	EM377 =	000000	EM61	052112	GGB2	015104
EM233	060556	EM315	064504	EM4	047001	EM62	052214	GGB20	015266
EM234	060602	EM316	064605	EM40	050332	EM63	052320	GGB25	015304
EM235	060627	EM317	064706	EM400 =	000000	EM64	052346	GGCDON	025156
EM236	060660	EM32	050075	EM401	070543	EM65	052422	GGCTP1	025036
EM237	060705	EM320	065007	EM402	070566	EM66	052445	GGC1	024744
EM24	047624	EM321	065110	EM403	070610	EM67	052504	GGC10	025050
EM240	060733	EM322	065211	EM404	070742	EM7	047167	GGC11	025066
EM241	060765	EM323	065246	EM405	070772	EM70	052605	GGC2	025004
EM242	061012	EM324	065305	EM406	071016	EM71	052676	GGC20	025124
EM243	061040	EM325	065344	EM407	071041	EM72	052715	GGC25	025104
EM244	061072	EM326 =	065344	EM41	050360	EM73	052776	GTSWR =	104405
EM245	061116	EM327	065505	EM410	071174	EM74	053017	HHBFF0	015464
EM246	061143	EM33	050136	EM411	071225	EM75	053041	HHBFF1	015474
EM247	061174	EM330	065607	EM412	071251	EM76	053064	HHBDON	015620
EM25	047666	EM331	065712	EM413	071274	EM77	053125	HHBTP1	015434
EM250	061225	EM332	067166	EM414	071427	ENDTES	042542	HHBTP2	015454
EM251	061252	EM333 =	065246	EM415	071460	ENDTST	040000	HHB1	015324
EM252	061300	EM334	066015	EM416	071505	ERM10	043572	HHB10	015504
EM253	061332	EM335	066111	EM417	071531	ERROR =	104000	HHB15	015540
EM254	061364	EM336	066213	EM42	050406	ERRVEC =	000004	HHB2	015374
EM255	061420	EM337	066267	EM420	071577	ERTYPE	045624	HHB20	015566
EM256	061454	EM34	050161	EM421	071631	ERT1	046004	HHB25	015604
EM257	061502	EM340	066371	EM422	071656	ERT2	046222	HHC DON	025410
EM26	047712	EM341	066473	EM423	071702	ERT3	046226	HHCTP1	025262
EM260	061531	EM342	066577	EM424	071750	ERT4	046236	HHCTP2	025272
EM261	061566	EM343	066701	EM425	072002	ERT5	046250	HHC1	025162
EM262	061625	EM344	067003	EM426	072026	FALTRP	041162	HHC10	025302
EM263	061725	EM345	067260	EM427	072051	FFBFF0	014674	HHC11	025320
EM264	061753	EM346	067360	EM43	050465	FFBFF1	014704	HHC2	025230
EM265	062050	EM347	067456	EM430	072204	FFBDON	015030	HHC20	025356
EM266	062141	EM35	050220	EM431	072235	FFBTP1	014654	HHC25	025336
EM267	062254	EM350	067502	EM432	072310	FFBTP2	014664	HT =	000011
EM27	047751	EM351	067530	EM433	072335	FFB1	014544	IDONE	040012
EM270	062351	EM352	067634	EM434	072361	FFB10	014714	IIBBF0	015764
EM271	062412	EM353	067740	EM435	072515	FFB15	014750	IIBBF1	015774
EM272	062460	EM354	070044	EM436	072547	FFB2	014614	IIBDON	016120
EM273	062551	EM355	070150	EM437	072624	FFB20	014776	IIBTP1	015734
EM274	062606	EM356	070254	EM44	050571	FFB25	015014	IIBTP2	015754
EM275	062645	EM357	070352	EM440	072652	FFCDON	024740	IIB1	015624
EM276	062745	EM36	050262	EM441	073466	FFCTP1	024632	IIB10	016004
EM277	063042	EM360	070450	EM442	073522	FFCTP2	024642	IIB15	016040
EM3	046746	EM361	072700	EM443	073554	FFC1	024542	IIB2	015674
EM30	050013	EM362	072723	EM45	050671	FFC10	024652	IIB20	016066
EM300	063116	EM363	073033	EM46	050747	FFC11	024670	IIB25	016104
EM301	063213	EM364	073101	EM47	051053	FFC2	024606	IICDON	025524
EM302	063237	EM365	073201	EM5	047041	FFC20	024706	IIC1	025414
EM303	063265	EM366	073264	EM50	051153	FPSPUR	046254	IIC2	025442
EM304	063313	EM367	073347	EM51	051267	FPVECT =	000244	IIC20	025514
EM305	063402	EM37	050306	EM52	051313	GGBBF0	015164	IIC3	025462
EM306	063505	EM370	073432	EM53	051337	GGBBF1	015174	INCDCT	040016
EM307	063672	EM371 =	000000	EM54	051363	GGBDON	015320	IOTRAP =	000020
EM31	050037	EM372 =	000000	EM55	051442	GGBTP1	015144	IOTVEC =	000020
EM310	063774	EM373 =	000000	EM56	051570	GGBTP2	015154	JJBBF0	016260
EM311	064077	EM374 =	000000	EM57	051672	GGB1	015034	JJBBF1	016270
EM312	064200	EM375 =	000000	EM6	047063	GGB10	015204	JJB DON	016414
EM313	064302	EM376 =	000000	EM60	052002	GGB15	015240	JJBTP1	016236

JJBTP2	016250	KKC4	026020	MMC4	030460	OOCTB1	032452	QOBTP2	020414
JJB1	016124	KKC5	026056	MMC5	030536	OOC1	032346	QOB1	020260
JJB10	016300	KKC6	026114	MMC6	030614	OOC10	032462	QOB10	020424
JJB15	016334	KKC7	026152	MMC7	030672	OOC15	032502	QOB15	020444
JJB2	016174	KKC8	026210	MMC8	030750	OOC2	032414	QOB2	020330
JJB20	016362	KKC9	026246	MMC9	031026	OOC20	032524	QOB20	020462
JJB25	016400	LABEL1	040530	MMR0	= 177572	OOC25	032546	QQCDON	033270
KDPAR0=	172360	LDCDSU	030036	MMR2	= 177576	OOODON	006726	QQCTB0	033136
KDPAR1=	172362	LDCFSU	027020	MMR3	= 172516	OOOT	006636	QQCTB1	033142
KDPAR2=	172364	LDCT	027224	MMVECT=	000250	OOO1	006572	QQCTB2	033152
KDPAR3=	172366	LDXSUB	031610	MNUMBE=	000443	OOO2	006616	QQC1	033032
KDPAR4=	172370	LDXT	032100	MODE1	040476	OOO3	006652	QQC10	033156
KDPAR7=	172376	LF	= 000012	MS1	046477	OOO4	006720	QQC15	033176
KDPDR0=	172320	LLBBF0	017042	MS10	= 046537	PIRQ	= 177772	QQC2	033104
KDPDR1=	172322	LLBBF1	017052	MS11	046577	PIRQVE=	000240	QQC20	033220
KDPDR2=	172324	LLBDON	017160	MS2	046515	POWERM	046426	QQC25	033242
KDPDR3=	172326	LLBTP1	017022	MS20	046622	PPBDON	020254	QQQBF0	007332
KDPDR4=	172330	LLBTP2	017032	MS21	046634	PPBTP1	020162	QQQBF1	007346
KDPDR7=	172336	LLB1	016722	MS3	046537	PPBTP2	020172	QQQDON	007550
KIPAR0=	172340	LLB10	017062	MS4	046555	PPB1	020050	QQQTP1	007362
KIPAR1=	172342	LLB15	017116	MULTER	042252	PPB10	020202	QQQ1	007172
KIPAR2=	172344	LLB2	016766	NATBF1	023562	PPB15	020222	QQQ10	007372
KIPAR3=	172346	LLB25	017144	NATER1	023512	PPB2	020116	QQQ2	007210
KIPAR4=	172350	LLCDON	030242	NATER2	023530	PPB20	020240	QQQ20	007412
KIPAR7=	172356	LLC1	027240	NATER3	023544	PPCDON	033026	QQQ22	007426
KIPDR0=	172300	LLC10	027766	NATINS	023240	PPCTB0	032700	QQQ23	007460
KIPDR1=	172302	LLC2	027306	NATRET	023524	PPCTB1	032704	QQQ24	007506
KIPDR2=	172304	LLC3	027354	NATSUB	023160	PPC1	032600	QQQ25	007532
KIPDR3=	172306	LLC4	027422	NNBFF0	017554	PPC10	032714	QQQ3	007246
KIPDR4=	172310	LLC5	027470	NNBDON	017634	PPC15	032734	QQQ4	007274
KIPDR7=	172316	LLC6	027536	NNBTP1	017534	PPC2	032646	RDCHR	= 104407
KKBBF0	016562	LLC7	027604	NNBTP2	017544	PPC20	032756	RESREG=	104411
KKBBF1	016572	LLC8	027652	NNB1	017436	PPC25	033000	RESVEC=	000010
KKBDON	016716	LLC9	027720	NNB10	017564	PPPBF0	007070	RETURN	042334
KKBTTP1	016532	LOOP	006570	NNB11	017614	PPPBF1	007104	RRBDON	020716
KKBTTP2	016552	LPERR	= 104413	NNB15	017620	PPPDON	007166	RRBTP1	020622
KKB1	016420	MMBBF0	017314	NNB2	017462	PPPTP1	007120	RRBTP2	020632
KKB10	016602	MMBBF1	017324	NNCTDON	032342	PPP1	006732	RRBTP3	020642
KKB15	016636	MMBDON	017432	NNCTB0	032214	PPP10	007130	RRB1	020502
KKB2	016470	MMBTP1	017264	NNCTB1	032220	PPP15	007150	RRB10	020644
KKB20	016664	MMBTP2	017304	NNC1	032114	PPP2	006750	RRB15	020664
KKB25	016702	MMB1	017164	NNC10	032230	PPP3	007012	RRB2	020556
KKCDON	027234	MMB10	017334	NNC15	032250	PPP4	007032	RRB20	020702
KKC1	025666	MMB15	017370	NNC2	032162	PROGNUM=	000003	RRCDON	033534
KKC10	026304	MMB2	017230	NNC20	032272	PR0	= 000000	RRCCTB0	033402
KKC11	026342	MMB25	017416	NNC25	032314	PR1	= 000040	RRCCTB1	033406
KKC12	026400	MMCDON	032110	NODAT	042364	PR2	= 000100	RRCCTB2	033416
KKC13	026436	MMC1	030246	OObDON	020044	PR3	= 000140	RRC1	033274
KKC14	026474	MMC10	031104	OObTP1	017752	PR4	= 000200	RRC10	033422
KKC15	026532	MMC11	031162	OObTP2	017762	PR5	= 000240	RRC15	033442
KKC16	026570	MMC12	031240	OOb1	017640	PR6	= 000300	RRC2	033350
KKC17	026626	MMC13	031316	OOb10	017772	PR7	= 000340	RRC20	033464
KKC18	026664	MMC14	031374	OOb15	020012	PS	= 177776	RRC25	033506
KKC19	026722	MMC15	031452	OOb2	017706	PSW	= 177776	RRRDON	010072
KKC2	025724	MMC16	031530	OOb20	020030	PWRVEC=	000024	RRREXP	007722
KKC20	026760	MMC2	030324	OOCDON	032574	QOBDON	020476	RRRTP1	007702
KKC3	025762	MMC3	030402	OOCTB0	032446	QOBTP1	020374	RRRTP2	007712

RRR1 007554	STORE 042400	TST21 014252	TTB15 021316	UUU3 010766
RRR10 007732	STXBF 036714	TST22 014542	TTB2 021212	VVBODN 014032
RRR11 007754	STXSUB 036506	TST23 015032	TTB20 021334	VVB1 013736
RRR12 007774	SWR 001140	TST24 015322	TTCDON 034272	VVB10 014000
RRR15 010026	SWREG 000176	TST25 015622	TTCTB0 034132	VVB15 014016
RRR2 007632	SW0 = 000001	TST26 016122	TTCTB1 034136	VVB2 013754
RRR25 010006	SW00 = 000001	TST27 016416	TTCTB2 034146	VVCBFO 034474
RRR3 007634	SW01 = 000002	TST3 007170	TTC1 034012	VVCODN 034502
RRR4 007660	SW02 = 000004	TST30 016720	TTC10 034152	VVCTP1 034462
RSETUP= 104412	SW03 = 000010	TST31 017162	TTC15 034172	VVC1 034402
RTI 042356	SW04 = 000020	TST32 017434	TTC2 034070	VVC2 034432
R6 = %000006	SW05 = 000040	TST33 017636	TTC20 034214	VVC3 034442
R7 = %000007	SW06 = 000100	TST34 020046	TTC25 034236	VVBF0 011246
SAVREG= 104410	SW07 = 000200	TST35 020256	TTC30 034266	VVVDON 011356
SCOPE = 000004	SW08 = 000400	TST36 020500	TTTA1 010540	VVVTP1 011256
SPACE 046472	SW09 = 001000	TST37 020720	TTTA2 010542	VVV1 011126
SR1 = 177574	SW1 = 000002	TST4 007552	TTTA3 010544	VVV10 011266
SSBDON 021140	SW10 = 002000	TST40 021142	TTTBF0 010526	VVV11 011310
SSBTP1 021044	SW11 = 004000	TST41 021352	TTTDON 010652	VVV15 011322
SSBTP2 021054	SW12 = 010000	TST42 021600	TTTTP1 010552	VVV2 011210
SSBTP3 021064	SW13 = 020000	TST43 022042	TTT1 010406	VVV20 011340
SSB1 020722	SW14 = 040000	TST44 022314	TTT10 010562	WwBDON 023574
SSB10 021066	SW15 = 100000	TST45 023576	TTT11 010604	WwB1 022316
SSB15 021106	SW2 = 000004	TST46 023756	TTT15 010616	WwB2 022374
SSB2 021000	SW3 = 000010	TST47 024134	TTT2 010472	WwB3 022452
SSB20 021124	SW4 = 000020	TST5 010074	TTT20 010634	WwB4 022530
SSCDON 034006	SW5 = 000040	TST50 024330	TTT3 010516	WwB5 022606
SSCTB0 033652	SW6 = 000100	TST51 024540	TYPE = 104401	WwB6 022664
SSCTB1 033656	SW7 = 000200	TST52 024742	TYPOC = 104402	WwB7 022742
SSC1 033540	SW8 = 000400	TST53 025160	TYPON = 104404	WwB8 023020
SSC10 033666	SW9 = 001000	TST54 025412	TYPOS = 104403	WwB9 023076
SSC15 033706	TAB = 000011	TST55 025526	UUBDON 021576	WwCDON 036144
SSC2 033610	TBITVE= 000014	TST56 025664	UUBTP1 021430	WwC1 034506
SSC20 033730	TCCBF0 025642	TST57 027236	UUBTP2 021502	WwC10 035146
SSC25 033752	TCCBF1 025652	TST6 010404	UUB1 021354	WwC11 035212
SSC30 034002	TCCDON 025662	TST60 030244	UUB10 021512	WwC12 035256
SSSA1 010244	TCC1 025530	TST61 032112	UUB15 021550	WwC13 035322
SSSBF0 010234	TCC2 025552	TST62 032344	UUB2 021426	WwC14 035366
SSSDON 010402	TCC3 025616	TST63 032576	UUB20 021532	WwC15 035432
SSSTP1 010254	TKVEC = 000060	TST64 033030	UUCBF0 034370	WwC16 035476
SSSTP2 010264	TPVEC = 000064	TST65 033272	UUCDON 034376	WwC17 035542
SSS1 010076	TRAPV 037716	TST66 033536	UUCTP1 034356	WwC18 035606
SSS10 010274	TRAPVE= 000034	TST67 034010	UUC1 034276	WwC2 034552
SSS11 010316	TRPV 041230	TST7 010654	UUC2 034326	WwC4 034616
SSS15 010326	TRTVEC= 000014	TST70 034274	UUC3 034336	WwC5 034662
SSS2 010154	TST1 006570	TST71 034400	UUUA1 011010	WwC6 034726
SSS20 010344	TST10 011124	TST72 034504	UUUA2 011012	WwC7 034772
SSS25 010364	TST100 042564	TST73 036146	UUUA3 011014	WwC8 035036
STACK = 001100	TST11 011360	TST74 036216	UUUBF0 010776	WwC9 035102
START 006106	TST12 011632	TST75 036730	UUUDON 011122	WwWBF0 011510
STCDFS 013026	TST13 012466	TST76 037134	UUUTP1 011022	WwWBF1 011530
STCDT 013310	TST14 013322	TST77 040014	UUU1 010656	WwWDON 011630
STCFDS 012172	TST15 013422	TTBDON 021350	UUU10 011032	WwWTP1 011520
STCFT 012454	TST16 013634	TTBTP1 021256	UUU11 011054	WwW1 011362
STCIBF 036134	TST17 013734	TTBTP2 021266	UUU15 011066	WwW10 011540
STCSUB 035654	TST2 006730	TTB1 021144	UUU2 010742	WwW11 011562
STKLMT= 177774	TST20 014034	TTB10 021276	UUU20 011104	WwW15 011574

WWW2	011452	ZZZDON	013420	\$ENVM	001337	\$NULL	001154	\$TESTN	001322
WWW20	011612	ZZZ1	013324	\$EOP	042564	\$NWTST=	000001	\$TIMES	001302
XXBDON	022040	ZZZ10	013366	\$EOPCT	042612	\$SOCNT	044444	\$TKB	001146
XXBTP1	021726	ZZZ15	013404	\$ERFLG	001103	\$SOMODE	044446	\$TKS	001144
XXBTP2	021736	ZZZ2	013342	\$ERMAX	001115	\$SOVER	043320	\$TMP0	001232
XXB1	021602	\$APTHD	006072	\$ERROR	043336	\$SPASS	001324	\$TMP1	001234
XXB10	021766	\$ATYC	044474	\$ERRPC	001116	\$SPASTM	006100	\$TMP10	001252
XXB15	022006	\$ATY1	044450	\$ERRTB	001442	\$SPWRAD	045600	\$TMP11	001254
XXB2	021654	\$ATY3	044456	\$ERTTL	001112	\$SPWRDN	045440	\$TMP12	001256
XXB20	022024	\$ATY4	044466	\$ESCAP	001304	\$SPWRMG	045574	\$TMP13	001260
XXB25	021746	\$AUTOB	001134	\$ETABL	001336	\$SPWRUP	045512	\$TMP14	001262
XXCDON	036214	\$BASE	001372	\$ETEND	001442	\$QUES	001312	\$TMP15	001264
XXC1	036150	\$BDADR	001122	\$FATAL	001320	\$RDCHR	045200	\$TMP16	001266
XXXDON	012464	\$BDDAT	001126	\$FFLG	044714	\$RDSZ =	000001	\$TMP17	001270
XXX1	011634	\$BELL	001306	\$FILLC	001156	\$REGAD	001160	\$TMP2	001236
XXX2	011710	\$CDW1	001376	\$FILLS	001155	\$REG0	001162	\$TMP20	001272
XXX3	011764	\$CDW2	001400	\$GDADR	001120	\$REG1	001164	\$TMP21	001274
XXX4	012040	\$CHARC	044216	\$GDDAT	001124	\$REG10	001202	\$TMP22	001276
XXX5	012114	\$CKSWR	044716	\$GET42	042750	\$REG11	001204	\$TMP23	001300
YYBDON	022312	\$CLR.T	042766	\$GTSWR	044766	\$REG12	001206	\$TMP3	001240
YYBTP1	022176	\$CMTAG	001100	\$HD =	000003	\$REG13	001210	\$TMP4	001242
YYBTP2	022206	\$CM1 =	000024	\$HIBTS	006072	\$REG14	001212	\$TMP5	001244
YYBTP3	022216	\$CM2 =	000050	\$ICNT	001104	\$REG15	001214	\$TMP6	001246
YYB1	022044	\$CM3 =	000024	\$ILLUP	045616	\$REG16	001216	\$TMP7	001250
YYB10	022240	\$CM4 =	000024	\$INTAG	001135	\$REG17	001220	\$TN =	000100
YYB15	022260	\$CNTLG	045325	\$ITEMB	001114	\$REG2	001166	\$TPB	001152
YYB2	022124	\$CNTLU	045320	\$LF	001314	\$REG20	001222	\$TPFLG	001157
YYB20	022276	\$CPUOP	001344	\$LFLG	044713	\$REG21	001224	\$TPS	001150
YYB25	022220	\$CRLF	001313	\$LOOP	043044	\$REG22	001226	\$TRAP	045354
YYCDON	036726	\$DDW0	001402	\$LPADR	001106	\$REG23	001230	\$TRAP2	045376
YYC1	036220	\$DDW1	001404	\$LPERR	001110	\$REG3	001170	\$TRP =	000014
YYC2	036256	\$DDW10	001426	\$MADR1	001350	\$REG4	001172	\$TRPAD	045410
YYC3	036314	\$DDW11	001430	\$MADR2	001354	\$REG5	001174	\$TSTM	006076
YYC4	036352	\$DDW12	001432	\$MADR3	001360	\$REG6	001176	\$TSTM	001102
YYC5	036410	\$DDW13	001434	\$MADR4	001364	\$REG7	001200	\$TYPE	043670
YYC6	036446	\$DDW14	001436	\$MAIL	001316	\$RESRE	043632	\$TYPEC	044102
YYYDON	013320	\$DDW15	001440	\$MAMS1	001346	\$RTNAD	043046	\$TYPEX	044220
YYY1	012470	\$DDW2	001406	\$MAMS2	001352	\$RTRN	043042	\$TYPOC	044246
YYY2	012544	\$DDW3	001410	\$MAMS3	001356	\$SAVRE	043574	\$TYPON	044262
YYY3	012620	\$DDW4	001412	\$MAMS4	001362	\$SAVR6	045622	\$TYPOS	044222
YYY4	012674	\$DDW5	001414	\$MBADR	006074	\$SCOPE	043056	\$UNIT	001330
YYY5	012750	\$DDW6	001416	\$MFLG	044712	\$SETUP=	000137	\$UNITM	006102
ZZCBF	037120	\$DDW7	001420	\$MNEW	045343	\$STUP =	177777	\$USWR	001342
ZZCDON	037132	\$DDW8	001422	\$MSGAD	001332	\$SVLAD	043264	\$VECT1	001366
ZZC1	036732	\$DDW9	001424	\$MSGLG	001334	\$SVPC =	006072	\$VECT2	001370
ZZC10	037072	\$DEVCT	001326	\$MSGTY	001316	\$SWR =	177400	\$XTSTR	043070
ZZC12	037106	\$DEVMT	001374	\$MSWR	045332	\$SWREG	001340	\$SET4=	000001
ZZC15	037112	\$DOAGN	043006	\$MTYP1	001347	\$SWRMK=	000000	\$OFILL	044445
ZZC2	036742	\$ENDAD	042776	\$MTYP2	001353	\$SWRMS=	000200	.LPER	046350
ZZC3	036770	\$ENDCT	042620	\$MTYP3	001357	\$TAB	046475	.RSET	046356
ZZC5	037066	\$ENULL	043052	\$MTYP4	001363	\$TBIT	043050	.\$X =	006072
ZZF1	037136	\$ENV	001336	\$MXCNT	043334	\$TERM =	000030		

. ABS. 077136 000
 000000 001
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 58584 WORDS (229 PAGES)

I 3

SEQ 0240

DYNAMIC MEMORY: 20536 WORDS (78 PAGES)
ELAPSED TIME: 00:13:55
CKFPCB.BIN,CKFPCB.SEQ=CKFPCB.MLB/ML,CKFPCB.P11