

1 .REM &

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

IDENTIFICATION

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

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

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

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

CONTENTS

57	
58	
59	
60	
61	
62	
63	
64	1. ABSTRACT
65	
66	2. REQUIREMENTS
67	2.1 EQUIPMENT
68	2.2 STORAGE
69	2.3 PRELIMINARY PROGRAMS
70	
71	3. LOADING PROCEDURE
72	
73	4. STARTING PROCEDURE
74	4.1 CONTROL SWITCH SETTINGS
75	4.2 STARTING ADDRESS
76	4.3 PROGRAM AND OPERATOR INTERACTION
77	
78	5. OPERATING PROCEDURE
79	5.1 OPERATIONAL SWITCH SETTINGS
80	5.3 OPERATOR ACTION
81	
82	6. ERRORS
83	6.1 SUMMARY
84	6.2 ERROR RECOVERY
85	
86	7. RESTRICTIONS
87	7.1 STARTING RESTRICTIONS
88	7.2 OPERATING RESTRICTIONS
89	
90	8. MISCELLANEOUS
91	8.1 EXECUTION TIMES
92	8.2 STACK POINTER
93	8.3 PASS COUNT
94	8.4 T-BIT TRAPPING
95	8.5 SOFTWARE SWITCH REGISTER
96	8.6 INTERRUPTS TEST
97	8.7 ACT, APT AND XXDP COMPATIBILITY
98	
99	9. PROGRAM DESCRIPTION
100	9.1 XXXXX
101	
102	10. LISTING
103	10.1 XXXXX
104	
105	
106	
107	
108	
109	
110	
111	
112	1. ABSTRACT

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

-----

THE THREE PROGRAMS:

DFFPA DFFPB DFFPC

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/34 FP11-A FLOATING POINT PROCESSOR, THE DESIGN IS AN ATTEMPT TO REACH ALL ROM STATES, TAKE ALL BRANCH MICRO TESTS (BUT'S) AND VERIFY ALL THE LOGIC. THEY CONSIST OF 155 (OCT) INDIVIDUAL TESTS SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY FAULTS WITH A MINIMUM HARDWARE OR SOFTWARE LEVEL. THE TESTS ARE PARTIONED INTO THREE STAND-ALONE PROGRAMS DESCRIBED BELOW.

NOTE THAT ERROR REPORTS IN THESE PROGRAMS ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS HAVE BEEN RUN AND IN MOST CASE THAT THERE IS ONLY A SINGLE POINT FAULT IN THE FP11-A. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. DFFPA

DFFPA TESTS:

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

B. DFFPB

DFFPB TESTS:

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

C. DFFPC

DFFPC TESTS:

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

169	ABSF AND ABSD
170	TSTF AND TSTD
171	NEGF, ABSF AND TSTF (ALL SOURCE MODES)
172	NEGF, ABSF AND TSTF (ALL SOURCE MODES)
173	LDFPS (ALL SOURCE MODES)
174	LDCIF AND LDCLF
175	LDCID AND LDCLD
176	LDEXP
177	STFPS (ALL DESTINATION MODES)
178	STCFL AND STCFI
179	STCDL AND STCDI
180	STEXP
181	STST

2. REQUIREMENTS  
-----

2.1 EQUIPMENT

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

2.2 STORAGE

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

2.3 PRELIMINARY PROGRAMS

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

3. LOADING PROCEDURE  
-----

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

4. STARTING PROCEDURE  
-----

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5.1

4.2 PROGRAM AND OPERATOR ACTION

224

- 225
  - 226
  - 227
  - 228
  - 229
  - 230
  - 231
  - 232
  - 233
  - 234
  - 235
  - 236
  - 237
  - 238
  - 239
  - 240
  - 241
  - 242
  - 243
  - 244
  - 245
  - 246
  - 247
  - 248
  - 249
  - 250
  - 251
  - 252
  - 253
  - 254
  - 255
  - 256
  - 257
  - 258
  - 259
  - 260
  - 261
  - 262
  - 263
  - 264
  - 265
  - 266
  - 267
  - 268
  - 269
  - 270
  - 271
  - 272
  - 273
  - 274
  - 275
  - 276
  - 277
  - 278
  - 279
  - 280
1. LOAD PROGRAM INTO MEMORY
  2. LOAD ADDRESS 200
  3. SET CONSOLE SWITCHES (IF CONSOLE IS PRESENT)
  4. PRESS START
  - ON FIRST PASS THE PROGRAM WILL IDENTIFY ITSELF. NOTE THAT IF THERE IS NO PHYSICAL CONSOLE THE PROGRAM WILL REQUEST THE OPERATOR FOR INITIAL VALUE FOR THE SOFTWARE SWITCH REGISTER (SEE SECTION 8.5). IF RUNNING UNDER ACT, APT OR CHAIN THIS DOES NOT APPLY.
  5. THE PROGRAM WILL LOOP AND AN END OF PASS AND ERROR SUMMARY WILL BE TYPED AT THE END OF EVERY PASS.

5. OPERATING PROCEDURE  
-----

5.1 OPERATIONAL SWITCH SETTINGS

THE SWITCH SETTING ARE:

	OCTAL	
SW<15>=1...	100000	HALT ON ERROR
SW<14>=1...	40000	LOOP ON CURRENT TEST
SW<13>=1...	20000	INHIBIT ERROR TYPE OUTS
SW<12>=1...	10000	INHIBIT T-BIT TRAPPING
SW<11>=1...	4000	INHIBIT ITERATIONS
SW<10>=1...	2000	RING TTY BELL ON ERROR
SW<9>=1....	1000	LOOP ON ERROR
SW<8>=1.....	400	LOOP ON TEST SPECIFIED IN SW<6> THROUGH SW<0>
SW<7>=1.....	200	PRINT ERROR SUMMARY EVEN IF SW<13>=1, THIS APPLIES ONLY TO PROGRAM DFFPA.
SW<7>=1.....	200	DESELECT CORRECT INTERRUPT TEST IN PROGRAM DFFPB, NOTE THAT THIS TEST WILL AUTOMATICALLY BE DESELECTED BY THE ABSENCE OF THE SPECIAL TEST EQUIPMENT DESIGNED TO CONDUCT THIS TEST, IF THIS EQUIPMENT IS NOT INSTALLED THERE IS NO NEED TO DESELECT THIS TEST. THIS APPLIES ONLY TO PROGRAM DFFPB!

6. ERRORS  
-----

6.1 SUMMARIES

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

281 (TYPED ON THE TTY). EVERY ERROR THOUGH IS LOGGED  
 282 AND AN ERROR SUMMARY IS PRINTED WHEN THE TEST IS  
 283 COMPLETE. NOTE THAT IS SW<13>=1 THIS SUMMARY WILL  
 284 NOT BE TYPED UNLESS SW<7>=1. IN OTHER WORDS TO GET  
 285 JUST AN ERROR SUMMARY FROM EITHER OF THESE TWO TESTS  
 286 1 AND 11 IN PROGRAM DFFPA BOTH SWITCHES 13 AND 7  
 287 MUST = 1.

6.2 ERROR RECOVERY

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

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

7. RESTRICTIONS

-----

NONE

8. MISCELLANEOUS

-----

8.1 EXECUTION TIMES

LESS THAN 10 SECONDS FOR EACH PROGRAM ON ANY PASS.

8.2 STACK POINTER

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

8.3 PASS COUNT

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

8.4 T-BIT TRAPPING

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

8.5 SOFTWARE SWITCH REGISTER

336

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

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

8.6 INTERRUPTS TEST

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

8.7 ACT, APT AND XXDP COMPATIBILITY

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

9. PROGRAM DESCRIPTION  
-----

393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448

TEST 1                   STF WITH ILLEGAL ACCUMULATOR TEST  
-----

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

TEST 2                   FDST MODE 1, FLOATING MODE, TEST  
-----

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

TEST 3                   FDST MODE 2 TEST  
-----

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

TEST 4                   FDST MODE 2, WITH GR7, TEST  
-----

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

TEST 5                   FDST MODE 4 TEST  
-----

THIS IS A TEST OF STD WITH FDST MODE 4.

TEST 6                   FDST MODE 3 TEST  
-----

THIS IS A TEST OF FDST MODE 3 USING STD.

TEST 7                   FDST MODE 5 TEST  
-----

THIS IS A TEST OF FDST MODE 5 USING STD.

TEST 10                  FDST MODE 6, INDEX MODE, TEST  
-----

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

TEST 11                  FDST MODE 7, INDEX DEFERRED MODE, TEST  
-----

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

TEST 12                  STCFD TEST  
-----



449 THIS IS A TEST OF THE STCFD INSTRUCTION,  
450  
451 TEST 13 STCDF TEST  
452 -----  
453  
454 THIS IS A TEST OF THE STCDF INSTRUCTION,  
455  
456 TEST 14 STCDF WITH ILLEGAL ACCUMULATOR TEST  
457 -----  
458  
459 THIS TEST STCFD WITH ILLEGAL AC 6.  
460  
461 TEST 15 CLRD TEST  
462 -----  
463  
464 THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS,  
465  
466 TEST 16 CLRD WITH ILLEGAL ACCUMULATOR TEST  
467 -----  
468  
469 THIS IS A TEST OF CLRD WITH ILLEGAL AC7.  
470  
471 TEST 17 NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST  
472 -----  
473  
474 THIS IS A TEST OF THE SPECIAL DEST FLOWS USING THE  
475 NEGD INST WITH MODE ZERO AND ILLEGAL AC7.  
476  
477 TEST 20 NEGF, ABSF AND TSTF SOURCE MODE 0 TEST  
478 -----  
479  
480 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS,  
481 THE NEGD INSTRUCTION IS USED TO TEST MODE 0  
482  
483 TEST 21 NEGF, ABSF AND TSTF SOURCE MODE 1 TEST  
484 -----  
485  
486 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS,  
487 THE NEGD INSTRUCTION IS USED TO TEST MODE 1  
488  
489 TEST 22 NEGF, ABSF AND TSTF SOURCE MODE 2 TEST  
490 -----  
491  
492 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS,  
493 THE ABSD INSTRUCTION IS USED TO TEST MODE 2  
494  
495 TEST 23 NEGF, ABSF AND TSTF SOURCE MODE 4 TEST  
496 -----  
497  
498 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS,  
499 THE ABSD INSTRUCTION IS USED TO TEST MODE 4  
500  
501 TEST 24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST  
502 -----  
503  
504 THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS,

505 THE ABSD INSTRUCTION IS USED TO TEST MODE 3  
506  
507 TEST 25 NEGf, ABSF AND TSTF SOURCE MODE 5 TEST  
508 -----  
509  
510 THIS IS A TEST THE NEGf, ABSF AND TSTF SOURCE FLOWS,  
511 THE NEGd INSTRUCTION IS USED TO TEST MODE 5  
512  
513 TEST 26 NEGf, ABSF AND TSTF SOURCE MODE 6 TEST  
514 -----  
515  
516 THIS IS A TEST THE NEGf, ABSF AND TSTF SOURCE FLOWS,  
517 THE ABSD INSTRUCTION IS USED TO TEST MODE 6  
518  
519 TEST 27 NEGf, ABSF AND TSTF SOURCE MODE 7 TEST  
520 -----  
521  
522 THIS IS A TEST THE NEGf, ABSF AND TSTF SOURCE FLOWS,  
523 THE ABSD INSTRUCTION IS USED TO TEST MODE 6  
524  
525 TEST 30 NEGf, ABSF AND TSTF SOURCE MODE 6, GR7, TEST  
526 -----  
527  
528 THIS IS A TEST THE NEGf, ABSF AND TSTF SOURCE FLOWS,  
529 THE NEGd INSTRUCTION IS USED TO TEST MODE 6  
530  
531 TEST 31 NEGf, ABSF AND TSTF SOURCE MODE 7, GR7, TEST  
532 -----  
533  
534 THIS IS A TEST THE NEGf, ABSF AND TSTF SOURCE FLOWS,  
535 THE ABSD INSTRUCTION IS USED TO TEST MODE 7  
536  
537 TEST 32 SPECIAL DEST, MODE 0, TEST  
538 -----  
539  
540 THIS IS A TEST OF THE NEGf ABSF AND TSTF DESTINATION  
541 FLOWS MODE 0 USING THE NEGd INSTR.  
542  
543 TEST 33 SPECIAL DEST, MODE 1, TEST  
544 -----  
545  
546 THIS IS A TEST OF THE NEGf ABSF AND TSTF DESTINATION  
547 FLOWS MODE 1 USING THE NEGd INSTR.  
548  
549 TEST 34 SPECIAL DEST, MODE 2, TEST  
550 -----  
551  
552 THIS IS A TEST OF THE NEGf ABSF AND TSTF DESTINATION  
553 FLOWS MODE 2 USING THE NEGd INSTR.  
554  
555 TEST 35 SPECIAL DEST, MODE 4, TEST  
556 -----  
557  
558 THIS IS A TEST OF THE NEGf ABSF AND TSTF DESTINATION  
559 FLOWS MODE 4 USING THE NEGd INSTR.  
560

561 TEST 36 SPECIAL DEST, MODE 3, TEST  
562 -----  
563  
564 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
565 FLOWS MODE 3 USING THE NEGD INSTR,  
566  
567 TEST 37 SPECIAL DEST, MODE 5, TEST  
568 -----  
569  
570 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
571 FLOWS MODE 5 USING THE NEGD INSTR,  
572  
573 TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST  
574 -----  
575  
576 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
577 FLOWS MODE 2 USING THE NEGF INSTR,  
578  
579 TEST 41 SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST  
580 -----  
581  
582 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
583 FLOWS MODE 2(IMMEDIATE) USING THE NEGD INSTR,  
584  
585 TEST 42 SPECIAL DEST, MODE 6, TEST  
586 -----  
587  
588 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
589 FLOWS MODE 6 USING THE NEGD INSTR,  
590  
591 TEST 43 SPECIAL DEST, MODE 7, TEST  
592 -----  
593  
594 THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION  
595 FLOWS MODE 7 USING THE NEGD INSTR,  
596  
597 TEST 44 NEGD, ABSD AND TSTD TEST  
598 -----  
599  
600 THIS IS A TEST OF THE NEGD ABSD AND TSTD  
601 INSTRUCTIONS,  
602  
603 TEST 45 SOURCE MODES, MODE 1 (FL=0), TEST  
604 -----  
605  
606 THIS IS A TEST OF SOURCE MODE 1 USING THE LDFPS  
607 INSTR  
608  
609 TEST 46 SOURCE MODES, MODE 2 (FL=0), TEST  
610 -----  
611  
612 THIS IS A TEST OF SOURCE MODE 2 USING THE LDFPS  
613 INSTR  
614  
615 TEST 47 SOURCE MODES, MODE 4 (FL=0), TEST  
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  
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

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

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

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

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

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

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

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

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

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

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

THIS IS A TEST OF THE LDCLD WITH IMMEDIATE ADDRESSING MODE

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

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

TEST 56 LDCIF AND LDCLF TEST  
-----

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

TEST 57 LDCID AND LDCLD TEST  
-----

THIS IS A TEST OF LDCID AND LDCLD

TEST 60 LDEXP TEST  
-----

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

```
673 TEST 61 DESTINATION MODES, MODE 1 (FL=0), TEST
674 -----
675
676 THIS IS A TEST OF DESTINATION MODE 1 USING THE STFPS
677 INSTRUCTION
678
679 TEST 62 DESTINATION MODES, MODE 2 (FL=0), TEST
680 -----
681
682 THIS IS A TEST OF DESTINATION MODE 2 USING THE STFPS
683 INSTRUCTION
684
685 TEST 63 DESTINATION MODES, MODE 4 (FL=0), TEST
686 -----
687
688 THIS IS A TEST OF DESTINATION MODE 4 USING THE STFPS
689 INSTRUCTION
690
691 TEST 64 DESTINATION MODES, MODE 3 (FL=0), TEST
692 -----
693
694 THIS IS A TEST OF DESTINATION MODE 3 USING THE STFPS
695 INSTRUCTION
696
697 TEST 65 DESTINATION MODES, MODE 5 (FL=0), TEST
698 -----
699
700 THIS IS A TEST OF DESTINATION MODE 5 USING THE STFPS
701 INSTRUCTION
702
703 TEST 66 DESTINATION MODES, MODE 6 (FL=0), TEST
704 -----
705
706 THIS IS A TEST OF DESTINATION MODE 6 USING THE STFPS
707 INSTRUCTION
708
709 TEST 67 DESTINATION MODES, MODE 7 (FL=0), TEST
710 -----
711
712 THIS IS A TEST OF DESTINATION MODE 7 USING THE STFPS
713 INSTRUCTION
714
715 TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST
716 -----
717
718 THIS IS A TEST OF DESTINATION MODE 2 USING STCOL
719 WITH REGISTER 0
720
721 TEST 71 DESTINATION MODES, MODE 4 (FL=1), TEST
722 -----
723
724 THIS IS A TEST OF DESTINATION MODE 4 USING STCDL
725 WITH REGISTER 0
726
727 TEST 72 STCDI AND STCDL TEST
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  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784

THIS IS A TEST OF THE STCDI AND STCDL INSTRUCTIONS.  
NOTE THAT A SUBROUTINE, 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

10. LISTING  
-----

&  
MNUMBER=443  
PROGNUM=3

000443  
000003

.LIST ME  
.NLIST MD,MC,CND

785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840

000001  
160000  
  
000244  
177400  
000200  
000011  
000015  
  
001100  
  
000011

```
.ENABL ABS

.TITLE MAINDEC-11-FPP34-A      PDP 11/34 FPP DIAGNOSTIC
;*COPYRIGHT (C) AUG 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS, 01754
;*
;*PROGRAM BY ANTHONY S. VEZZA
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
;*
STN=1
SSWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

FPVECT=244
SSWR=177400
SSWRMSK=200
TAB=11
CRLF=15

.SBTTL BASIC DEFINITIONS

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

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

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

850

;;\*GENERAL PURPOSE REGISTER DEFINITIONS

851		R0=	%0	;;GENERAL REGISTER
852	000000	R1=	%1	;;GENERAL REGISTER
853	000001	R2=	%2	;;GENERAL REGISTER
854	000002	R3=	%3	;;GENERAL REGISTER
855	000003	R4=	%4	;;GENERAL REGISTER
856	000004	R5=	%5	;;GENERAL REGISTER
857	000005	R6=	%6	;;GENERAL REGISTER
858	000006	R7=	%7	;;GENERAL REGISTER
859	000007	SP=	%6	;;STACK POINTER
860	000006	PC=	%7	;;PROGRAM COUNTER

862

;;\*PRIORITY LEVEL DEFINITIONS

863		PR0=	0	;;PRIORITY LEVEL 0
864	000000	PR1=	40	;;PRIORITY LEVEL 1
865	000040	PR2=	100	;;PRIORITY LEVEL 2
866	000100	PR3=	140	;;PRIORITY LEVEL 3
867	000140	PR4=	200	;;PRIORITY LEVEL 4
868	000200	PR5=	240	;;PRIORITY LEVEL 5
869	000240	PR6=	300	;;PRIORITY LEVEL 6
870	000300	PR7=	340	;;PRIORITY LEVEL 7

872

;;\*"SWITCH REGISTER" SWITCH DEFINITIONS

873		SW15=	100000
874	100000	SW14=	40000
875	040000	SW13=	20000
876	020000	SW12=	10000
877	010000	SW11=	4000
878	004000	SW10=	2000
879	002000	SW09=	1000
880	001000	SW08=	400
881	000400	SW07=	200
882	000200	SW06=	100
883	000100	SW05=	40
884	000040	SW04=	20
885	000020	SW03=	10
886	000010	SW02=	4
887	000004	SW01=	2
888	000002	SW00=	1

889

890	000001	.EQUIV	SW09,SW9
891		.EQUIV	SW08,SW8
892		.EQUIV	SW07,SW7
893		.EQUIV	SW06,SW6
894		.EQUIV	SW05,SW5
895		.EQUIV	SW04,SW4
896		.EQUIV	SW03,SW3



```

897      .EQUIV SW02,SW2
898      .EQUIV SW01,SW1
899      .EQUIV SW00,SW0
900
901      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
902      100000      BIT15= 100000
903      040000      BIT14= 40000
904      020000      BIT13= 20000
905      010000      BIT12= 10000
906      004000      BIT11= 4000
907      002000      BIT10= 2000
908      001000      BIT09= 1000
909      000400      BIT08= 400
910      000200      BIT07= 200
911      000100      BIT06= 100
912      000040      BIT05= 40
913      000020      BIT04= 20
914      000010      BIT03= 10
915      000004      BIT02= 4
916      000002      BIT01= 2
917      000001      BIT00= 1
918      .EQUIV BIT09,BIT9
919      .EQUIV BIT08,BIT8
920      .EQUIV BIT07,BIT7
921      .EQUIV BIT06,BIT6
922      .EQUIV BIT05,BIT5
923      .EQUIV BIT04,BIT4
924      .EQUIV BIT03,BIT3
925      .EQUIV BIT02,BIT2
926      .EQUIV BIT01,BIT1
927      .EQUIV BIT00,BIT0
928
929      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
930      000004      ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
931      000010      RESVEC= 10         ;;RESERVED AND ILLEGAL INSTRUCTIONS
932      000014      TBITVEC=14         ;; "T" BIT
933      000014      TRTVEC= 14         ;;TRACE TRAP
934      000014      BPTVEC= 14         ;;BREAKPOINT TRAP (BPT)
935      000020      IOTVEC= 20         ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
936      000024      PWRVEC= 24         ;;POWER FAIL
937      000030      EMTVEC= 30         ;;EMULATOR TRAP (EMT) **ERROR**
938      000034      TRAPVEC=34         ;; "TRAP" TRAP
939      000060      TKVEC= 60          ;;TTY KEYBOARD VECTOR
940      000064      TPVEC= 64          ;;TTY PRINTER VECTOR
941      000240      PIRQVEC=240        ;;PROGRAM INTERRUPT REQUEST VECTOR
942      .SBTTL FPP REGISTER DEFINITIONS
943      000000      AC0      =%0
944      000001      AC1      =%1
945      000002      AC2      =%2
946      000003      AC3      =%3
947      000004      AC4      =%4
948      000005      AC5      =%5
949      000006      AC6      =%6
950      000007      AC7      =%7
951
952      .SBTTL TRAP CATCHER
  
```

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

. = 0  
;\*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"  
;\*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS  
;\*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS  
. = 174  
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER  
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER  
.SBTTL STARTING ADDRESS(ES)  
JMP @#START ;JUMP TO STARTING ADDRESS OF PROGRAM

```
963          ,SBTTL  COMMON TAGS
964
965          ;*****
966          ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
967          ;*USED IN THE PROGRAM.
968
969          001100          .=1100
970          001100          $CMTAG:          ;;START OF COMMON TAGS
971          001100          000000          .WORD          0          ;;CONTAINS THE TEST NUMBER
972          001102          000          $TSTNM:          .BYTE          0          ;;CONTAINS ERROR FLAG
973          001103          000          $ERFLG:          .BYTE          0          ;;CONTAINS SUBTEST ITERATION COUNT
974          001104          000000          $ICNT:          .WORD          0          ;;CONTAINS SCOPE LOOP ADDRESS
975          001106          000000          $LPADR:          .WORD          0          ;;CONTAINS SCOPE RETURN FOR ERRORS
976          001110          000000          $LPERR:          .WORD          0          ;;CONTAINS TOTAL ERRORS DETECTED
977          001112          000000          $ERTTL:          .WORD          0          ;;CONTAINS ITEM CONTROL BYTE
978          001114          000          $ITEMB:          .BYTE          0          ;;CONTAINS MAX. ERRORS PER TEST
979          001115          001          $ERMAX:          .BYTE          1          ;;CONTAINS PC OF LAST ERROR INSTRUCTION
980          001116          000000          $ERRPC:          .WORD          0          ;;CONTAINS ADDRESS OF 'GOOD' DATA
981          001120          000000          $GDADR:          .WORD          0          ;;CONTAINS ADDRESS OF 'BAD' DATA
982          001122          000000          $BDADR:          .WORD          0          ;;CONTAINS 'GOOD' DATA
983          001124          000000          $GDDAT:          .WORD          0          ;;CONTAINS 'BAD' DATA
984          001126          000000          $BDDAT:          .WORD          0          ;;RESERVED--NOT TO BE USED
985          001130          000000          .WORD          0
986          001132          000000          .WORD          0
987          001134          000          $AUTOB:          .BYTE          0          ;;AUTOMATIC MODE INDICATOR
988          001135          000          $INTAG:          .BYTE          0          ;;INTERRUPT MODE INDICATOR
989          001136          000000          .WORD          0
990          001140          177570          SWR:          .WORD          DSWR          ;;ADDRESS OF SWITCH REGISTER
991          001142          177570          DISPLAY:          .WORD          DDISP          ;;ADDRESS OF DISPLAY REGISTER
992          001144          177560          $TKS:          177560          ;;TTY KBD STATUS
993          001146          177562          $TKB:          177562          ;;TTY KBD BUFFER
994          001150          177564          $TPS:          177564          ;;TTY PRINTER STATUS REG. ADDRESS
995          001152          177566          $TPB:          177566          ;;TTY PRINTER BUFFER REG. ADDRESS
996          001154          000          $NULL:          .BYTE          0          ;;CONTAINS NULL CHARACTER FOR FILLS
997          001155          002          $FILLS:          .BYTE          2          ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
998          001156          012          $FILLC:          .BYTE          12          ;;INSERT FILL CHARS. AFTER A "LINE FEED"
999          001157          000          $TPFLG:          .BYTE          0          ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1000         001160          000000          $REGAD:          .WORD          0          ;;CONTAINS THE ADDRESS FROM
1001         1001          ;;WHICH ($REG0) WAS OBTAINED
1002         001162          000000          $REG0:          .WORD          0          ;;CONTAINS (($REGAD)+0)
1003         001164          000000          $REG1:          .WORD          0          ;;CONTAINS (($REGAD)+2)
1004         001166          000000          $REG2:          .WORD          0          ;;CONTAINS (($REGAD)+4)
1005         001170          000000          $REG3:          .WORD          0          ;;CONTAINS (($REGAD)+6)
1006         001172          000000          $REG4:          .WORD          0          ;;CONTAINS (($REGAD)+10)
1007         001174          000000          $REG5:          .WORD          0          ;;CONTAINS (($REGAD)+12)
1008         001176          000000          $REG6:          .WORD          0          ;;CONTAINS (($REGAD)+14)
1009         001200          000000          $REG7:          .WORD          0          ;;CONTAINS (($REGAD)+16)
1010         001202          000000          $REG10:          .WORD          0          ;;CONTAINS (($REGAD)+20)
1011         001204          000000          $REG11:          .WORD          0          ;;CONTAINS (($REGAD)+22)
1012         001206          000000          $REG12:          .WORD          0          ;;CONTAINS (($REGAD)+24)
1013         001210          000000          $REG13:          .WORD          0          ;;CONTAINS (($REGAD)+26)
1014         001212          000000          $REG14:          .WORD          0          ;;CONTAINS (($REGAD)+30)
1015         001214          000000          $REG15:          .WORD          0          ;;CONTAINS (($REGAD)+32)
1016         001216          000000          $REG16:          .WORD          0          ;;CONTAINS (($REGAD)+34)
1017         001220          000000          $REG17:          .WORD          0          ;;CONTAINS (($REGAD)+36)
1018         001222          000000          $REG20:          .WORD          0          ;;CONTAINS (($REGAD)+40)
```

```

1019 001224 000000 $REG21: .WORD 0 ;;CONTAINS (($REGAD)+42)
1020 001226 000000 $REG22: .WORD 0 ;;CONTAINS (($REGAD)+44)
1021 001230 000000 $REG23: .WORD 0 ;;CONTAINS (($REGAD)+46)
1022 001232 000000 $TMP0: .WORD 0 ;;USER DEFINED
1023 001234 000000 $TMP1: .WORD 0 ;;USER DEFINED
1024 001236 000000 $TMP2: .WORD 0 ;;USER DEFINED
1025 001240 000000 $TMP3: .WORD 0 ;;USER DEFINED
1026 001242 000000 $TMP4: .WORD 0 ;;USER DEFINED
1027 001244 000000 $TMP5: .WORD 0 ;;USER DEFINED
1028 001246 000000 $TMP6: .WORD 0 ;;USER DEFINED
1029 001250 000000 $TMP7: .WORD 0 ;;USER DEFINED
1030 001252 000000 $TMP10: .WORD 0 ;;USER DEFINED
1031 001254 000000 $TMP11: .WORD 0 ;;USER DEFINED
1032 001256 000000 $TMP12: .WORD 0 ;;USER DEFINED
1033 001260 000000 $TMP13: .WORD 0 ;;USER DEFINED
1034 001262 000000 $TMP14: .WORD 0 ;;USER DEFINED
1035 001264 000000 $TMP15: .WORD 0 ;;USER DEFINED
1036 001266 000000 $TMP16: .WORD 0 ;;USER DEFINED
1037 001270 000000 $TMP17: .WORD 0 ;;USER DEFINED
1038 001272 000000 $TMP20: .WORD 0 ;;USER DEFINED
1039 001274 000000 $TMP21: .WORD 0 ;;USER DEFINED
1040 001276 000000 $TMP22: .WORD 0 ;;USER DEFINED
1041 001300 000000 $TMP23: .WORD 0 ;;USER DEFINED
1042 001302 000000 $TIMES: 0 ;;MAX, NUMBER OF ITERATIONS
1043 001304 000000 $ESCAPE:0 ;;ESCAPE ON ERROR ADDRESS
1044 001306 177607 000377 $BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL
1045 001312 077 $QUES: .ASCII /?/ ;;QUESTION MARK
1046 001313 015 $CRLF: .ASCII <15> ;;CARRIAGE RETURN
1047 001314 000012 $LF: .ASCIZ <12> ;;LINE FEED
1048 ;*****
1049 ,SBTTL APT MAILBOX-ETABLE
1050
1051 ;*****
1052 ,EVEN
1053 001316 $MAIL: ;;APT MAILBOX
1054 001316 000000 $MSGTY: .WORD AMSGTY ;;MESSAGE TYPE CODE
1055 001320 000000 $FATAL: .WORD AFATAL ;;FATAL ERROR NUMBER
1056 001322 000000 $TESTN: .WORD ATESTN ;;TEST NUMBER
1057 001324 000000 $PASS: .WORD APASS ;;PASS COUNT
1058 001326 000000 $DEVCT: .WORD ADEVCT ;;DEVICE COUNT
1059 001330 000000 $UNIT: .WORD AUNIT ;;I/O UNIT NUMBER
1060 001332 000000 $MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS
1061 001334 000000 $MSGLG: .WORD AMSGLG ;;MESSAGE LENGTH
1062 001336 $ETABLE: ;;APT ENVIRONMENT TABLE
1063 001336 000 $ENV: .BYTE AENV ;;ENVIRONMENT BYTE
1064 001337 000 $ENVM: .BYTE AENVM ;;ENVIRONMENT MODE BITS
1065 001340 000000 $SWREG: .WORD ASWREG ;;APT SWITCH REGISTER
1066 001342 000000 $USWR: .WORD AUSWR ;;USER SWITCHES
1067 001344 000000 $CPUOP: .WORD ACPUOP ;;CPU TYPE,OPTIONS
1068 ;*
1069 ;* 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1070 ;* 11/70=06,PDQ=07,Q=10
1071 ;* BIT 10=REAL TIME CLOCK
1072 ;* BIT 9=FLOATING POINT PROCESSOR
1073 ;* BIT 8=MEMORY MANAGEMENT
1074 001346 000 $MAMS1: .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE

```

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

1117 .SBTTL ERROR POINTER TABLE

1118  
 1119 ;\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
 1120 ;\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
 1121 ;\*LOCATION \$ITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT,  
 1122 ;\*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).  
 1123 ;\*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:  
 1124

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

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

1173	001620	070467				
1174					;ITEM 17	
1175	001622	043721	067705	071162	.WORD	EM17,DH17,DT17,DF17
1176	001630	070511				
1177					;ITEM 20	
1178	001632	043757	067555	071174	.WORD	EM20,DH20,DT20,DF20
1179	001640	070467				
1180					;ITEM 21	
1181	001642	044020	067644	071134	.WORD	EM21,DH21,DT21,DF21
1182	001650	070515				
1183					;ITEM 22	
1184	001652	044020	067644	071134	.WORD	EM22,DH22,DT22,DF22
1185	001660	070515				
1186					;ITEM 23	
1187	001662	044043	067705	071162	.WORD	EM23,DH23,DT23,DF23
1188	001670	070511				
1189					;ITEM 24	
1190	001672	044102	067555	071174	.WORD	EM24,DH24,DT24,DF24
1191	001700	070467				
1192					;ITEM 25	
1193	001702	044144	067644	071134	.WORD	EM25,DH25,DT25,DF25
1194	001710	070515				
1195					;ITEM 26	
1196	001712	044170	067705	071162	.WORD	EM26,DH26,DT26,DF26
1197	001720	070511				
1198					;ITEM 27	
1199	001722	044227	067555	071174	.WORD	EM27,DH27,DT27,DF27
1200	001730	070467				
1201					;ITEM 30	
1202	001732	044271	067644	071134	.WORD	EM30,DH30,DT30,DF30
1203	001740	070515				
1204					;ITEM 31	
1205	001742	044315	067705	071162	.WORD	EM31,DH31,DT31,DF31
1206	001750	070511				
1207					;ITEM 32	
1208	001752	044353	067555	071174	.WORD	EM32,DH32,DT32,DF32
1209	001760	070467				
1210					;ITEM 33	
1211	001762	044414	067644	071134	.WORD	EM33,DH33,DT33,DF33
1212	001770	070515				
1213					;ITEM 34	
1214	001772	044437	067705	071162	.WORD	EM34,DH34,DT34,DF34
1215	002000	070511				
1216					;ITEM 35	
1217	002002	044476	067555	071174	.WORD	EM35,DH35,DT35,DF35
1218	002010	070467				
1219					;ITEM 36	
1220	002012	044540	067644	071134	.WORD	EM36,DH36,DT36,DF36
1221	002020	070515				
1222					;ITEM 37	
1223	002022	044564	070034	071216	.WORD	EM37,DH37,DT37,DF37
1224	002030	070527				
1225					;ITEM 40	
1226	002032	044610	070034	071216	.WORD	EM40,DH40,DT40,DF40
1227	002040	070527				
1228					;ITEM 41	

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



1285						;ITEM 64	
1286	002272	046624	067644	071134		.WORD	EM64,DH64,DT64,DF64
1287	002300	070477					
1288						;ITEM 65	
1289	002302	046700	067375	071174		.WORD	EM65,DH65,DT65,DF65
1290	002310	070467					
1291						;ITEM 66	
1292	002312	046723	067555	071112		.WORD	EM66,DH66,DT66,DF66
1293	002320	070467					
1294						;ITEM 67	
1295	002322	046762	067375	071112		.WORD	EM67,DH67,DT67,DF67
1296	002330	070467					
1297						;ITEM 70	
1298	002332	047063	067465	071112		.WORD	EM70,DH70,DT70,DF70
1299	002340	070467					
1300						;ITEM 71	
1301	002342	047154	067644	071326		.WORD	EM71,DH71,DT71,DF71
1302	002350	070571					
1303						;ITEM 72	
1304	002352	047173	067375	071112		.WORD	EM72,DH72,DT72,DF72
1305	002360	070467					
1306						;ITEM 73	
1307	002362	047254	067644	071362		.WORD	EM73,DH73,DT73,DF73
1308	002370	070571					
1309						;ITEM 74	
1310	002372	047275	067555	071112		.WORD	EM74,DH74,DT74,DF74
1311	002400	070467					
1312						;ITEM 75	
1313	002402	047317	067375	071046		.WORD	EM75,DH75,DT75,DF75
1314	002410	070467					
1315						;ITEM 76	
1316	002412	047342	067705	071162		.WORD	EM76,DH76,DT76,DF76
1317	002420	070511					
1318						;ITEM 77	
1319	002422	047403	067644	071362		.WORD	EM77,DH77,DT77,DF77
1320	002430	070571					
1321						;ITEM 100	
1322	002432	047425	067555	071112		.WORD	EM100,DH100,DT100,DF100
1323	002440	070467					
1324						;ITEM 101	
1325	002442	047450	067375	071046		.WORD	EM101,DH101,DT101,DF101
1326	002450	070467					
1327						;ITEM 102	
1328	002452	047474	067705	071162		.WORD	EM102,DH102,DT102,DF102
1329	002460	070511					
1330						;ITEM 103	
1331	002462	047535	067644	071362		.WORD	EM103,DH103,DT103,DF103
1332	002470	070571					
1333						;ITEM 104	
1334	002472	047557	067555	071112		.WORD	EM104,DH104,DT104,DF104
1335	002500	070467					
1336						;ITEM 105	
1337	002502	047602	067375	071046		.WORD	EM105,DH105,DT105,DF105
1338	002510	070467					
1339						;ITEM 106	
1340	002512	047626	067705	071162		.WORD	EM106,DH106,DT106,DF106

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

1397	002742	050477	067375	071046	.WORD	EM131,DH131,DT131,DF131
1398	002750	070467				
1399					;ITEM 132	
1400	002752	050524	067705	071162	.WORD	EM132,DH132,DT132,DF132
1401	002760	070511				
1402					;ITEM 133	
1403	002762	050567	067644	071362	.WORD	EM133,DH133,DT133,DF133
1404	002770	070571				
1405					;ITEM 134	
1406	002772	050613	067375	071046	.WORD	EM134,DH134,DT134,DF134
1407	003000	070467				
1408					;ITEM 135	
1409	003002	050641	067644	071134	.WORD	EM135,DH135,DT135,DF135
1410	003010	070515				
1411					;ITEM 136	
1412	003012	050714	067644	071134	.WORD	EM136,DH136,DT136,DF136
1413	003020	070515				
1414					;ITEM 137	
1415	003022	050733	067375	071174	.WORD	EM137,DH137,DT137,DF137
1416	003030	070467				
1417					;ITEM 140	
1418	003032	050754	067644	071134	.WORD	EM140,DH140,DT140,DF140
1419	003040	070515				
1420					;ITEM 141	
1421	003042	050775	067555	071112	.WORD	EM141,DH141,DT141,DF141
1422	003050	070467				
1423					;ITEM 142	
1424	003052	051044	067375	071112	.WORD	EM142,DH142,DT142,DF142
1425	003060	070467				
1426					;ITEM 143	
1427	003062	051067	067644	071134	.WORD	EM143,DH143,DT143,DF143
1428	003070	070515				
1429					;ITEM 144	
1430	003072	051111	067555	071112	.WORD	EM144,DH144,DT144,DF144
1431	003100	070467				
1432					;ITEM 145	
1433	003102	051161	067375	071112	.WORD	EM145,DH145,DT145,DF145
1434	003110	070467				
1435					;ITEM 146	
1436	003112	051205	067644	071134	.WORD	EM146,DH146,DT146,DF146
1437	003120	070515				
1438					;ITEM 147	
1439	003122	051227	067555	071112	.WORD	EM147,DH147,DT147,DF147
1440	003130	070467				
1441					;ITEM 150	
1442	003132	051277	067375	071112	.WORD	EM150,DH150,DT150,DF150
1443	003140	070467				
1444					;ITEM 151	
1445	003142	051323	067644	071134	.WORD	EM151,DH151,DT151,DF151
1446	003150	070515				
1447					;ITEM 152	
1448	003152	051346	067555	071112	.WORD	EM152,DH152,DT152,DF152
1449	003160	070467				
1450					;ITEM 153	
1451	003162	051417	067375	071112	.WORD	EM153,DH153,DT153,DF153
1452	003170	070467				

1453					;ITEM 154	
1454	003172	051444	067644	071134	.WORD	EM154,DH154,DT154,DF154
1455	003200	070515				
1456					;ITEM 155	
1457	003202	051467	067555	071112	.WORD	EM155,DH155,DT155,DF155
1458	003210	070467				
1459					;ITEM 156	
1460	003212	051540	067375	071112	.WORD	EM156,DH156,DT156,DF156
1461	003220	070467				
1462					;ITEM 157	
1463	003222	051565	067644	071134	.WORD	EM157,DH157,DT157,DF157
1464	003230	070515				
1465					;ITEM 160	
1466	003232	051607	067555	071112	.WORD	EM160,DH160,DT160,DF160
1467	003240	070467				
1468					;ITEM 161	
1469	003242	051701	067375	071112	.WORD	EM161,DH161,DT161,DF161
1470	003250	070467				
1471					;ITEM 162	
1472	003252	051725	067644	071134	.WORD	EM162,DH162,DT162,DF162
1473	003260	070515				
1474					;ITEM 163	
1475	003262	051750	067375	071112	.WORD	EM163,DH163,DT163,DF163
1476	003270	070467				
1477					;ITEM 164	
1478	003272	051775	067745	071112	.WORD	EM164,DH164,DT164,DF164
1479	003300	070467				
1480					;ITEM 165	
1481	003302	052573	070034	071216	.WORD	EM165,DH165,DT165,DF165
1482	003310	070527				
1483					;ITEM 166	
1484	003312	052614	070034	071216	.WORD	EM166,DH166,DT166,DF166
1485	003320	070527				
1486					;ITEM 167	
1487	003322	052635	070034	071216	.WORD	EM167,DH167,DT167,DF167
1488	003330	070527				
1489					;ITEM 170	
1490	003332	052656	070034	071216	.WORD	EM170,DH170,DT170,DF170
1491	003340	070527				
1492					;ITEM 171	
1493	003342	052701	070034	071216	.WORD	EM171,DH171,DT171,DF171
1494	003350	070527				
1495					;ITEM 172	
1496	003352	052724	070034	071216	.WORD	EM172,DH172,DT172,DF172
1497	003360	070527				
1498					;ITEM 173	
1499	003362	052747	070124	071262	.WORD	EM173,DH173,DT173,DF173
1500	003370	070550				
1501					;ITEM 174	
1502	003372	052772	070124	071262	.WORD	EM174,DH174,DT174,DF174
1503	003400	070550				
1504					;ITEM 175	
1505	003402	053015	070124	071262	.WORD	EM175,DH175,DT175,DF175
1506	003410	070550				
1507					;ITEM 176	
1508	003412	047106	067375	071112	.WORD	EM176,DH176,DT176,DF176

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

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

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

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



1733	004542	060355	070034	071416	.WORD	EM311,DH311,DT311,DF311
1734	004550	070664				
1735					;ITEM 312	
1736	004552	060456	070034	071416	.WORD	EM312,DH312,DT312,DF312
1737	004560	070664				
1738					;ITEM 313	
1739	004562	060560	070034	071416	.WORD	EM313,DH313,DT313,DF313
1740	004570	070664				
1741					;ITEM 314	
1742	004572	060661	070034	071416	.WORD	EM314,DH314,DT314,DF314
1743	004600	070664				
1744					;ITEM 315	
1745	004602	060762	070034	071416	.WORD	EM315,DH315,DT315,DF315
1746	004610	070664				
1747					;ITEM 316	
1748	004612	061063	070034	071416	.WORD	EM316,DH316,DT316,DF316
1749	004620	070664				
1750					;ITEM 317	
1751	004622	061164	070034	071416	.WORD	EM317,DH317,DT317,DF317
1752	004630	070664				
1753					;ITEM 320	
1754	004632	061265	070034	071416	.WORD	EM320,DH320,DT320,DF320
1755	004640	070664				
1756					;ITEM 321	
1757	004642	061366	070034	071416	.WORD	EM321,DH321,DT321,DF321
1758	004650	070664				
1759					;ITEM 322	
1760	004652	061467	070034	071542	.WORD	EM322,DH322,DT322,DF322
1761	004660	070734				
1762					;ITEM 323	
1763	004662	061524	070034	071542	.WORD	EM323,DH323,DT323,DF323
1764	004670	070734				
1765					;ITEM 324	
1766	004672	061563	070124	071606	.WORD	EM324,DH324,DT324,DF324
1767	004700	070755				
1768					;ITEM 325	
1769	004702	061622	070034	071542	.WORD	EM325,DH325,DT325,DF325
1770	004710	070734				
1771					;ITEM 326	
1772	004712	061622	070034	071542	.WORD	EM326,DH326,DT326,DF326
1773	004720	070734				
1774					;ITEM 327	
1775	004722	061763	070034	071542	.WORD	EM327,DH327,DT327,DF327
1776	004730	070734				
1777					;ITEM 330	
1778	004732	062065	070034	071542	.WORD	EM330,DH330,DT330,DF330
1779	004740	070734				
1780					;ITEM 331	
1781	004742	062170	070034	071542	.WORD	EM331,DH331,DT331,DF331
1782	004750	070734				
1783					;ITEM 332	
1784	004752	063444	070034	071542	.WORD	EM332,DH332,DT332,DF332
1785	004760	070734				
1786					;ITEM 333	
1787	004762	061524	070034	071542	.WORD	EM333,DH333,DT333,DF333
1788	004770	070734				

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

1845	005220	070606				
1846					;ITEM 357	
1847	005222	064630	070261	071070	.WORD	EM357,DH357,DT357,DF357
1848	005230	070606				
1849					;ITEM 360	
1850	005232	064726	067705	071162	.WORD	EM360,DH360,DT360,DF360
1851	005240	070616				
1852					;ITEM 361	
1853	005242	067156	067375	071416	.WORD	EM361,DH361,DT361,DF361
1854	005250	070606				
1855					;ITEM 362	
1856	005252	000000	000000	000000	.WORD	EM362,DH362,DT362,DF362
1857	005260	000000				
1858					;ITEM 363	
1859	005262	000000	000000	000000	.WORD	EM363,DH363,DT363,DF363
1860	005270	000000				
1861					;ITEM 364	
1862	005272	000000	000000	000000	.WORD	EM364,DH364,DT364,DF364
1863	005300	000000				
1864					;ITEM 365	
1865	005302	000000	000000	000000	.WORD	EM365,DH365,DT365,DF365
1866	005310	000000				
1867					;ITEM 366	
1868	005312	000000	000000	000000	.WORD	EM366,DH366,DT366,DF366
1869	005320	000000				
1870					;ITEM 367	
1871	005322	000000	000000	000000	.WORD	EM367,DH367,DT367,DF367
1872	005330	000000				
1873					;ITEM 370	
1874	005332	000000	000000	000000	.WORD	EM370,DH370,DT370,DF370
1875	005340	000000				
1876					;ITEM 371	
1877	005342	000000	000000	000000	.WORD	EM371,DH371,DT371,DF371
1878	005350	000000				
1879					;ITEM 372	
1880	005352	000000	000000	000000	.WORD	EM372,DH372,DT372,DF372
1881	005360	000000				
1882					;ITEM 373	
1883	005362	000000	000000	000000	.WORD	EM373,DH373,DT373,DF373
1884	005370	000000				
1885					;ITEM 374	
1886	005372	000000	000000	000000	.WORD	EM374,DH374,DT374,DF374
1887	005400	000000				
1888					;ITEM 375	
1889	005402	000000	000000	000000	.WORD	EM375,DH375,DT375,DF375
1890	005410	000000				
1891					;ITEM 376	
1892	005412	000000	000000	000000	.WORD	EM376,DH376,DT376,DF376
1893	005420	000000				
1894					;ITEM 377	
1895	005422	000000	000000	000000	.WORD	EM377,DH377,DT377,DF377
1896	005430	000000				
1897					;ITEM 400	
1898	005432	000000	000000	000000	.WORD	EM400,DH400,DT400,DF400
1899	005440	000000				
1900					;ITEM 401	

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

1957					;ITEM 424	
1958	005672	066226	070221	071162	.WORD	EM424,DH424,DT424,DF424
1959	005700	070616				
1960					;ITEM 425	
1961	005702	066260	067555	071112	.WORD	EM425,DH425,DT425,DF425
1962	005710	070606				
1963					;ITEM 426	
1964	005712	066304	067375	071112	.WORD	EM426,DH426,DT426,DF426
1965	005720	070606				
1966					;ITEM 427	
1967	005722	066327	067705	071162	.WORD	EM427,DH427,DT427,DF427
1968	005730	070616				
1969					;ITEM 430	
1970	005732	066462	070221	071162	.WORD	EM430,DH430,DT430,DF430
1971	005740	070616				
1972					;ITEM 431	
1973	005742	066513	067705	071162	.WORD	EM431,DH431,DT431,DF431
1974	005750	070616				
1975					;ITEM 432	
1976	005752	066566	067555	071112	.WORD	EM432,DH432,DT432,DF432
1977	005760	070606				
1978					;ITEM 433	
1979	005762	066613	067375	071112	.WORD	EM433,DH433,DT433,DF433
1980	005770	070606				
1981					;ITEM 434	
1982	005772	066637	067705	071162	.WORD	EM434,DH434,DT434,DF434
1983	006000	070616				
1984					;ITEM 435	
1985	006002	066773	070221	071162	.WORD	EM435,DH435,DT435,DF435
1986	006010	070616				
1987					;ITEM 436	
1988	006012	067025	067705	071162	.WORD	EM436,DH436,DT436,DF436
1989	006020	070616				
1990					;ITEM 437	
1991	006022	067102	067555	071112	.WORD	EM437,DH437,DT437,DF437
1992	006030	070606				
1993					;ITEM 440	
1994	006032	067130	067555	071112	.WORD	EM440,DH440,DT440,DF440
1995	006040	070606				
1996					;ITEM 441	
1997	006042	067201	070351	071652	.WORD	EM441,DH441,DT441,DF441
1998	006050	071017				
1999					;ITEM 442	
2000	006052	067235	070417	071670	.WORD	EM442,DH442,DT442,DF442
2001	006060	071017				
2002					;ITEM 443	
2003	006062	067267	070417	071670	.WORD	EM443,DH443,DT443,DF443
2004	006070	071017				
2005						
2006						
2007					.SBTTL	ACT11 HOOKS
2008						
2009					;;*****	
2010					;HOOKS REQUIRED BY ACT11	
2011		006072			SSVPC=.	;SAVE PC
2012		000046			.=46	

2013 000046 037354  
 2014 000052 000052  
 2015 000052 000000  
 2016 006072  
 2017  
 2018  
 2019  
 2020  
 2021  
 2022 006072  
 2023 000024  
 2024 000024 000200  
 2025 000044  
 2026 000044 006072  
 2027 006072  
 2028  
 2029  
 2030  
 2031  
 2032 006072  
 2033 006072 000000  
 2034 006074 001316  
 2035 006076 000010  
 2036 006100 000040  
 2037 006102 000000  
 2038 006104 000052  
 2039  
 2040  
 2041 006106  
 2042  
 2043  
 2044 006106 012706 001100  
 2045 006112 005026  
 2046 006114 022706 001140  
 2047 006120 001374  
 2048 006122 012706 001100  
 2049  
 2050 006126 012737 037434 000020  
 2051 006134 012737 000340 000022  
 2052 006142 012737 037714 000030  
 2053 006150 012737 000340 000032  
 2054 006156 012737 041662 000034  
 2055 006164 012737 000340 000036  
 2056 006172 012737 041746 000024  
 2057 006200 012737 000340 000026  
 2058 006206 016767 030764 030754  
 2059 006214 005067 173062  
 2060 006220 005067 173060  
 2061 006224 112767 000001 172663  
 2062  
 2063  
 2064 006232 012737 037420 000014  
 2065 006240 012737 000340 000016  
 2066 006246 012767 000002 031144  
 2067 006254 012737 006302 000010  
 2068 006262 005046

```

SENDAD ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN ,SEOP
.=52
.WORD 0 ;;2)SET LOC.52 TO ZERO
.=SSVPC ;; RESTORE PC
.SBTTL APT PARAMETER BLOCK

;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.SX= ;;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
.=,SX ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

$APTHD:
$HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM: .WORD 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
.WORD $ETEND=$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 # $TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV #340,@#TRAPVEC+2;LEVEL 7
MOV # $PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
MOV #340,@#PWRVEC+2 ;;LEVEL 7
MOV $ENDCT,SEOPCT ;;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" (SEOP) 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
    
```

```

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

```

```

2125
2126
2127 ;*****
2128 ;*TEST 1          STF WITH ILLEGAL ACCUMULATOR TEST
2129 ;*
2130 ;*THIS IS A TEST OF THE ST INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.
2131 ;*
2132 ;*****
2133 006576 000004 TST1: SCOPE
2134
2135 006600 0001:
2136 006600 104413 LPEER                ;SET UP THE LOOP ON ERROR ADDRESS.
2137 006602 005000 CLR R0                ;SET THE FPS.
2138 006604 170100 LDFPS R0
2139
2140 006606 012737 006644 000244 MOV #000T,@#FPVECT ;SET UP FOR FP TRAPS.
2141 006614 012737 006622 001236 MOV #1$,@#$TMP2
2142
2143 006622 174007 1$: STF AC0,AC7 ;THIS TEST INSTRUCTION SHOULD
2144 ;CAUSE A TRAP.
2145
2146 ;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
2147 006624 0002:
2148 006624 170200 STFPS R0 ;GET FPS.
2149 006626 010037 001240 MOV R0,@#$TMP3
2150 006632 170300 STST R0 ;GET FEC.
2151 006634 010037 001242 MOV R0,@#$TMP4
2152 006640 104001 3$: ERROR 1 ;STF WITH ILLEGAL ACCUMULATOR, MODE
2153 ;0, DIDN'T TRAP, ST 765 TO ST 537.
2154 006642 000434 BR OODONE
2155
2156 ;TRAP TO 000T, HERE, WHEN THE EXPECTED ERROR OCCURS.
2157 006644 011600 000T: MOV (SP),R0 ;MAKE SURE THE ERROR OCCURRED
2158 006646 022700 006624 CMP #0002,R0 ;AT THE CORRECT ADDRESS.
2159 006652 001402 BEQ 0003 ;BRANCH IF TRAP ADDRESS CORRECT,
2160 006654 000137 042564 JMP @#FPSPUR ;IF INCORRECT GO REPORT SPURIOUS
2161 ;FP TRAP.
2162
2163 006660 170204 0003: STFPS R4 ;GET FPS.
2164 006662 170305 STST R5 ;GET FEC.
2165 006664 010437 001240 MOV R4,@#$TMP3 ;SAVE DATA INCASE OF ERROR.
2166 006670 010537 001242 MOV R5,@#$TMP4
2167 006674 012702 100000 MOV #100000,R2 ;EXPECTED FPS
2168 006700 012703 000002 MOV #2,R3 ;EXPECTED FEC
2169 006704 010237 001244 MOV R2,@#$TMP5
2170 006710 010337 001246 MOV R3,@#$TMP6
2171 006714 022626 CMP (SP)+,(SP)+ ;RESET THE STACK.
2172
2173 006716 020204 CMP R2,R4 ;WAS FPS CORRECT?
2174 006720 001402 BEQ 0004 ;BRANCH IF YES.
2175 ;OTHERWISE REPORT FPS INCORRECTLY
2176 006722 104002 1$: ERROR 2 ;SET AFTER USE OF ILLEGAL ACC.
2177 006724 000403 BR OODONE
2178
2179 006726 020305 0004: CMP R3,R5 ;WAS THE FEC CORRECT?
2180 006730 001401 BEQ OODONE ;BRANCH IF CORRECT,

```



```

2181
2182 006732 104003          1$:      ERROR      3          ;OTHERWISE REPORT INCORRECT FEC
2183                                     ;AFTER USE OF ILLEGAL ACC,
2184 006734
2185 006734 104412          OOODONE:
2186                                     RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
2187                                     ;SEE IF THE USER HAS EXPRESSED
2188                                     ;THE DESIRE TO CHANGE THE SOFTWARE
2189                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2190                                     ;THE USER TYPED CONTROL G?).
2191
2192
2193
2194                                     ;)*****
2195                                     ;*TEST 2          FDST MODE 1, FLOATING MODE, TEST
2196                                     ;*
2197                                     ;*THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.
2198                                     ;*
2199                                     ;)*****
2200 006736 000004          TST2:   SCOPE
2201
2202 006740          PPP1:
2203 006740 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
2204
2205 006742 012700 177777          MOV      #-1,R0          ;SET UP A BACKGROUND PATTERN IN THE
2206 006746 012701 007076          MOV      #PPPBF0,R1      ;INPUT BUFFER,
2207 006752 012702 000014          MOV      #14,R2
2208 006756 010021          PPP2:   MOV      R0,(R1)+
2209 006760 077202          SOB      R2,PPP2
2210
2211 006762 012700 000200          MOV      #200,R0          ;SET FD MODE,
2212 006766 170100          LDFPS   R0
2213 006770 012700 007126          MOV      #PPPTP1,R0      ;PUT TEST DATA INTO AC0,
2214 006774 172410          LDD     (R0),AC0
2215
2216 006776 012700 007112          MOV      #PPPBF1,R0      ;FDST ADDRESS,
2217 007002 005002          CLR     R2              ;CLEAR THE FPS,
2218 007004 170102          LDFPS   R2
2219 007006 012737 007020 001236          MOV      #PPP3,@$TMP2
2220 007014 010037 001240          MOV      R0,@$TMP3
2221
2222 007020 174010          PPP3:   STF      AC0,(R0)      ;TEST INSTRUCTION,
2223
2224 007022 022700 007112          CMP     #PPPBF1,R0      ;WAS R0 MODIFIED DURING EXECUTION?
2225 007026 001404          BEQ     PPP4            ;BRANCH IF R0 NOT MODIFIED, CORRECT,
2226
2227 007030 010037 001242          MOV     R0,@$TMP4      ;OTHERWISE REPORT ERROR, R0 MODIFIED.
2228 007034 104004          1$:    ERROR      4
2229 007036 000456          BR      PPPDONE        ;GO TO NEXT TEST,
2230
2231 007040 012700 007112          PPP4:   MOV     #PPPBF1,R0      ;CHECK THE DATA IN THE OUTPUT BUFFER,
2232 007044 012701 007126          MOV     #PPPTP1,R1
2233 007050 022021          CMP     (R0)+,(R1)+
2234 007052 001031          BNE    PPP10           ;BRANCH IF INCORRECT,
2235 007054 022011          CMP     (R0)+,(R1)
2236 007056 001027          BNE    PPP10           ;BRANCH IF INCORRECT,

```

```

2237 007060 022720 177777 CMP #-1,(R0)+ ;WAS FLOATING MODE USED?
2238 007064 001034 BNE PPP15 ;BRANCH IF NOT,
2239 007066 022710 177777 CMP #-1,(R0)
2240 007072 001031 BNE PPP15
2241 007074 000437 BR PPPDONE ;GO TO NEXT TEST,
2242
2243 007076 177777 177777 177777 PPPBF0: .WORD -1,-1,-1,-1,-1,-1
2244 007104 177777 177777 177777
2245
2246 007112 177777 177777 177777 PPPBF1: .WORD -1,-1,-1,-1,-1,-1
2247 007120 177777 177777 177777
2248
2249 007126 123456 023456 PPPTP1: .WORD 123456,23456
2250 007132 034567 045671 .WORD 34567,45671
2251
2252 ;REPORT DATA IN OUT PUT BUFFER INCORRECT,
2253 007136 012737 007126 001242 PPP10: MOV #PPPTP1,@$TMP4
2254 007144 012737 007112 001240 MOV #PPPBF1,@$TMP3
2255 007152 104005 1$: ERROR 5 ;BAD DATA,
2256 007154 000407 BR PPPDONE
2257
2258 ;REPORT FLOATING MODE NOT USED, BUT FD FAILED,
2259 007156 012737 007126 001242 PPP15: MOV #PPPTP1,@$TMP4
2260 007164 012737 007112 001240 MOV #PPPBF1,@$TMP3
2261 007172 104006 1$: ERROR 6 ;ST 707 TO 245 INTO 244 (BUT FD).
2262
2263 007174 PPPDONE:
2264 007174 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
2265 ;SEE IF THE USER HAS EXPRESSED
2266 ;THE DESIRE TO CHANGE THE SOFTWARE
2267 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2268 ;THE USER TYPED CONTROL G?).
2269
2270
2271
2272
2273 ;*****
2274 ;*TEST 3 FDST MODE 2 TEST
2275 ;*
2276 ;*THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2,
2277 ;*
2278 ;*****
2279 007176 000004 TST3: SCOPE
2280
2281 ;FIRST TEST STF,
2282 007200 QQQ1:
2283 007200 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
2284
2285 007202 012700 177777 MOV #-1,R0 ;SET UP THE OUTPUT BUFFER,
2286 007206 012701 007340 MOV #QQQBFO,R1
2287 007212 012702 000014 MOV #14,R2
2288 007216 010021 QQQ2: MOV R0,(R1)+
2289 007220 077202 SOB R2,QQQ2
2290
2291 007222 012700 000200 MOV #200,R0 ;SET FD MODE,
2292 007226 170100 LDFPS R0
  
```

```

2293 007230 012700 007370      MOV      #QQQTP1,R0      ;SETUP AC0.
2294 007234 172410      LDD      (R0),AC0
2295
2296 007236 012700 007354      MOV      #QQQBF1,R0      ;FDST ADDRESS.
2297 007242 005002      CLR      R2
2298 007244 170102      LDFPS   R2              ;SET FPS.
2299 007246 012737 007254 001236  MOV      #QQQ3,@#TMP2
2300
2301 007254 174020      QQQ3:   STF      AC0,(R0)+      ;TEST INSTRUCTION.
2302
2303 007256 022700 007360      CMP      #QQQBF1+4,R0      ;WAS R0 INCREMENTED BY 4 PROPERLY?
2304
2305 007262 001407      BEQ      QQQ4              ;BRANCH IF R0 CORRECT.
2306 007264 010037 001242      MOV      R0,@#TMP4        ;REPORT R0 INCORRECT AFTER FDST MODE 2.
2307 007270 012737 007360 001240  MOV      #QQQBF1+4,@#TMP3
2308 007276 104007      1$:     ERROR      7              ;BAD CONSTANT USED OR DIDN'T GO 527 TO 642
2309 007300 000526      BR       QQQDONE
2310 007302 012700 007354      QQQ4:   MOV      #QQQBF1,R0      ;WAS THE OUTPUT DATA CORRECT?
2311 007306 012701 007370      MOV      #QQQTP1,R1
2312 007312 022021      CMP      (R0)+,(R1)+
2313 007314 001031      BNE     QQQ10              ;BRANCH IF INCORRECT.
2314 007316 022021      CMP      (R0)+,(R1)+
2315 007320 001027      BNE     QQQ10              ;BRANCH IF INCORRECT.
2316 007322 022027 177777      CMP      (R0)+,#-1        ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
2317 007326 001024      BNE     QQQ10              ;BRANCH IF INCORRECT.
2318 007330 022027 177777      CMP      (R0)+,#-1
2319 007334 001021      BNE     QQQ10              ;BRANCH IF INCORRECT.
2320 007336 000430      BR       QQQ20
2321 007340 177777 177777 177777  QQQBF0: .WORD      -1,-1,-1,-1,-1,-1
2322 007346 177777 177777 177777
2323 007354 177777 177777 177777  QQQBF1: .WORD      -1,-1,-1,-1,-1,-1
2324 007362 177777 177777 177777
2325 007370 076543      QQQTP1: 76543
2326 007372 065432      65432
2327 007374 054321      54321
2328 007376 043210      43210
2329      ;REPORT OUTPUT DATA INCORRECT:
2330 007400 012737 007370 001240  QQQ10:  MOV      #QQQTP1,@#TMP3
2331 007406 012737 007354 001242  MOV      #QQQBF1,@#TMP4
2332 007414 104010      1$:     ERROR      10              ;BAD DATA
2333 007416 000457      BR       QQQDONE
2334
2335      ;NOW TEST STD MODE 2.
2336
2337 007420      QQQ20:
2338 007420 104413      LPERR
2339 007422 012700 007340      MOV      #QQQBF0,R0      ;SET UP THE LOOP ON ERROR ADDRESS,
2340 007426 010001      MOV      R0,R1          ;SET UP DEFAULT INPUT DATA BUFFER.
2341 007430 012702 000014      MOV      #14,R2
2342 007434 010021      QQQ22:  MOV      R0,(R1)+
2343 007436 077202      SOB     R2,QQQ22
2344 007440 012700 000200      MOV      #200,R0        ;ENTER FLOATING DOUBLE MODE,
2345 007444 170100      LDFPS   R0
2346 007446 012700 007370      MOV      #QQQTP1,R0      ;LOAD AC0.
2347 007452 172410      LDD      (R0),AC0
2348 007454 012700 007354      MOV      #QQQBF1,R0      ;SET DESTINATION ADDRESS,

```

```

2349 007460 012737 007466 001236      MOV      #QQQ23,@#TMP2
2350 007466 174020      QQQ23:  STD      AC0,(R0)+      ;TEST INSTRUCTION,
2351 007470 022700 007364      CMP      #QQQBF1+10,R0   ;WAS R0 INCREMENTED BY 10 CORRECTLY?
2352 007474 001407      BEQ      QQQ24           ;BRANCH IF CORRECT.
2353 007476 010037 001242      MOV      R0,@#TMP4       ;REPORT R0 INCORRECTLY INCREMENTED.
2354 007502 012737 007364 001240      MOV      #QQQBF1+10,@#TMP3
2355 007510 104011      1$:      ERROR      11      ;DO NOT INCREM BY 10 BAD CONSTANT
2356 007512 000421      BR       QQQDONE
2357 007514 012700 007354      QQQ24:  MOV      #QQQBF1,R0     ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
2358 007520 012701 007370      MOV      #QQQTP1,R1
2359 007524 012702 000004      MOV      #4,R2
2360 007530 022021      1$:      CMP      (R0)+,(R1)+
2361 007532 001002      BNE      QQQ25           ;BRANCH IF INCORRECT.
2362 007534 077203      SOB      R2,1$
2363 007536 000407      BR       QQQDONE
2364      ;REPORT DATA INCORRECT.
2365 007540 012737 007370 001240      QQQ25:  MOV      #QQQTP1,@#TMP3
2366 007546 012737 007354 001242      MOV      #QQQBF1,@#TMP4
2367 007554 104012      1$:      ERROR      12      ;BAD DATA
2368 007556      QQQDONE:
2369 007556 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
2370      ;SEE IF THE USER HAS EXPRESSED
2371      ;THE DESIRE TO CHANGE THE SOFTWARE
2372      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2373      ;THE USER TYPED CONTROL G?).
2374
2375      ;*****
2376      ;*TEST 4      FDST MODE 2, WITH GR7, TEST
2377      ;*
2378      ;*THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.
2379      ;*
2380      ;*****
2381 007560 000004      TST4:    SCOPE
2382
2383 007562      RRR1:
2384 007562 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS,
2385 007564 012700 007642      MOV      #RRR3,R0       ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
2386 007570 012701 007710      MOV      #RRRTP1,R1
2387 007574 012702 000004      MOV      #4,R2
2388 007600 012021      1$:      MOV      (R0)+,(R1)+
2389 007602 077202      SOB      R2,1$
2390 007604 012700 000200      MOV      #200,R0        ;ENTER FLOATING DOUBLE MODE.
2391 007610 170100      LDFPS     R0
2392 007612 012700 007720      MOV      #RRRTP2,R0     ;SET UP AC0,
2393 007616 172410      LDD      (R0),AC0
2394 007620 012737 007740 000004      MOV      #RRR10,@#ERRVECT ;SET UP FOR AN ODD ADDRESS.
2395 007626 012737 007640 001236      MOV      #RRR2,@#TMP2
2396 007634 005001      CLR      R1
2397 007636 005004      CLR      R4
2398      ;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
2399      ;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
2400      ;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
2401      ;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
2402      ;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
2403      ;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
2404 007640 174027      RRR2:    STD      AC0,(R7)+      ;TEST INSTRUCTION,

```

```

2405 007642 005201          RRR3:  INC    R1          ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4,
2406 007644 005201          INC    R1
2407 007646 005201          INC    R1
2408 007650 005201          INC    R1
2409 007652 012700 007730   MOV    #RRREXP,R0      ;SEE IF THE DATA WAS OUTPUT CORRECTLY,
2410 007656 012702 007642   MOV    #RRR3,R2
2411 007662 012703 000004   MOV    #4,R3
2412 007666 022022          RRR4:  CMP    (R0)+,(R2)+
2413 007670 001051          BNE    RRR25          ;BRANCH IF INCORRECT,
2414 007672 077303          SOB    R3,RRR4
2415 007674 005704          TST    R4            ;MAKE SURE R4 IS 0,
2416 007676 001056          BNE    RRR15          ;BRANCH IF R4 IS INCORRECT,
2417 007700 022701 000003   CMP    #3,R1         ;SEE IF R1 IS CORRECT,
2418 007704 001053          BNE    RRR15          ;BRANCH IF R1 IS INCORRECT,
2419 007706 000474          BR     RRRDONE
2420                          ;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3,
2421 007710 005201          RRRTP1: INC   R1
2422 007712 005201          INC   R1
2423 007714 005201          INC   R1
2424 007716 005201          INC   R1
2425                          ;THIS IS THE DATA PUT IN AC0 BEFORE EXECUTION OF THE STD,
2426 007720 005204          RRRTP2: INC   R4
2427 007722 005204          INC   R4
2428 007724 005204          INC   R4
2429 007726 005204          INC   R4
2430                          ;THIS IS THE EXPECTED DATA AT RRR3 AFTER EXECUTION OF THE STD,
2431 007730 005204          RRREXP: INC  R4
2432 007732 005201          INC  R1
2433 007734 005201          INC  R1
2434 007736 005201          INC  R1
2435                          ;IF A FAILURE IN THE FDST FLOWS RESULTS IN AN ODD ADDRESS TRAP THROUGH
2436                          ;4 TO HERE:
2437 007740 011602          RRR10: MOV   (SP),R2      ;SEE IF THE TRAP WAS BECAUSE OF AN ODD ADDRESS,
2438 007742 032702 000001   BIT   #1,R2
2439 007746 001005          BNE   RRR11          ;BRANCH IF YES,
2440 007750 020227 007644   CMP   R2,#RRR3+2     ;SEE IF THE TRAP OCCURRED AT THE TEST INSTRUCTION,
2441 007754 001412          BEQ   RRR12          ;BRANCH IF YES,
2442 007756 000137 042620   JMP   @#CSPUR        ;OTHERWISE REPORT A SPURIOUS TRAP THROUGH VECTOR 4,
2443                          ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP,
2444 007762 010237 001236   RRR11: MOV   R2,@#STMP2
2445 007766 012737 007644 001240  MOV   #RRR3+2,@#STMP3
2446 007774 022626          CMP   (SP)+,(SP)+
2447 007776 104013          1$:   ERROR  13          ;BAD CONSTANT #2 + PC ODD ADDR,
2448 010000 000437          BR    RRRDONE
2449 010002 010237 001236   RRR12: MOV   R2,@#STMP2
2450 010006 022626          CMP   (SP)+,(SP)+
2451 010010 104014          1$:   ERROR  14          ;ODD ADDRESS TRAP
2452 010012 000432          BR    RRRDONE        ;WRONG MODE USED,
2453
2454                          ;REPORT DATA INCORRECT:
2455 010014 012737 007642 001240  RRR25: MOV   #RRR3,@#STMP3
2456 010022 012737 007730 001242  MOV   #RRREXP,@#STMP4
2457 010030 104015          1$:   ERROR  15          ;BAD DATA BUT GR7 FAIL
2458 010032 000422          BR    RRRDONE
2459
2460                          ;REPORT PC INCORRECT MODIFIED DURING THE EXECUTION OF FDST IMMEDIATE

```

```

2461 ;MODE, THE PC SHOULD HAVE BEEN INCREMENTED BY 2 BUT IT WASN'T.
2462 ;USE R1 AND R4 TO COMPUTE THE ACTUAL ACTION THAT WAS TAKEN ON THE PC.
2463 010034 012737 007644 001240 RRR15: MOV #RRR3+2,@$TMP3
2464 010042 005704 TST R4 ;IS R4 CLEAR.
2465 010044 001404 BEQ 1$
2466 010046 012737 007642 001242 MOV #RRR3,@$TMP4
2467 010054 000410 BR 2$
2468 010056 012702 007644 1$: MOV #RRR3+2,R2
2469 010062 062701 177775 ADD #-3,R1
2470 010066 006301 ASL R1
2471 010070 160102 SUB R1,R2
2472 010072 010237 001242 MOV R2,@$TMP4
2473 010076 2$:
2474 010076 104016 3$: ERROR 16 ;BAD CONSTANT PC+
2475 010100 RRRDONE:
2476 010100 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
2477 ;SEE IF THE USER HAS EXPRESSED
2478 ;THE DESIRE TO CHANGE THE SOFTWARE
2479 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2480 ;THE USER TYPED CONTROL G?).
2481
2482 ;*****
2483 ;*TEST 5 FDST MODE 4 TEST
2484 ;*
2485 ;*THIS IS A TEST OF STD WITH FDST MODE 4.
2486 ;*
2487 ;*****
2488 010102 000004 TST5: SCOPE
2489
2490 010104 SSS1:
2491 010104 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
2492 010106 012700 177777 MOV #-1,R0 ;SET UP THE OUTPUT BUFFER,
2493 010112 012701 010242 MOV #SSSBF0,R1
2494 010116 012702 000010 MOV #10,R2
2495 010122 010021 1$: MOV R0,(R1)+
2496 010124 077202 SOB R2,1$
2497 010126 012700 000200 MOV #200,R0 ;ENTER FLOATING DOUBLE MODE,
2498 010132 170100 LDFPS R0
2499 010134 012700 010262 MOV #SSSTP1,R0 ;SET UP AC0.
2500 010140 172410 LDD (R0),AC0
2501 010142 012737 010302 000004 MOV #SSS10,@$ERRVECT ;SET UP FOR A TRAP TO 4.
2502 010150 012737 010162 001236 MOV #SSS2,@$TMP2
2503 010156 012700 010252 MOV #SSSA1,R0 ;SET UP THE DESTINATION ADDRESS.
2504
2505 010162 174040 SSS2: STD AC0,-(R0) ;TEST INSTRUCTION,
2506 010164 005201 INC R1
2507 010166 020027 010242 CMP R0,#SSSBF0 ;SEE IF R0 WAS DECREMENTED PROPERLY.
2508 010172 001060 BNE SSS15 ;BRANCH IF R0 IS INCORRECT.
2509 010174 012700 010242 MOV #SSSBF0,R0 ;WAS THE OUTPUT DATA CORRECT?
2510 010200 012701 010262 MOV #SSSTP1,R1
2511 010204 012702 000004 MOV #4,R2
2512 010210 022021 1$: CMP (R0)+,(R1)+
2513 010212 001057 BNE SSS20 ;BRANCH IF INCORRECT.
2514 010214 077203 SOB R2,1$
2515 010216 012700 177777 MOV #-1,R0 ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
2516 010222 012701 010252 MOV #SSSA1,R1

```

2517 010226 012702 000004

2\$: MOV #4,R2  
 CMP R0,(R1)+  
 BNE SSS25 ;BRANCH IF INCORRECT.  
 SOB R2,2\$  
 BR SSSDONE

2522

;THIS IS THE OUTPUT DATA BUFFER.

2524 010242 177777

SSSBF0: -1

2525 010244 177777

-1

2526 010246 177777

-1

2527 010250 177777

-1

2528 010252 177777

SSSA1: -1

2529 010254 177777

-1

2530 010256 177777

-1

2531 010260 177777

-1

2532

;THIS IS THE TEST DATA LOADED INTO AC0:

2534 010262 147250

SSSTP1: 147250

2535 010264 036147

36147

2536 010266 025036

25036

2537 010270 147250

147250

2538 010272 177777

SSSTP2: -1

2539 010274 177777

-1

2540 010276 177777

-1

2541 010300 177777

-1

2542

;IF AN ODD ADDRESS TRAP OCCURS COME HERE;

2544 010302 011600

SSS10: MOV (SP),R0 ;SEE IF THE TRAP ACCURRED ON THE TEST INSTRUCTION,  
 CMP R0,#SSS2+2

2545 010304 020027 010164

BEQ SSS11 ;BRANCH IF YES.

2546 010310 001405

CMP R0,#SSS2+4

2547 010312 020027 010166

BEQ SSS11 ;BRANCH IF YES.

2548 010316 001402

JMP @#CPSPUR ;OTHERWISE GO REPORT A SPURIOUS TRAP THROUGH 4,

2549 010320 000137 042620

;REPORT FAILURE IN FDST FLOWS RESULTED IN AN ODD ADDRESS.

2550

2551 010324 010037 001236

SSS11: MOV R0,@#TMP2

2552 010330 104017

2\$: ERROR 17 ;FDST FORK X ODD AD RES,  
 BR SSSDONE

2553 010332 000426

2554

;REPORT R0 INCORRECTLY DECREMENTED.

2556 010334 010037 001242

SSS15: MOV R0,@#TMP4

2557 010340 012737 010242 001240

1\$: MOV #SSSBF0,@#TMP3 ;R0 NOT DECRE PROP  
 BR SSSDONE

2558 010346 104020

2559 010350 000417

2560

;REPORT OUTPUT DATA INCORRECT:

2562 010352 012737 010242 001240

SSS20: MOV #SSSBF0,@#TMP3

2563 010360 012737 010262 001242

1\$: MOV #SSSTP1,@#TMP4 ;BAD DATA  
 BR SSSDONE

2564 010366 104021

2565 010370 000407

2566 010372 012737 010252 001242

SSS25: MOV #SSSA1,@#TMP4

2567 010400 012737 010272 001240

1\$: MOV #SSSTP2,@#TMP3 ;DATA BAD OUTSIDE TARGET AREA  
 BR SSSDONE

2568 010406 104022

2569 010410

2570 010410 104412

RSETUP ;GO INITIALIZE THE FPS AND STACK; AND

2571

;SEE IF THE USER HAS EXPRESSED

2572

;THE DESIRE TO CHANGE THE SOFTWARE

2573 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
 2574 ;THE USER TYPED CONTROL G?).  
 2575

2576 ;\*\*\*\*\*  
 2577 ;\*TEST 6 FDST MODE 3 TEST  
 2578 ;\*  
 2579 ;\*THIS IS A TEST OF FDST MODE 3 USING STD.  
 2580 ;\*

2581 ;\*\*\*\*\*  
 2582 010412 000004 TST6: SCOPE

2583  
 2584 010414 TTT1:  
 2585 010414 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
 2586 010416 012701 010534 MOV #TTBF0,R1 ;SET UP THE OUTPUT DATA BUFFER.  
 2587 010422 012700 177777 MOV #-1,R0  
 2588 010426 012702 000013 MOV #13,R2  
 2589 010432 010021 1\$: MOV R0,(R1)+  
 2590 010434 077202 SOB R2,1\$  
 2591 010436 012737 010534 010550 MOV #TTBF0,@TTTA2  
 2592 010444 012700 000200 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE.  
 2593 010450 170100 LDFPS R0  
 2594 010452 012700 010560 MOV #TTTTP1,R0 ;SET UP AC0.  
 2595 010456 172410 LDD (R0),AC0  
 2596 010460 012737 010570 000004 MOV #TTT10,@ERRVECT ;SET UP FOR TRAPS TO 4.  
 2597 010466 016737 000006 001236 MOV TTT2,@\$TMP2  
 2598 010474 012700 010550 MOV #TTTA2,R0 ;SET UP THE DESTINATION ADDRESS.

2599  
 2600 010500 174030 TTT2: STD AC0,@(R0)+ ;TEST INSTRUCTION.  
 2601  
 2602 010502 020027 010552 CMP R0,#TTTA2+2 ;SEE IF R0 WAS INCREMENTED CORRECTLY.  
 2603 010506 001046 BNE TTT15 ;BRANCH IF INCORRECT.  
 2604 010510 012701 010534 MOV #TTBF0,R1 ;CHECK THE OUTPUT DATA BUFFER.  
 2605 010514 012702 010560 MOV #TTTTP1,R2  
 2606 010520 012703 000004 MOV #4,R3  
 2607 010524 022122 TTT3: CMP (R1)+,(R2)+  
 2608 010526 001045 BNE TTT20 ;BRANCH IF NOT CORRECT.  
 2609 010530 077303 SOB R3,TTT3  
 2610 010532 000452 BR TTTDONE

2611 ;THIS IS THHE OUTPUT DATA BUFFER:

2612  
 2613 010534 177777 TTTBF0: -1  
 2614 010536 177777 -1  
 2615 010540 177777 -1  
 2616 010542 177777 -1  
 2617 010544 177777 -1  
 2618 010546 177777 TTTA1: -1  
 2619 010550 010534 TTTA2: TTTBF0  
 2620 010552 177777 TTTA3: -1  
 2621 010554 177777 -1  
 2622 010556 177777 -1  
 2623 010560 101213 TTTTTP1: 101213  
 2624 010562 141516 141516  
 2625 010564 071727 71727  
 2626 010566 037475 37475

2627  
 2628 ;TRAP THROUGH VECTOR 4 TO HERE,



```

2629 010570 011602          TTT10: MOV      (SP),R2          ;SEE IF THE TRAP ADDRESS IS THAT OF THE TEST INSTRUCTION
2630 010572 020227 010502    CMP      R2,#TTT2+2
2631 010576 001405          BEQ      TTT11              ;BRANCH IF YES.
2632 010600 020227 010504    CMP      R2,#TTT2+4
2633 010604 001402          BEQ      TTT11              ;BRANCH IF YES.
2634 010606 000137 042620    JMP      @#CPSPUR          ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
2635
2636                          ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
2637 010612 010237 001236    TTT11:  MOV      R2,@#STMP2
2638 010616 022626          CMP      (SP)+,(SP)+
2639 010620 104023          1$:      ERROR   23          ;BET FDST X ODD ADR
2640 010622 000416          BR       TTTDONE
2641
2642                          ;REPORT R0 INCORRECT:
2643 010624 010037 001242    TTT15:  MOV      R0,@#STMP4
2644 010630 012737 010552 001240  MOV      #TTTA2+2,@#STMP3
2645 010636 104024          1$:      ERROR   24          ;R0 NOT INCREMENT PROPERLY
2646 010640 000407          BR       TTTDONE
2647
2648                          ;REPORT INCORRECT OUTPUT DATA:
2649 010642 012737 010534 001240  TTT20:  MOV      #TTTBF0,@#STMP3
2650 010650 012737 010560 001242    MOV      #TTTTP1,@#STMP4
2651 010656 104025          1$:      ERROR   25          ;BAD DATA
2652 010660
2653 010660 104412          TTTDONE: RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
2654                          ;SEE IF THE USER HAS EXPRESSED
2655                          ;THE DESIRE TO CHANGE THE SOFTWARE
2656                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2657                          ;THE USER TYPED CONTROL G?).
2658
2659                          ;*****
2660                          ;*TEST 7          FDST MODE 5 TEST
2661                          ;*
2662                          ;*THIS IS A TEST OF FDST MODE 5 USING STD.
2663                          ;*
2664                          ;*****
2665 010662 000004          TST7:   SCOPE
2666
2667 010664          UUU1:
2668 010664 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
2669 010666 012701 011004    MOV      #UUUBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
2670 010672 012700 177777    MOV      #-1,R0
2671 010676 012702 000013    MOV      #13,R2
2672 010702 010021          1$:      MOV      R0,(R1)+
2673 010704 077202          SOB      R2,1$
2674 010706 012737 011004 011016  MOV      #UUUBF0,@#UUUA1
2675 010714 012700 000200    MOV      #200,R0        ;ENTER DOUBLE FLOATING MODE.
2676 010720 170100          LDFPS   R0
2677 010722 012700 011030    MOV      #UUUTP1,R0      ;SET UP AC0.
2678 010726 172410          LDD      (R0),AC0
2679 010730 012737 011040 000004  MOV      #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
2680 010736 016737 000006 001236  MOV      UUU2,@#STMP2
2681 010744 012700 011020    MOV      #UUUA2,R0      ;SET UP THE DESTINATION ADDRESS.
2682 010750 174050          UUU2:  STD      AC0,@-(R0)  ;TEST INSTRUCTION.
2683 010752 020027 011016    CMP      R0,#UUUA2-2    ;WAS R0 DECRIMENTED PROPERLY?
2684 010756 001046          BNE     UUU15          ;BRANCH IF R0 IS INCORRECT.

```

```

2685 010760 012701 011004      MOV      #UUUBF0,R1      ;WAS THE DATA OUTPUT CORRECTLY?
2686 010764 012702 011030      MOV      #UUUTP1,R2
2687 010770 012703 000004      MOV      #4,R3
2688 010774 022122      UUU3:   CMP      (R1)+,(R2)+
2689 010776 001045      BNE     UUU20          ;BRANCH IF DATA IS INCORRECT.
2690 011000 077303      SOB     R3,UUU3
2691 011002 000452      BR      UUUDONE
2692

```

```

2693      ;THIS IS THE OUTPUT DATA BUFFER
2694 011004 177777      UUUBF0: -1
2695 011006 177777      -1
2696 011010 177777      -1
2697 011012 177777      -1
2698 011014 177777      -1
2699 011016 011004      UUUA1:  UUUBF0
2700 011020 177777      UUUA2:  -1
2701 011022 177777      UUUA3:  -1
2702 011024 177777      -1
2703 011026 177777      -1
2704 011030 020212      UUUTP1: 20212
2705 011032 023242      23242
2706 011034 026273      26273
2707 011036 031323      031323
2708

```

```

2709      ;IF A TRAP TO 4 OCCURS COME HERE.
2710 011040 011602      UUU10:  MOV      (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
2711 011042 020227 010752      CMP      R2,#UUU2+2
2712 011046 001405      BEQ     UUU11          ;BRANCH IF YES.
2713 011050 020227 010754      CMP      R2,#UUU2+4
2714 011054 001402      BEQ     UUU11          ;BRANCH IF YES.
2715 011056 000137 042620      JMP     @#CPSPUR      ;OTHERWISE REPORT A SPURIOUS TRAP TO 4.
2716      ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4,
2717 011062 010237 001236      UUU11:  MOV      R2,@#STMP2
2718 011066 022626      CMP      (SP)+,(SP)+
2719 011070 104026      1$:     ERROR   26          ;BET FDST X ODD ADR
2720 011072 000416      BR      UUUDONE
2721

```

```

2722      ;REPORT R0 INCORRECT.
2723 011074 010037 001242      UUU15:  MOV      R0,@#STMP4
2724 011100 012737 011022 001240      MOV      #UUUA2+2,@#STMP3
2725 011106 104027      1$:     ERROR   27          ;R0 NOT INCREMENT PROPERLY
2726 011110 000407      BR      UUUDONE
2727

```

```

2728      ;REPORT BAD DATA.
2729 011112 012737 011004 001242      UUU20:  MOV      #UUUBF0,@#STMP4
2730 011120 012737 011030 001240      MOV      #UUUTP1,@#STMP3
2731 011126 104030      1$:     ERROR   30          ;BAD DATA
2732 011130      UUUDONE:
2733 011130 104412      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
2734      ;SEE IF THE USER HAS EXPRESSED
2735      ;THE DESIRE TO CHANGE THE SOFTWARE
2736      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2737      ;THE USER TYPED CONTROL G?).
2738

```

```

2739      ;)*****
2740      ;*TEST 10      FDST MODE 6, INDEX MODE, TEST

```

```

2741 ;*
2742 ;*THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.
2743 ;*
2744 ;*****
2745 011132 000004 TST10: SCOPE
2746
2747 011134 VVV1:
2748 011134 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
2749 011136 012700 000200 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE,
2750 011142 170100 LDFPS R0
2751 011144 012701 011254 MOV #VVVBF0,R1 ;SET UP THE OUT PUT DATA BUFFER,
2752 011150 012700 177777 MOV #-1,R0
2753 011154 012702 000004 MOV #4,R2
2754 011160 010021 1$: MOV R0,(R1)+
2755 011162 077202 SOB R2,1$
2756 011164 012737 011274 000004 MOV #VVV10,@ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR,
2757 011172 012700 011264 MOV #VVVTP1,R0 ;SET UP AC0,
2758 011176 172410 LDD (R0),AC0
2759 011200 012737 011216 001236 MOV #VVV2,@$TMP2
2760 011206 012700 003353 MOV #VVVBF0-5701,R0 ;SET UP THE DESTINATION ADDRESS,
2761 011212 012701 000001 MOV #1,R1
2762 011216 174060 005701 VVV2: STD AC0,5701(R0) ;TEST INSTRUCTION,
2763
2764 011222 020027 003353 CMP R0,#VVVBF0-5701 ;SEE IF R0 WAS MODIFIED,
2765 011226 001040 BNE VVV15 ;BRANCH IF INCORRECT,
2766 011230 012702 011254 MOV #VVVBF0,R2 ;WAS THE OUTPUT DATA CORRECT,
2767 011234 012703 011264 MOV #VVVTP1,R3
2768 011240 012704 000004 MOV #4,R4
2769 011244 022223 1$: CMP (R2)+,(R3)+
2770 011246 001037 BNE VVV20 ;BRANCH IF INCORRECT DATA,
2771 011250 077403 SOB R4,1$
2772 011252 000444 BR VVVDONE
2773 011254 177777 VVVBF0: -1
2774 011256 177777 -1
2775 011260 177777 -1
2776 011262 177777 -1
2777 011264 030313 VVVTP1: 30313
2778 011266 023334 23334
2779 011270 035363 35363
2780 011272 074041 74041
2781
2782 ;COME HERE AFTER A TRAP THROUGH VECTOR 4,
2783 011274 011602 VVV10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR,
2784 011276 020227 011220 CMP R2,#VVV2+2
2785 011302 001405 BEQ VVV11 ;BRANCH IF YES,
2786 011304 020227 011222 CMP R2,#VVV2+4
2787 011310 001402 BEQ VVV11 ;BRANCH IF YES,
2788 011312 000137 042564 JMP @#FPPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,
2789 ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4,
2790 011316 010237 001236 VVV11: MOV R2,@$TMP2
2791 011322 022626 CMP (SP)+,(SP)+
2792 011324 104031 1$: ERROR 31 ;FDST FORK X ODD ADD
2793 011326 000416 BR VVVDONE
2794
2795 ;REPORT R0 MODIFIED,
2796 011330 010037 001242 VVV15: MOV R0,@$TMP4

```

```

2797 011334 012737 003353 001240      MOV      #VVVBF0-5701,@$TMP3
2798 011342 104032      1$:      ERROR      32      ;R0 MODIFIED!
2799 011344 000407      BR        VVVDONE
2800
2801      ;REPORT INCORRECT DATA,
2802 011346 012737 011254 001240 VVV20:   MOV      #VVVBF0,@$TMP3
2803 011354 012737 011264 001242      MOV      #VVVTP1,@$TMP4
2804 011362 104033      1$:      ERROR      33      ;BAD DATA
2805 011364      VVVDONE:
2806 011364 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
2807      ;SEE IF THE USER HAS EXPRESSED
2808      ;THE DESIRE TO CHANGE THE SOFTWARE
2809      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
2810      ;THE USER TYPED CONTROL G?).
2811
2812      ;*****
2813      ;*TEST 11      FDST MODE 7, INDEX DEFERRED MODE, TEST
2814      ;*
2815      ;*THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD,
2816      ;*
2817      ;*****
2818 011366 000004      TST11:   SCOPE
2819
2820 011370      WWW1:
2821 011370 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS,
2822 011372 012700 000200      MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE,
2823 011376 170100      LDFPS     R0
2824 011400 012701 011516      MOV      #WWWBF0,R1      ;SET UP THE OUTPUT DATA BUFFER,
2825 011404 012700 177777      MOV      #-1,R0
2826 011410 012702 000004      MOV      #4,R2
2827 011414 010021      1$:      MOV      R0,(R1)+
2828 011416 077202      SOB      R2,1$
2829 011420 012737 011546 000004      MOV      #WWW10,@$ERRVECT ;SET UP FOR TRAPS TO 4,
2830 011426 012700 011526      MOV      #WWWTP1,R0      ;SET UP AC0,
2831 011432 172410      LDD      (R0),AC0
2832 011434 012737 011460 001236      MOV      #WWW2,@$TMP2
2833 011442 012700 003635      MOV      #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS,
2834 011446 012701 000001      MOV      #1,R1
2835 011452 012737 011516 011536      MOV      #WWWBF0,@$WWWBF1
2836 011460 174070 005701      WWW2:   STD      AC0,@5701(R0) ;TEST INSTRUCTION.
2837
2838 011464 020027 003635      CMP      R0,#WWWBF1-5701 ;IS R0 CORRECT?
2839 011470 001044      BNE      WWW15      ;BRANCH IF INCORRECT,
2840 011472 012702 011516      MOV      #WWWBF0,R2      ;WAS THE DATA OUTPUT CORRECTLY?
2841 011476 012703 011526      MOV      #WWWTP1,R3
2842 011502 012704 000004      MOV      #4,R4
2843 011506 022223      1$:      CMP      (R2)+,(R3)+
2844 011510 001043      BNE      WWW20      ;BRANCH IF DATA IS INCORRECT,
2845 011512 077403      SOB      R4,1$
2846 011514 000450      BR        WWWDONE
2847 011516 177777      WWWBF0:  -1
2848 011520 177777      -1
2849 011522 177777      -1
2850 011524 177777      -1
2851 011526 041424      WWWTP1:  41424
2852 011530 034445      34445
  
```

2853 011532 046475  
 2854 011534 051525  
 2855 011536 177777  
 2856 011540 177777  
 2857 011542 177777  
 2858 011544 177777  
 2859  
 2860  
 2861 011546 011602  
 2862 011550 020227 011462  
 2863 011554 001405  
 2864 011556 020227 011464  
 2865 011562 001402  
 2866 011564 000137 042564  
 2867  
 2868 011570 010237 001236  
 2869 011574 022626  
 2870 011576 104034  
 2871 011600 000416  
 2872  
 2873  
 2874 011602 010037 001242  
 2875 011606 012737 003615 001240  
 2876 011614 104035  
 2877 011616 000407  
 2878  
 2879  
 2880 011620 012737 011516 001240  
 2881 011626 012737 011526 001242  
 2882 011634 104036  
 2883 011636  
 2884 011636 104412  
 2885  
 2886  
 2887  
 2888  
 2889  
 2890  
 2891  
 2892  
 2893  
 2894  
 2895  
 2896 011640 000004  
 2897  
 2898  
 2899 011642  
 2900 011642 104413  
 2901 011644 004767 000330  
 2902 011650 000000  
 2903 011652 000000  
 2904 011654 000000  
 2905 011656 000000  
 2906 011660 000000  
 2907 011662 000000  
 2908 011664 000000

46475  
 051525  
 WWWBF1: -1  
 -1  
 -1  
 -1  
 ;TRAP THROUGH 4 TO HERE.  
 WWW10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR,  
 CMP R2,#WWW2+2  
 BEQ WWW11 ;BRANCH IF YES,  
 CMP R2,#WWW2+4  
 BEQ WWW11 ;BRANCH IF YES,  
 JMP @#FPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,  
 ;REPORT FAILURE OF FDST FORK RESULTED IN AN ODD ADDRESS TRAP TO 4,  
 WWW11: MOV R2,@\$STMP2  
 CMP (SP)+,(SP)+  
 1\$: ERROR 34 ;FDST FORK X ODD ADD  
 BR WWWDONE  
 ;REPORT R0 MODIFIED,  
 WWW15: MOV R0,@\$STMP4  
 MOV #WWWBF0-5701,@\$STMP3  
 1\$: ERROR 35 ;R0 MODIFIED!  
 BR WWWDONE  
 ;REPORT DATA INCORRECT  
 WWW20: MOV #WWWBF0,@\$STMP3  
 MOV #WWWTP1,@\$STMP4  
 1\$: ERROR 36 ;BAD DATA  
 WWWDONE:  
 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
 ;SEE IF THE USER HAS EXPRESSED  
 ;THE DESIRE TO CHANGE THE SOFTWARE  
 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
 ;THE USER TYPED CONTROL G?).  
 ;\*\*\*\*\*  
 ;\*TEST 12 STCFD TEST  
 ;\*  
 ;\*THIS IS A TEST OF THE STCFD INSTRUCTION.  
 ;\*  
 ;\*\*\*\*\*  
 TST12: SCOPE  
 ;AC=0  
 XXX1:  
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
 JSR PC,STCFDS  
 1\$: 0 ;AC  
 0  
 0  
 0  
 2\$: 0 ;RES  
 0  
 0

2909	011666	000000		0	
2910	011670	000000		0	;ERROR RES.
2911	011672	000000		0	
2912	011674	177777		-1	
2913	011676	177777		-1	
2914	011700	047000		47000	;FPS BEFORE EXECUTION.
2915	011702	047004		47004	;FPS AFTER EXECUTION.
2916	011704	177777		-1	;FEC
2917	011706	147004		147004	;ERROR FPS.
2918	011710	104042		ERROR 42	;FDFL<---FDFLXST 767
2919	011712	000401		BR 65	
2920	011714	104043		ERROR 43	;BUT EZBT X ST560 TO 061 INTO 261
2921	011716				
2922					
2923	011716				
2924	011716	104413		LPERR	;SET UP THE LOOP ON ERROR ADDRESS.
2925	011720	004767	000254	JSR PC,STCFDS	
2926	011724	017203		17203	;AC
2927	011726	142536		142536	
2928	011730	047506		47506	
2929	011732	172031		172031	
2930	011734	017203		17203	;RES
2931	011736	142536		142536	
2932	011740	000000		0	
2933	011742	000000		0	
2934	011744	017203		17203	;ERROR RES.
2935	011746	142536		142536	
2936	011750	047506		47506	
2937	011752	172031		172031	
2938	011754	040000		40000	;FPS BEFORE EXECUTION.
2939	011756	040000		40000	;FPS AFTER EXECUTION.
2940	011760	177777		-1	;FEC
2941	011762	177777		-1	;ERROR FPS.
2942	011764	104044		ERROR 44	;X11(1,0)<---0 X ST766
2943	011766	000401		BR 65	
2944	011770	104040		ERROR 40	
2945	011772				
2946					
2947	011772				
2948	011772	104413		LPERR	;SET UP THE LOOP ON ERROR ADDRESS.
2949	011774	004767	000200	JSR PC,STCFDS	
2950	012000	050717		50717	;AC
2951	012002	027374		27374	
2952	012004	075767		75767	
2953	012006	077071		77071	
2954	012010	050717		50717	;RES
2955	012012	027374		27374	
2956	012014	000000		0	
2957	012016	000000		0	
2958	012020	000000		0	;ERROR RES.
2959	012022	000000		0	
2960	012024	000000		0	
2961	012026	000000		0	
2962	012030	047000		47000	;FPS BEFORE EXECUTION.
2963	012032	047000		47000	;FPS AFTER EXECUTION.
2964	012034	177777		-1	;FEC



```

3021
3022 ;THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
3023 ;THE STCFD INSTRUCTION AND CHECK THE RESULTS. A CALL
3024 ;TO IT IS MADE THUS:
3025 ;
3026 ; JSR PC,@#STCFDS
3027 ; ACARG: .WORD X,X,X,X ;AC OPERAND
3028 ; RES: .WORD X,X,X,X ;EXPECTED RESULT
3029 ; ERRES: .WORD X,X,X,X ;ERROR RESULT
3030 ; FPSB: .WORD X ;FPS BEFORE EXECUTION
3031 ; FPSA: .WORD X ;FPS AFTER EXECUTION
3032 ; FEC: .WORD X ;EXPECTED FEC
3033 ; ERFPS: .WORD X ;ERROR FPS,
3034 ; ERR1: ERROR X ;DATA ERROR,
3035 ; BR CONT
3036 ; ERR2: ERROR X ;FPS ERROR,
3037 ; CONT: ;RETURN ADDRESS
3038 ;
3039 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR), THEN
3040 ;THE STCFD INSTRUCTION IS EXECUTED.
3041 ;THE RESULT IS CHECKED AGAINST RES, IF THE RESULT IS CORRECT THEN THE FPS IS
3042 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
3043 ;TO THE CALLING ROUTINE AT CONT, IF THE FPS IS BAD STCFDS
3044 ;COMPARE IT TO ERROR FPS, IF THIS MATCHES THEN STCFDS WILL RETURN
3045 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
3046 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT, IF THE RESULT OF THE
3047 ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
3048 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN
3049 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
3050 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
3051 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
3052 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
3053
3054 012200 012601 STCFDS: MOV (SP)+,R1 ;PICK UP THE POINTER TO THE OPERANDS.
3055 012202 012700 000200 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE,
3056 012206 170100 LDFPS R0
3057 012210 010100 MOV R1,R0 ;LOAD AC0.
3058 012212 172410 LDD (R0),AC0
3059 012214 012700 177777 MOV #-1,R0 ;FILL THE OUTPUT BUFFER WITH -1'S,
3060 012220 012702 012462 MOV #STCFT,R2
3061 012224 012703 000004 MOV #4,R3
3062 012230 010022 1$: MOV R0,(R2)+
3063 012232 077302 SOB R3,1$
3064 012234 016100 000030 MOV 30(R1),R0 ;LOAD THE FPS.
3065 012240 170100 LDFPS R0
3066 012242 012737 012254 001236 MOV #2$,@#TMP2
3067 012250 012700 012462 MOV #STCFT,R0 ;SET UP THE DESTINATION ADDRESS.
3068 012254 176010 2$: STCFD AC0,(R0) ;TEST INSTRUCTION.
3069
3070 012256 170204 STFPS R4 ;GET THE FPS.
3071 012260 170305 STST R5 ;GET THE FEC.
3072 012262 010102 MOV R1,R2 ;SAVE THE DATA IN CASE OF ERROR.
3073 012264 010237 001240 MOV R2,@#TMP3
3074 012270 062702 000010 ADD #10,R2
3075 012274 010237 001244 MOV R2,@#TMP5
3076 012300 012737 012462 001242 MOV #STCFT,@#TMP4

```



```

3077 012306 010437 001250      MOV      R4,0##$TMP7
3078 012312 016137 000032 001252  MOV      32(R1),0##$TMP10
3079
3080 012320 010102      MOV      R1,R2          ;CHECK THE RESULT,
3081 012322 062702 000010      ADD      #10,R2
3082 012326 012703 012462      MOV      #STCFT,R3
3083 012332 012700 000004      MOV      #4,R0
3084 012336 022223      3$:      CMP      (R2)+,(R3)+
3085 012340 001014      BNE      15$          ;BRANCH IF INCORRECT,
3086 012342 077003      SOB      R0,3$
3087
3088 012344 016102 000032      MOV      32(R1),R2
3089 012350 020204      CMP      R2,R4          ;IS THE FPS CORRECT?
3090 012352 001025      BNE      20$          ;BRANCH IF FPS INCORRECT,
3091 012354 005702      TST      R2            ;IF EXPECTED FPS IS NEGATIVE, THEN
3092 012356 100003      BPL      4$            ;GO AHEAD AND CHECK THE FEC,
3093 012360 026105 000036      CMP      36(R1),R5
3094 012364 001027      BNE      25$          ;BRANCH IF FEC IS INCORRECT,
3095 012366 000161 000046      4$:      JMP      46(R1)        ;RETURN,
3096
3097      ;RESULT INCORRECT:
3098 012372 010102      15$:     MOV      R1,R2          ;SEE IF ERROR WAS ANTICIPATED,
3099 012374 062702 000020      ADD      #20,R2
3100 012400 012703 012462      MOV      #STCFT,R3
3101 012404 012700 000004      MOV      #4,R0
3102 012410 022223      16$:     CMP      (R2)+,(R3)+
3103 012412 001003      BNE      17$          ;BRANCH IF NOT ANTICIPATED,
3104 012414 077003      SOB      R0,16$
3105 012416 000161 000040      JMP      40(R1)        ;IF ERROR WAS ANTICIPATED RETURN,
3106      ;OTHERWISE REPORT RESULT INCORRECT HERE,
3107 012422      17$:
3108 012422 104037      18$:     ERROR   37          ;DATA ERROR
3109 012424 000760      BR       4$
3110
3111      ;FPS INCORRECT:
3112 012426 020461 000034      20$:     CMP      R4,34(R1)    ;WAS THE ERROR ANTICIPATED,
3113 012432 001002      BNE      21$          ;BRANCH IF NOT ANTICIPATED,
3114 012434 000161 000044      JMP      44(R1)        ;IF IT WAS ANTICIPATED RETURN,
3115
3116      ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE,
3117 012440      21$:
3118 012440 104040      22$:     ERROR   40          ;FPS X
3119 012442 000751      BR       4$
3120
3121      ;REPORT FEC INCORRECT:
3122 012444 016137 000036 001256  25$:     MOV      36(R1),0##$TMP12
3123 012452 010537 001254      MOV      R5,0##$TMP11
3124 012456 104041      26$:     ERROR   41          ;FEC X
3125 012460 000742      BR       4$
3126 012462 177777 177777 177777  STCFT:   -1,-1,-1,-1
3127 012470 177777
3128 012472      XXXDONE:
3129 012472 104412      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3130      ;SEE IF THE USER HAS EXPRESSED
3131      ;THE DESIRE TO CHANGE THE SOFTWARE
3132      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS

```

```

3133                                     ;THE USER TYPED CONTROL G?).
3134
3135 ;*****
3136 ;*TEST 13          STCDF TEST
3137 ;*
3138 ;*THIS IS A TEST OF THE STCDF INSTRUCTION.
3139 ;*
3140 ;*****
3141 012474 000004 TST13: SCOPE
3142
3143 ;AC=0
3144 012476 YYY1:
3145 012476 104413 LPERR
3146 012500 004767 000330 JSR PC,STCDFs ;SET UP THE LOOP ON ERROR ADDRESS,
3147 012504 000000 1$: 0 ;AC
3148 012506 000000 0
3149 012510 000000 0
3150 012512 000000 0
3151 012514 000000 2$: 0 ;RES
3152 012516 000000 0
3153 012520 177777 -1
3154 012522 177777 -1
3155 012524 000000 3$: 0 ;ERROR RES.
3156 012526 000000 0
3157 012530 000000 0
3158 012532 000000 0
3159 012534 047200 4$: 47200 ;FPS BEFORE EXECUTION.
3160 012536 047204 47204 ;FPS AFTER EXECUTION,
3161 012540 177777 -1 ;FEC
3162 012542 177777 -1 ;ERROR FPS.
3163 012544 104054 5$: ERROR 54 ;FDFL<---FDFL X ST767
3164 012546 000401 BR 6$
3165 012550 104052 ERROR 52 ;FPS INCORRECT.
3166 012552 6$:
3167 ;
3168 012552 YYY2:
3169 012552 104413 LPERR
3170 012554 004767 000254 JSR PC,STCDFs ;SET UP THE LOOP ON ERROR ADDRESS,
3171 012560 067574 1$: 67574 ;AC0
3172 012562 073727 73727
3173 012564 170777 170777
3174 012566 067574 67574
3175 012570 067574 2$: 67574 ;RES
3176 012572 073730 73730
3177 012574 177777 -1
3178 012576 177777 -1
3179 012600 067574 3$: 67574 ;ERROR RES.
3180 012602 073727 73727
3181 012604 177777 -1
3182 012606 177777 -1
3183 012610 040200 4$: 40200 ;FPS BEFORE EXECUTION.
3184 012612 040200 40200 ;FPS AFTER EXECUTION,
3185 012614 177777 -1 ;FEC
3186 012616 177777 -1 ;ERROR FPS.
3187 012620 104055 5$: ERROR 55 ;EITHER ROUND FAILED OR WENT TO 766 X1(1,0)<---0 INTO 76
3188 012622 000401 BR 6$
  
```

3189	012624	104052			ERROR	52	
3190	012626			6s:			
3191				;			
3192	012626			YYY3:			
3193	012626	104413			LPERR		;SET UP THE LOOP ON ERROR ADDRESS,
3194	012630	004767	000200		JSR	PC,STCDFs	
3195	012634	077777		1s:	77777		;AC0
3196	012636	177777			-1		
3197	012640	100000			100000		
3198	012642	000000			0		
3199	012644	000000		2s:	0		;RES
3200	012646	000000			0		
3201	012650	177777			-1		
3202	012652	177777			-1		
3203	012654	077777		3s:	77777		;ERROR RES,
3204	012656	177777			-1		
3205	012660	177777			-1		
3206	012662	177777			-1		
3207	012664	040200		4s:	40200		;FPS BEFORE EXECUTION,
3208	012666	040206			40206		;FPS AFTER EXECUTION,
3209	012670	177777			-1		;FEC
3210	012672	040204			40204		;ERROR FPS,
3211	012674	104055		5s:	ERROR	55	
3212	012676	000401			BR	6s	
3213	012700	104056			ERROR	56	;BUT EZBT X ST421 TO 062 INTO 262
3214	012702			6s:			
3215				;			
3216	012702			YYY4:			
3217	012702	104413			LPERR		;SET UP THE LOOP ON ERROR ADDRESS,
3218	012704	004767	000124		JSR	PC,STCDFs	
3219	012710	077777		1s:	77777		;AC0
3220	012712	177777			-1		
3221	012714	100000			100000		
3222	012716	000000			0		
3223	012720	000000		2s:	0		;RES
3224	012722	000000			0		
3225	012724	177777			-1		
3226	012726	177777			-1		
3227	012730	077777		3s:	77777		;ERROR RES,
3228	012732	177777			-1		
3229	012734	177777			-1		
3230	012736	177777			-1		
3231	012740	040200		4s:	40200		;FPS BEFORE EXECUTION,
3232	012742	040206			40206		;FPS AFTER EXECUTION,
3233	012744	177777			-1		;FEC
3234	012746	140206			140206		;ERROR FPS,
3235	012750	104055		5s:	ERROR	55	
3236	012752	000401			BR	6s	
3237	012754	104057			ERROR	57	;BUT FIV ST262 TO 123 INTO 103
3238	012756			6s:			
3239				;			
3240	012756			YYY5:			
3241	012756	104413			LPERR		;SET UP THE LOOP ON ERROR ADDRESS,
3242	012760	004767	000050		JSR	PC,STCDFs	
3243	012764	177777		1s:	177777		;AC0
3244	012766	177777			-1		

3245	012770	100000		100000	
3246	012772	000000		0	
3247	012774	100000	2s:	100000	;RES
3248	012776	000000		0	
3249	013000	177777		-1	
3250	013002	177777		-1	
3251	013004	000000	3s:	0	;ERROR RES.
3252	013006	000000		0	
3253	013010	177777		-1	
3254	013012	177777		-1	
3255	013014	047200	4s:	47200	;FPS BEFORE EXECUTION.
3256	013016	147216		147216	;FPS AFTER EXECUTION.
3257	013020	000010		10	;FEC
3258	013022	047206		47206	;ERROR FPS.
3259	013024	104060	5s:	ERROR 60	;BUT FIV ST262 FAIL TO 103 INT 123
3260	013026	000401		BR 66	
3261	013030	104061		ERROR 61	;BUT FLAG ST 147 X TO ST 361 INTO 365
3262	013032	000535	6s:	BR	YYDONE

3263 ;THIS SUBROUTINE, STCDF, IS USED TO SET UP THE OPERANDS, EXECUTE  
 3264 ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL  
 3265 ;TO IT IS MADE THUS:  
 3266 ;

3267	;	JSR	PC, @#STCFDS	
3268	;	ACARG:	.WORD X,X,X,X	;AC OPERAND
3269	;	RES:	.WORD X,X,X,X	;EXPECTED RESULT
3270	;	ERRES:	.WORD X,X,X,X	;ERROR RESULT
3271	;	FPSB:	.WORD X	;FPS BEFORE EXECUTION
3272	;	FPSA:	.WORD X	;FPS AFTER EXECUTION
3273	;	FEC:	.WORD X	;EXPECTED FEC
3274	;	ERFPS:	.WORD X	;ERROR FPS.
3275	;	ERR1:	ERROR X	;DATA ERROR.
3276	;		BR CONT	
3277	;	ERR2:	ERROR X	;FPS ERROR.
3278	;	CONT:		;RETURN ADDRESS

3279 ;  
 3280 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN  
 3281 ;THE STCFD INSTRUCTION IS EXECUTED.  
 3282 ;THE RESULT IS CHECKED AGAINST RES, IF THE RESULT IS CORRECT THEN THE FPS IS  
 3283 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL  
 3284 ;TO THE CALLING ROUTINE AT CONT, IF THE FPS IS BAD STCFDS  
 3285 ;COMPARE IT TO ERROR FPS, IF THIS MATCHES THEN STCFDS WILL RETURN  
 3286 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF  
 3287 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT, IF THE RESULT OF THE  
 3288 ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE  
 3289 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN  
 3290 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS  
 3291 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE  
 3292 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL  
 3293 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.  
 3294 ;

3295	013034	012601	STCDFs:	MOV	(SP)+,R1	;PICK UP THE POINTER TO THE OPERANDS.
3296	013036	012700		MOV	#200,R0	;ENTER DOUBLE FLOATING MODE.
3297	013042	170100		LDFPS	R0	
3298	013044	010100		MOV	R1,R0	;LOAD AC0.
3299	013046	172410		LDD	(R0),AC0	
3300	013050	012700		MOV	#-1,R0	;FILL THE OUTPUT BUFFER WITH -1'S.

3301	013054	012702	013316		MOV	#STCDT,R2	
3302	013060	012703	000004		MOV	#4,R3	
3303	013064	010022		1\$:	MOV	R0,(R2)+	
3304	013066	077302			SOB	R3,1\$	
3305	013070	016100	000030		MOV	30(R1),R0	;LOAD THE FPS.
3306	013074	170100			LDFPS	R0	
3307	013076	012737	013110	001236	MOV	#2\$,@#STMP2	
3308	013104	012700	013316		MOV	#STCDT,R0	;SET UP THE DESTINATION ADDRESS.
3309	013110	176010		2\$:	STCDF	AC0,(R0)	;TEST INSTRUCTION.
3310							
3311	013112	170204			STFPS	R4	;GET THE FPS.
3312	013114	170305			STST	R5	;GET THE FEC.
3313	013116	010102			MOV	R1,R2	;SAVE THE DATA IN CASE OF ERROR.
3314	013120	010237	001240		MOV	R2,@#STMP3	
3315	013124	062702	000010		ADD	#10,R2	
3316	013130	010237	001244		MOV	R2,@#STMP5	
3317	013134	012737	013316	001242	MOV	#STCDT,@#STMP4	
3318	013142	010437	001250		MOV	R4,@#STMP7	
3319	013146	016137	000032	001252	MOV	32(R1),@#STMP10	
3320							
3321	013154	010102			MOV	R1,R2	;CHECK THE RESULT.
3322	013156	062702	000010		ADD	#10,R2	
3323	013162	012703	013316		MOV	#STCDT,R3	
3324	013166	012700	000004		MOV	#4,R0	
3325	013172	022223		3\$:	CMP	(R2)+,(R3)+	
3326	013174	001014			BNE	15\$	;BRANCH IF INCORRECT.
3327	013176	077003			SOB	R0,3\$	
3328							
3329	013200	016102	000032		MOV	32(R1),R2	
3330	013204	020204			CMP	R2,R4	;IS THE FPS CORRECT?
3331	013206	001025			BNE	20\$	;BRANCH IF FPS INCORRECT.
3332	013210	005702			TST	R2	;IF EXPECTED FPS IS NEGATIVE, THEN
3333	013212	100003			BPL	4\$	;GO AHEAD AND CHECK THE FEC.
3334	013214	026105	000034		CMP	34(R1),R5	
3335	013220	001027			BNE	25\$	;BRANCH IF FEC IS INCORRECT.
3336	013222	000161	000046	4\$:	JMP	46(R1)	;RETURN.
3337							
3338							
3339	013226	010102					
3340	013230	062702	000020		MOV	R1,R2	;SEE IF ERROR WAS ANTICIPATED.
3341	013234	012703	013316		ADD	#20,R2	
3342	013240	012700	000004		MOV	#STCDT,R3	
3343	013244	022223		16\$:	MOV	#4,R0	
3344	013246	001003			CMP	(R2)+,(R3)+	
3345	013250	077003			BNE	17\$	;BRANCH IF NOT ANTICIPATED.
3346	013252	000161	000040		SOB	R0,16\$	
3347					JMP	40(R1)	;IF ERROR WAS ANTICIPATED RETURN.
3348	013256						;OTHERWISE REPORT RESULT INCORRECT HERE.
3349	013256	104051		17\$:			
3350	013260	000760		18\$:	ERROR	51	;DATA ERROR
3351					BR	4\$	
3352							
3353	013262	020461	000034				
3354	013266	001002		20\$:	CMP	R4,34(R1)	;WAS THE ERROR ANTICIPATED.
3355	013270	000161	000044		BNE	21\$	;BRANCH IF NOT ANTICIPATED.
3356					JMP	44(R1)	;IF IT WAS ANTICIPATED RETURN.

```

3357 ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE,
3358 013274 21s:
3359 013274 104052 22s: ERROR 52 ;FPS X
3360 013276 000751 BR 4s
3361
3362 ;REPORT FEC INCORRECT:
3363 013300 016137 000036 001256 25s: MOV 36(R1),@#TMP12
3364 013306 010537 001254 MOV R5,@#TMP11
3365 013312 104053 26s: ERROR 53 ;FEC X
3366 013314 000742 BR 4s
3367 013316 177777 177777 177777 STCDT: -1,-1,-1,-1
3368 013324 177777
3369 013326
3370 013326 104412 YYDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3371 ;SEE IF THE USER HAS EXPRESSED
3372 ;THE DESIRE TO CHANGE THE SOFTWARE
3373 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3374 ;THE USER TYPED CONTROL G?),
3375 ;*****
3376 ;*TEST 14 STCFD WITH ILLEGAL ACCUMULATOR TEST
3377 ;*
3378 ;*THIS TEST STCFD WITH ILLEGAL AC 6,
3379 ;*
3380 ;*****
3381 013330 000004 TST14: SCOPE
3382
3383 013332 ZZZ1:
3384 013332 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
3385 013334 012700 040000 MOV #40000,R0 ;DISSABLE INTERRUPTS,
3386 013340 170100 LDFPS R0
3387 013342 012737 013350 001236 MOV #ZZZ2,@#TMP2
3388 013350 176006 ZZZ2: STCFD AC0,AC6 ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR,
3389
3390 013352 170204 STFPS R4 ;GET FPS,
3391 013354 170305 STST R5 ;GET FEC,
3392 013356 020427 140000 CMP R4,#140000 ;IS FPS CORRECT?
3393 013362 001004 BNE ZZZ10 ;BRANCH IF INCORRECT FPS,
3394 013364 022705 000002 CMP #2,R5 ;IS FEC CORRECT?
3395 013370 001010 BNE ZZZ15 ;BRANCH IF INCORRECT,
3396 013372 000415 BR ZZZDONE
3397
3398 ;REPORT FPS INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR,
3399 013374 010437 001242 ZZZ10: MOV R4,@#TMP4
3400 013400 012737 140000 001240 MOV #140000,@#TMP3
3401 013406 104062 1s: ERROR 62 ;BUT FDST ST767 X TO 567 INTO 577
3402 013410 000406 BR ZZZDONE
3403
3404 ;REPORT FEC INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR,
3405 013412 010537 001242 ZZZ15: MOV R5,@#TMP4
3406 013416 012737 000002 001240 MOV #2,@#TMP3
3407 013424 104063 1s: ERROR 63 ;FEC<---2 ST577 X
3408 013426 ZZZDONE:
3409 013426 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3410 ;SEE IF THE USER HAS EXPRESSED
3411 ;THE DESIRE TO CHANGE THE SOFTWARE
3412 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS

```

```

3413                                     ;THE USER TYPED CONTROL G?).
3414
3415                                     ;*****
3416                                     ;*TEST 15      CLRD TEST
3417                                     ;*
3418                                     ;*THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.
3419                                     ;*
3420                                     ;*****
3421 013430 000004 TST15: SCOPE
3422 013432 AAB1:
3423 013432 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
3424 013434 012700 013620  MOV      #AABTP1,R0      ;SET UP OUTPUT BUFFER
3425 013440 012701 013610  MOV      #AABBF0,R1
3426 013444 012702 000004  MOV      #4,R2
3427 013450 012021          1$:      MOV      (R0)+,(R1)+
3428 013452 077202          SOB      R2,1$
3429 013454 012700 013610  MOV      #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS,
3430 013460 012701 000213  MOV      #213,R1        ;SET UP FPS.
3431 013464 170101          LDFPS     R1
3432 013466 012737 013474 001236  MOV      #2$,@#$TMP2
3433 013474 170410          2$:      CLRD     (R0)          ;TEST INSTRUCTION,
3434
3435 013476 170205          STFPS     R5          ;GET FPS.
3436 013500 012702 000004  MOV      #4,R2          ;SEE IF RESULT CLEAR, 0.
3437 013504 012701 013610  MOV      #AABBF0,R1
3438 013510 005721          3$:      TST     (R1)+
3439 013512 001010          BNE     AAB2          ;BRANCH IF RESULT INCORRECT, NOT 0.
3440 013514 077203          SOB     R2,3$
3441 013516 022705 000204  CMP     #204,R5        ;SEE IF FPS IS CORRECT,
3442 013522 001014          BNE     AAB3          ;BRANCH IF INCORRECT,
3443 013524 020027 013610  CMP     R0,#AABBF0    ;SEE IF R0 IS CORRECT,
3444 013530 001020          BNE     AAB4          ;BRANCH IF R0 IS INCORRECT,
3445 013532 000442          BR     AABDONE
3446
3447                                     ;RESULT NOT 0, REPORT ERROR.
3448 013534 012737 013610 001240  AAB2:  MOV      #AABBF0,@#$TMP3
3449 013542 012737 013630 001242  MOV      #AABTP2,@#$TMP4
3450 013550 104064          1$:      ERROR   64          ;BAD DATA = 0 X 11+ZERO ST770 X
3451 013552 000432          BR     AABDONE
3452
3453                                     ;REPORT FPS INCORRECT:
3454 013554 010437 001242  AAB3:  MOV      R4,@#$TMP4
3455 013560 012737 000204 001240  MOV      #204,@#$TMP3
3456 013566 104065          1$:      ERROR   65          ;BAD FPS
3457 013570 000423          BR     AABDONE
3458
3459                                     ;REPORT R0 INCORRECT.
3460 013572 010037 001242  AAB4:  MOV      R0,@#$TMP4
3461 013576 012737 013610 001240  MOV      #AABBF0,@#$TMP3
3462 013604 104066          1$:      ERROR   66
3463 013606 000414          BR     AABDONE
3464
3465                                     ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
3466 013610 073475 AABBF0: 73475
3467 013612 067707          67707
3468 013614 127347          127347

```

3469 013616 056770

56770

3470

;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.

3471 013620 073475

AABTP1: 73475

3472 013622 067707

67707

3473 013624 127347

127347

3474 013626 056770

56770

3475

;THIS IS THE EXPECTED DATA, RESULT:

3476 013630 000000

AABTP2: 0

3477 013632 000000

0

3478 013634 000000

0

3479 013636 000000

0

3480 013640

AABDONE:

3481 013640 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?),

3482

3483

3484

3485

3486

3487

;\*\*\*\*\*

3488

;\*TEST 16 CLR D WITH ILLEGAL ACCUMULATOR TEST

3489

;\*

3490

;\*THIS IS A TEST OF CLR D WITH ILLEGAL AC7,

3491

;\*

3492

;\*\*\*\*\*

3493 013642 000004

TST16: SCOPE

3494 013644

CCB1:

3495 013644 104413

LPERR

;SET UP THE LOOP ON ERROR ADDRESS,

3496 013646 012700 040200

MOV #40200,R0

;SET UP THE FPS, NO INTERRUPTS AND FD=1.

3497 013652 170100

LDFPS R0

3498 013654 012737 013662 001236

MOV #CCB2,0#STMP2

3499 013662 170407

CCB2: CLR D AC7

;TEST INSTRUCTION,

3500

3501 013664 170204

STFPS R4

;GET FPS.

3502 013666 170305

STST R5

;GET FEC.

3503 013670 020427 140200

CMP R4,#140200

;IS THE FPS CORRECT?

3504 013674 001004

BNE CCB10

;BRANCH IF FPS IS INCORRECT,

3505 013676 022705 000002

CMP #2,R5

;IS THE FEC CORRECT?

3506 013702 001010

BNE CCB15

;BRANCH IF FEC IS INCORRECT,

3507 013704 000415

BR CCBDONE

3508

3509

;REPORT INCORRECT FPS:

3510 013706 010437 001242

CCB10: MOV R4,0#STMP4

3511 013712 012737 140200 001240

MOV #140200,0#STMP3

3512 013720 104067

16: ERROR 67

;BUT FDST ST 700X TO 607 INTO 677

3513 013722 000406

BR CCBDONE

3514

3515

;REPORT INCORRECT FEC:

3516 013724 010537 001242

CCB15: MOV R5,0#STMP4

3517 013730 012737 000002 001240

MOV #2,0#STMP3

3518 013736 104070

16: ERROR 70

;FEC<---2 ST 677 X

3519 013740

CCBDONE:

3520 013740 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?),

3521

3522

3523

3524



```

3525
3526 ;*****
3527 ;*TEST 17      NEGf, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
3528 ;*
3529 ;*THIS IS A TEST OF THE SPECIAL
3530 ;*DEST FLOWS USING THE NEGd INST
3531 ;*WITH MODE ZERO AND ILLEGAL
3532 ;*AC7.
3533 ;*
3534 ;*****
3535 013742 000004 TST17: SCOPE
3536
3537 013744 VVB1:
3538 013744 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
3539 013746 012700 040200 MOV #40200,R0 ;SET UP THE FPS, FID=1 AND FD=1.
3540 013752 170100 LDFPS R0
3541 013754 012737 013762 001236 MOV #VVB2,@$TMP2
3542
3543 013762 170707 VVB2: NEGd AC7 ;TEST INSTRUCTION.
3544
3545 013764 170204 STFPS R4 ;GET FPS.
3546 013766 170305 STST R5 ;GET FEC.
3547
3548 013770 022704 140200 CMP #140200,R4 ;IS FPS CORRECT?
3549 013774 001004 BNE VVB10 ;BRANCH IF FPS IS INCORRECT,
3550 013776 022705 000002 CMP #2,R5 ;IS FEC CORRECT?
3551 014002 001010 BNE VVB15 ;BRANCH IF FEC IS INCORRECT.
3552 014004 000415 BR VVBDONE
3553
3554 ;REPORT INCORRECT FPS:
3555 014006 012737 140200 001240 VVB10: MOV #140200,@$TMP3
3556 014014 010437 001242 MOV R4,@$TMP4
3557 014020 104176 1$: ERROR 176 ;FPS BAD
3558 014022 000406 BR VVBDONE
3559
3560 ;REPORT FEC INCORRECT:
3561 014024 012737 000002 001240 VVB15: MOV #2,@$TMP3
3562 014032 010537 001242 MOV R5,@$TMP4
3563 014036 104177 1$: ERROR 177 ;FEC BAD
3564
3565 014040 VVBDONE:
3566 014040 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3567 ;SEE IF THE USER HAS EXPRESSED
3568 ;THE DESIRE TO CHANGE THE SOFTWARE
3569 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3570 ;THE USER TYPED CONTROL G?).
3571
3572 ;*****
3573 ;*TEST 20      NEGf, ABSF AND TSTF SOURCE MODE 0 TEST
3574 ;*
3575 ;*THIS IS A TEST THE NEGf, ABSF AND TSTF
3576 ;*SOURCE FLOWS. THE NEGd INSTRUCTION
3577 ;*IS USED TO TEST MODE 0
3578 ;*
3579 ;*****
3580 014042 000004 TST20: SCOPE

```

```

3581
3582 014044          DDB1:
3583 014044 104413  LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3584 014046 012700 000200  MOV #200,R0    ;SET FD MODE.
3585 014052 170100  LDFPS R0
3586 014054 012700 014216  MOV #DDBTP1,R0 ;SET UP AC0.
3587 014060 172410  LDD (R0),AC0  ;SET AC0 = 0
3588 014062 005000  CLR R0        ;CLEAR THE FPS.
3589 014064 170100  LDFPS R0
3590 014066 012700 014226  MOV #DDBTP2,R0 ;LOAD AC0 TO BE A FLOATING 0.
3591 014072 172410  LDF (R0),AC0 ;SET AC0=ZERO
3592                                ;FLOAT
3593 014074 012700 000201  MOV #201,R0    ;SET FD MODE.
3594 014100 170100  LDFPS R0
3595 014102 012737 014110 001236  MOV #DDB2,@#STMP2
3596
3597 014110 170700  DDB2: NEG0 AC0 ;TEST INSTRUCTION.
3598
3599 014112 170205  STFPS R5      ;GET FPS.
3600 014114 012700 000200  MOV #200,R0    ;SET FD MODE.
3601 014120 170100  LDFPS R0
3602 014122 012700 014236  MOV #DDBBF0,R0 ;GET THE RESULT OUT OF AC0.
3603 014126 174010  STD AC0,(R0) ;SEE IF THE RESULT IS CORRECT,
3604
3605 014130 012701 000004  MOV #4,R1
3606 014134 005720  1$: TST (R0)+
3607 014136 001005  BNE DDB5      ;BRANCH IF THE RESULT IS INCORRECT.
3608 014140 077103  SOB R1,1$
3609 014142 022705 000204  CMP #204,R5   ;IS THE FPS CORRECT?
3610 014146 001014  BNE DDB6      ;BRANCH IF THE FPS IS INCORRECT.
3611 014150 000442  BR DDBDONE
3612
3613 ;RESULT INCORRECT, REPORT FAILURE:
3614 014152 012737 014226 001242  DDB5: MOV #DDBTP2,@#STMP4 ;EXPECT D0
3615 014160 012737 014246 001240  MOV #DDBTP3,@#STMP3 ;PREV F0 IMPURE
3616 014166 012737 014236 001244  MOV #DDBBF0,@#STMP5 ;GOT
3617 014174 104071  1$: ERROR 71
3618 014176 000427  BR DDBDONE
3619
3620 ;REPORT FPS INCORRECT:
3621 014200 012737 000204 001240  DDB6: MOV #204,@#STMP3
3622 014206 010537 001242  MOV R5,@#STMP4
3623 014212 104072  1$: ERROR 72
3624 014214 000420  BR DDBDONE
3625
3626 ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
3627 014216 101112  DDBTP1: 101112
3628 014220 131415  131415
3629 014222 161710  161710
3630 014224 111213  111213
3631 014226 000000  DDBTP2: 0
3632 014230 000000  0
3633 014232 000000  0
3634 014234 000000  0
3635
3636 014236 177777  DDBBF0: -1
  
```

3637 014240 177777  
 3638 014242 177777  
 3639 014244 177777  
 3640 014246 000000  
 3641 014250 000000  
 3642 014252 161710  
 3643 014254 111213  
 3644  
 3645 014256  
 3646 014256 104412  
 3647  
 3648  
 3649  
 3650  
 3651  
 3652  
 3653  
 3654  
 3655  
 3656  
 3657  
 3658  
 3659  
 3660 014260 000004  
 3661  
 3662 014262  
 3663 014262 104413  
 3664 014264 012700 014372  
 3665 014270 012701 014422  
 3666 014274 012702 000004  
 3667 014300 012021  
 3668 014302 077202  
 3669 014304 012700 000200  
 3670 014310 170100  
 3671 014312 012700 014422  
 3672 014316 012737 014332 001236  
 3673 014324 012737 014432 000004  
 3674 014332 170710  
 3675  
 3676 014334 170205  
 3677 014336 012701 014422  
 3678 014342 012702 000004  
 3679 014346 005721  
 3680 014350 001046  
 3681 014352 077203  
 3682  
 3683 014354 020027 014422  
 3684 014360 001055  
 3685 014362 022705 000204  
 3686 014366 001061  
 3687 014370 000466  
 3688  
 3689  
 3690 014372 000177  
 3691 014374 167574  
 3692 014376 137271

-1  
 -1  
 -1  
 DDBT3: 0  
 0  
 161710  
 111213  
 DDBDONE:  
 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 21 NEGF, ABSF AND TSTF SOURCE MODE 1 TEST  
 ;\*  
 ;\*THIS IS A TEST THE NEGF, ABSF AND TSTF  
 ;\*SOURCE FLOWS, THE NEGD INSTRUCTION  
 ;\*IS USED TO TEST MODE 1  
 ;\*  
 ;)\*\*\*\*\*  
 TST21: SCOPE  
 EEB1:  
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 MOV #EEBTP1,R0 ;SET UP THE DATA BUFFER,  
 MOV #EEBBF1,R1  
 MOV #4,R2  
 1\$: MOV (R0)+,(R1)+  
 SOB R2,1\$  
 MOV #200,R0 ;SET FD MODE,  
 LDFPS R0  
 MOV #EEBBF1,R0 ;SET UP THE OPERAND ADDRESS,  
 MOV #EEB2,@\$TMP2  
 MOV #EEB10,@\$ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR,  
 EEB2: NEGD (R0) ;TEST INSTRUCTION.  
 STFPS R5 ;GET FPS,  
 MOV #EEBBF1,R1 ;SEE IF RESULT IS CORRECT,  
 MOV #4,R2  
 1\$: TST (R1)+  
 BNE EEB15 ;BRANCH IF NOT CORRECT,  
 SOB R2,1\$  
 CMP R0,#EEBBF1 ;IS R0 CORRECT?  
 BNE EEB20 ;BRANCH IF NOT CORRECT.  
 CMP #204,R5 ;IS THE FPS CORRECT?  
 BNE EEB25 ;BRANCH IF NOT CORRECT.  
 BR EEBDONE  
 ;THESE ARE TEST DATA TABLES AND A BUFFER,  
 EEBTP1: 177  
 167574  
 137271

3693 014400 107675  
 3694 014402 000000  
 3695 014404 000000  
 3696 014406 000000  
 3697 014410 000000  
 3698 014412 177777  
 3699 014414 177777  
 3700 014416 177777  
 3701 014420 177777  
 3702 014422 177777  
 3703 014424 177777  
 3704 014426 177777  
 3705 014430 177777  
 3706

107675  
 EEBTP2: 0  
 0  
 0  
 0  
 EEBBF0: -1  
 -1  
 -1  
 -1  
 EEBBF1: -1  
 -1  
 -1  
 -1

3707  
 3708 014432 011602  
 3709 014434 020227 014334  
 3710 014440 001405  
 3711 014442 020227 014336  
 3712 014446 001402  
 3713 014450 000137 042620  
 3714  
 3715 014454 022626  
 3716 014456 010237 001236  
 3717 014462 104107  
 3718 014464 000430  
 3719

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

3720  
 3721 014466 012737 014402 001242  
 3722 014474 012737 014372 001240  
 3723 014502 012737 014422 001244  
 3724 014510 104073  
 3725 014512 000415  
 3726

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

3727  
 3728 014514 012737 014422 001240  
 3729 014522 010037 001242  
 3730 014526 104074  
 3731 014530 000406  
 3732

;R0 INCORRECT:  
 EEB20: MOV #EEBBF1,@#STMP3  
 MOV R0,@#STMP4  
 1\$: ERROR 74 ;R0 BADX  
 BR EEBDONE

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

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

3738 014546  
 3739 014546 104412  
 3740

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

3741  
 3742  
 3743  
 3744  
 3745  
 3746  
 3747  
 3748

;\*\*\*\*\*  
 ;\*TEST 22 NEGF, ABSF AND TSTF SOURCE MODE 2 TEST  
 ;\*  
 ;\*THIS IS A TEST THE NEGF, ABSF AND TSTF

```

3749 ;*SOURCE FLOWS. THE ABSD INSTRUCTION
3750 ;*IS USED TO TEST MODE 2
3751 ;*
3752 ;)*****
3753 014550 000004 TST22: SCOPE
3754
3755 FFB1:
3756 014552 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3757 014554 012700 014662 MOV #FFBTP1,R0 ;SET UP THE DATA BUFFER.
3758 014560 012701 014712 MOV #FFBBF1,R1
3759 014564 012702 000004 MOV #4,R2
3760 014570 012021 1S: MOV (R0)+,(R1)+
3761 014572 077202 SOB R2,1$
3762 014574 012700 000200 MOV #200,R0 ;SET FD.
3763 014600 170100 LDFPS R0
3764 014602 012700 014712 MOV #FFBBF1,R0 ;SET UP THE OPERAND ADDRESS.
3765 014606 012737 014622 001236 MOV #FFB2,@$TMP2
3766 014614 012737 014722 000004 MOV #FFB10,@$ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR,
3767
3768 014622 170620 FFB2: ABSD (R0)+ ;TEST INSTRUCTION.
3769
3770 014624 170205 STFPS R5 ;GET FPS.
3771 014626 012701 014712 MOV #FFBBF1,R1 ;CHECK RESULT.
3772 014632 012702 000004 MOV #4,R2
3773 014636 005721 1S: TST (R1)+
3774 014640 001046 BNE FFB15 ;BRANCH IF INCORRECT.
3775 014642 077203 SOB R2,1$
3776
3777 014644 020027 014722 CMP R0,#FFBBF1+10 ;IS R0 CORRECT?
3778 014650 001055 BNE FFB20 ;BRANCH IF INCORRECT.
3779 014652 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
3780 014656 001061 BNE FFB25 ;BRANCH IF INCORRECT.
3781 014660 000466 BR FFBDONE
3782
3783 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3784 014662 000177 FFBTP1: 177
3785 014664 167574 167574
3786 014666 137271 137271
3787 014670 107675 107675
3788 014672 000000 FFBTP2: 0
3789 014674 000000 0
3790 014676 000000 0
3791 014700 000000 0
3792 014702 177777 FFBBF0: -1
3793 014704 177777 -1
3794 014706 177777 -1
3795 014710 177777 -1
3796 014712 177777 FFBBF1: -1
3797 014714 177777 -1
3798 014716 177777 -1
3799 014720 177777 -1
3800
3801 ;IF A TRAP TO 4 OCCURS COME HERE.
3802 014722 011602 FFB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
3803 014724 020227 014624 CMP R2,#FFB2+2
3804 014730 001405 BEQ 1$ ;BRANCH IF YES.

```

```

3805 014732 020227 014626          CMP      R2,#FFB2+4
3806 014736 001402                   BEQ      1$          ;BRANCH IF YES.
3807 014740 000137 042620          JMP      @#CPSPUR   ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
3808                                ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3809 014744 022626          1$:      CMP      (SP)+,(SP)+
3810 014746 010237 001236          MOV      R2,@#TMP2
3811 014752 104076          2$:      ERROR   76          ;ODD ADRES
3812 014754 000430          BR       FFBDONE    ;BUT FDSTX IN ST 771
3813
3814                                ;REPORT RESULT INCORRECT:
3815 014756 012737 014672 001240  FFB15:  MOV      #FFBTP2,@#TMP3
3816 014764 012737 014662 001242          MOV      #FFBTP1,@#TMP4
3817 014772 012737 014712 001244          MOV      #FFBBF1,@#TMP5
3818 015000 104077          1$:      ERROR   77          ;BAD DATA X11*0 ST 312X
3819 015002 000415          BR       FFBDONE
3820
3821                                ;REPORT R0 INCORRECT:
3822 015004 012737 014716 001240  FFB20:  MOV      #FFBBF1+4,@#TMP3
3823 015012 010037 001242          MOV      R0,@#TMP4
3824 015016 104100          1$:      ERROR   100        ;R0 BADX
3825 015020 000406          BR       FFBDONE
3826
3827                                ;REPORT FPS INCORRECT:
3828 015022 010537 001240  FFB25:  MOV      R5,@#TMP3
3829 015026 012737 000204 001244          MOV      #204,@#TMP5
3830 015034 104101          1$:      ERROR   101        ;FPS X
3831
3832 015036          FFBDONE:
3833 015036 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3834                                ;SEE IF THE USER HAS EXPRESSED
3835                                ;THE DESIRE TO CHANGE THE SOFTWARE
3836                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3837                                ;THE USER TYPED CONTROL G?),
3838                                ;*****
3839                                ;*TEST 23      NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
3840                                ;*
3841                                ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
3842                                ;*SOURCE FLOWS, THE ABSD INSTRUCTION
3843                                ;*IS USED TO TEST MODE 4
3844                                ;*
3845                                ;)*****
3846 015040 000004  TST23:  SCOPE
3847
3848 015042          GGB1:
3849 015042 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3850 015044 012700 015152          MOV      #GGBTP1,R0          ;SET UP THE DATA BUFFER.
3851 015050 012701 015172          MOV      #GGBBF0,R1
3852 015054 012702 000004          MOV      #4,R2
3853 015060 012021          1$:      MOV      (R0)+,(R1)+
3854 015062 077202          SOB      R2,1$
3855 015064 012700 000200          MOV      #200,R0          ;SET FD.
3856 015070 170100          LDFPS      R0
3857 015072 012700 015202          MOV      #GGBBF1,R0          ;SET UP THE OPERAND ADDRESS.
3858 015076 012737 015112 001236          MOV      #GGB2,@#TMP2
3859 015104 012737 015212 000004          MOV      #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
3860

```

```

3861 015112 170640 GGB2: ABSD -(R0) ;TEST INSTRUCTION.
3862
3863 015114 170205 STFPS R5 ;GET FPS.
3864 015116 012701 015172 MOV #GGBBF0,R1 ;CHECK RESULT.
3865 015122 012702 000004 MOV #4,R2
3866 015126 005721 1$: TST (R1)+
3867 015130 001046 BNE GGB15 ;BRANCH IF INCORRECT.
3868 015132 077203 SOB R2,1$
3869
3870 015134 020027 015172 CMP R0,#GGBBF0 ;IS R0 CORRECT?
3871 015140 001055 BNE GGB20 ;BRANCH IF INCORRECT.
3872 015142 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
3873 015146 001061 BNE GGB25 ;BRANCH IF INCORRECT.
3874 015150 000466 BR GGBDONE
3875
3876 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
3877 015152 000177 GGBTP1: 177
3878 015154 117273 117273
3879 015156 147576 147576
3880 015160 177071 177071
3881 015162 000000 GGBTP2: 0
3882 015164 000000 0
3883 015166 000000 0
3884 015170 000000 0
3885 015172 177777 GGBBF0: -1
3886 015174 177777 -1
3887 015176 177777 -1
3888 015200 177777 -1
3889 015202 177777 GGBBF1: -1
3890 015204 177777 -1
3891 015206 177777 -1
3892 015210 177777 -1
3893
3894 ;IF A TRAP TO 4 OCCURS COME HERE.
3895 015212 011602 GGB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
3896 015214 020227 015114 CMP R2,#GGB2+2
3897 015220 001405 BEQ 1$ ;BRANCH IF YES.
3898 015222 020227 015116 CMP R2,#GGB2+4
3899 015226 001402 BEQ 1$ ;BRANCH IF YES.
3900 015230 000137 042620 JMP 0#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,
3901 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
3902 015234 022626 1$: CMP (SP)+,(SP)+
3903 015236 010237 001236 MOV R2,@$TMP2
3904 015242 104102 2$: ERROR 102 ;ODD ADRES
3905 015244 000430 BR GGBDONE ;BUT FDSTX IN ST 771
3906
3907 ;REPORT RESULT INCORRECT:
3908 015246 012737 015162 001240 GGB15: MOV #GGBTP2,@$TMP3
3909 015254 012737 015152 001242 MOV #GGBTP1,@$TMP4
3910 015262 012737 015172 001244 MOV #GGBBF0,@$TMP5
3911 015270 104103 1$: ERROR 103 ;BAD DATA X11*0 ST 312X
3912 015272 000415 BR GGBDONE
3913
3914 ;REPORT R0 INCORRECT:
3915 015274 012737 015172 001240 GGB20: MOV #GGBBF01,@$TMP3
3916 015302 010037 001242 MOV R0,@$TMP4

```

```

3917 015306 104104          1$:  ERROR  104          ;R0 BADX
3918 015310 000406          BR      GGBDONE
3919
3920          ;REPORT FPS INCORRECT:
3921 015312 010537 001240    GGB25:  MOV      R5,0#$TMP3
3922 015316 012737 000204 001244  MOV      #204,0#$TMP5
3923 015324 104105          1$:  ERROR  105          ;FPS X
3924
3925 015326          GGBDONE:
3926 015326 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3927          ;SEE IF THE USER HAS EXPRESSED
3928          ;THE DESIRE TO CHANGE THE SOFTWARE
3929          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3930          ;THE USER TYPED CONTROL G?).
3931          ;*****
3932          ;*TEST 24      NEGf, ABSF AND TSTF SOURCE MODE 3 TEST
3933          ;*
3934          ;*THIS IS A TEST THE NEGf, ABSF AND TSTF
3935          ;*SOURCE FLOWS. THE ABSD INSTRUCTION
3936          ;*IS USED TO TEST MODE 3
3937          ;*
3938          ;*****
3939 015330 000004          TST24:  SCOPE
3940
3941 015332          HHB1:
3942 015332 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
3943 015334 012700 015442    MOV      #HHBTP1,R0          ;SET UP THE DATA BUFFER,
3944 015340 012701 015472    MOV      #HHBBF0,R1
3945 015344 012702 000010    MOV      #10,R2
3946 015350 012021          1$:  MOV      (R0)+,(R1)+
3947 015352 077202          SOB      R2,1$
3948 015354 012700 000200    MOV      #200,R0          ;SET FD,
3949 015360 170100          LDFPS      R0
3950 015362 012700 015502    MOV      #HHBBF1,R0          ;SET UP THE OPERAND ADDRESS,
3951 015366 012737 015402 001236  MOV      #HHB2,0#$TMP2
3952 015374 012737 015512 000004  MOV      #HHB10,0$ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR,
3953
3954 015402 170630          HHB2:  ABSD      @(R0)+          ;TEST INSTRUCTION,
3955
3956 015404 170205          STFPS      R5          ;GET FPS,
3957 015406 012701 015472    MOV      #HHBBF0,R1          ;CHECK RESULT,
3958 015412 012702 000004    MOV      #4,R2
3959 015416 005721          1$:  TST      (R1)+
3960 015420 001052          BNE      HHB15          ;BRANCH IF INCORRECT,
3961 015422 077203          SOB      R2,1$
3962 015424 020027 015504    CMP      R0,#HHBBF1+2          ;IS R0 CORRECT?
3963 015430 001061          BNE      HHB20          ;BRANCH IF INCORRECT,
3964 015432 022705 000204    CMP      #204,R5          ;IS THE FPS CORRECT?
3965 015436 001065          BNE      HHB25          ;BRANCH IF INCORRECT,
3966 015440 000472          BR      HHBDONE
3967
3968          ;THESE ARE TEST DATA TABLES AND DATA BUFFER,
3969 015442 000177          HHBTP1: 177
3970 015444 147576          147576
3971 015446 177071          177071
3972 015450 107576 015472 177777 107576,HHBBF0,-1,-1,-1

```



```

3973 015456 177777 177777
3974 015462 000000 000000 000000 HHBTP2: 0,0,0,0
3975 015470 000000
3976 015472 177777 HHBBF0: -1
3977 015474 177777 -1
3978 015476 177777 -1
3979 015500 177777 -1
3980 015502 177777 HHBBF1: -1
3981 015504 177777 -1
3982 015506 177777 -1
3983 015510 177777 -1
3984
3985
3986 015512 011602 ;IF A TRAP TO 4 OCCURS COME HERE.
3987 015514 020227 015404 HHB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
3988 015520 001405 CMP R2,#HNB2+2
3989 015522 020227 015406 BEQ 1$ ;BRANCH IF YES,
3990 015526 001402 CMP R2,#HNB2+4
3991 015530 000137 042620 BEQ 1$ ;BRANCH IF YES,
3992 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4,
3993 015534 022626 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,
3994 015536 010237 001236 1$: CMP (SP)+,(SP)+
3995 015542 104106 2$: ERROR 106 ;ODD ADRES
3996 015544 000430 BR HHBDONE ;BUT FDSTX IN ST 771
3997
3998 ;REPORT RESULT INCORRECT;
3999 015546 012737 015462 001240 HHB15: MOV #HHBTP2,@#STMP3
4000 015554 012737 015442 001242 MOV #HHBTP1,@#STMP4
4001 015562 012737 015472 001244 MOV #HHBBF0,@#STMP5
4002 015570 104110 1$: ERROR 110 ;BAD DATA X11*0 ST 3127
4003 015572 000415 BR HHBDONE
4004
4005 ;REPORT R0 INCORRECT;
4006 015574 012737 015504 001240 HNB20: MOV #HHBBF1+2,@#STMP3
4007 015602 010037 001242 MOV R0,@#STMP4
4008 015606 104111 1$: ERROR 111 ;R0 INCORRECT,
4009 015610 000406 BR HHBDONE
4010 ;REPORT FPS INCORRECT;
4011 015612 010537 001240 HNB25: MOV R5,@#STMP3
4012 015616 012737 000204 001244 MOV #204,@#STMP5
4013 015624 104112 1$: ERROR 112 ;FPSX
4014
4015 HHBDONE:
4016 015626 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4017 ;SEE IF THE USER HAS EXPRESSED
4018 ;THE DESIRE TO CHANGE THE SOFTWARE
4019 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4020 ;THE USER TYPED CONTROL G?),
4021 ;*****
4022 ;*TEST 25 NEGf, ABSF AND TSTF SOURCE MODE 5 TEST
4023 ;*
4024 ;*THIS IS A TEST THE NEGf, ABSF AND TSTF
4025 ;*SOURCE FLOWS. THE NEGf INSTRUCTION
4026 ;*IS USED TO TEST MODE 5
4027 ;*
4028 ;*****

```

```

4029 015630 000004          TST25: SCOPE
4030
4031 015632          IIB1:
4032 015632 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
4033 015634 012700 015742          MOV          #IIBTP1,R0          ;SET UP THE DATA BUFFER,
4034 015640 012701 015772          MOV          #IIBBF0,R1
4035 015644 012702 000010          MOV          #10,R2
4036 015650 012021          1$: MOV          (R0)+,(R1)+
4037 015652 077202          SOB          R2,1$
4038 015654 012700 000200          MOV          #200,R0          ;SET FD.
4039 015660 170100          LDFPS        R0
4040 015662 012700 016004          MOV          #IIBBF1+2,R0          ;SET UP THE OPERAND ADDRESS,
4041 015666 012737 015702 001236          MOV          #IIB2,@$TMP2
4042 015674 012737 016012 000004          MOV          #IIB10,@$ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR,
4043
4044 015702 170750          IIB2: NEG D    0-(R0)          ;TEST INSTRUCTION.
4045
4046 015704 170205          STFPS        R5          ;GET FPS.
4047 015706 012701 015772          MOV          #IIBBF0,R1          ;CHECK RESULT,
4048 015712 012702 000004          MOV          #4,R2
4049 015716 005721          1$: TST          (R1)+
4050 015720 001052          BNE          IIB15          ;BRANCH IF INCORRECT.
4051 015722 077203          SOB          R2,1$
4052 015724 020027 016002          CMP          R0,#IIBBF1          ;IS R0 CORRECT?
4053 015730 001061          BNE          IIB20          ;BRANCH IF INCORRECT.
4054 015732 022705 000204          CMP          #204,R5          ;IS THE FPS CORRECT?
4055 015736 001065          BNE          IIB25          ;BRANCH IF INCORRECT.
4056 015740 000472          BR          IIBDONE
4057
4058          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4059 015742 000176          IIBTP1: 176
4060 015744 177074          177074
4061 015746 127374          127374
4062 015750 157677 015772 177777          157677,IIBBF0,-1,-1,-1
4063 015756 177777 177777
4064 015762 000000          IIBTP2: 0
4065 015764 000000          0
4066 015766 000000          0
4067 015770 000000          0
4068 015772 177777          IIBBF0: -1
4069 015774 177777          -1
4070 015776 177777          -1
4071 016000 177777          -1
4072 016002 177777          IIBBF1: -1
4073 016004 177777          -1
4074 016006 177777          -1
4075 016010 177777          -1
4076
4077          ;IF A TRAP TO 4 OCCURS COME HERE.
4078 016012 011602          IIB10: MOV          (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4079 016014 020227 015704          CMP          R2,#IIB2+2
4080 016020 001405          BEQ          1$          ;BRANCH IF YES.
4081 016022 020227 015706          CMP          R2,#IIB2+4
4082 016026 001402          BEQ          1$          ;BRANCH IF YES.
4083 016030 000137 042620          JMP          @$CPSPUR          ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4084          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.

```

```

4085 016034 022626          1$:  CMP      (SP)+,(SP)+
4086 016036 010237 001236    MOV      R2,@$TMP2
4087 016042 104113          2$:  ERROR    113          ;ODD ADRES
4088 016044 000430          BR       IIBDONE        ;BUT FDSTX IN ST 771
4089
4090          ;REPORT RESULT INCORRECT:
4091 016046 012737 015762 001240 IIB15: MOV      #IIBTP2,@$TMP3
4092 016054 012737 015742 001242    MOV      #IIBTP1,@$TMP4
4093 016062 012737 015772 001244    MOV      #IIBBF0,@$TMP5
4094 016070 104114          1$:  ERROR    114          ;BAD DATA X11#0 ST 3127
4095 016072 000415          BR       IIBDONE
4096
4097          ;REPORT R0 INCORRECT:
4098 016074 012737 016002 001240 IIB20: MOV      #IIBBF1,@$TMP3
4099 016102 010037 001242    MOV      R0,@$TMP4
4100 016106 104115          1$:  ERROR    115          ;R0 BADX
4101 016110 000406          BR       IIBDONE
4102          ;REPORT FPS INCORRECT:
4103 016112 010537 001240 IIB25: MOV      R5,@$TMP3
4104 016116 012737 000204 001244    MOV      #204,@$TMP5
4105 016124 104116          1$:  ERROR    116          ;FPSX
4106
4107          IIBDONE:
4108 016126 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
4109          ;SEE IF THE USER HAS EXPRESSED
4110          ;THE DESIRE TO CHANGE THE SOFTWARE
4111          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4112          ;THE USER TYPED CONTROL G?).
4113
4114          ;*****
4115          ;*TEST 26      NEGf, ABSF AND TSTF SOURCE MODE 6 TEST
4116          ;*
4117          ;*THIS IS A TEST THE NEGf, ABSF AND TSTF
4118          ;*SOURCE FLOWS, THE ABSD INSTRUCTION
4119          ;*IS USED TO TEST MODE 6
4120          ;*
4121          ;*****
4122 016130 000004          TST26: SCOPE
4123
4124          JJB1:
4125 016132 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4126 016134 012700 016244    MOV      #JJBTP1,R0          ;SET UP THE DATA BUFFER,
4127 016140 012701 016266    MOV      #JJBFB0,R1
4128 016144 012702 000004    MOV      #4,R2
4129 016150 012021          1$:  MOV      (R0)+,(R1)+
4130 016152 077202          SOB      R2,1$
4131 016154 012700 000200    MOV      #200,R0          ;SET FD,
4132 016160 170100          LDFPS      R0
4133 016162 012700 016257    MOV      #JJBFB0-7,R0        ;SET UP THE OPERAND ADDRESS.
4134 016166 012737 016202 001236    MOV      #JJB2,@$TMP2
4135 016174 012737 016306 000004    MOV      #JJB10,@$ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR,
4136
4137 016202 170660 000007          JJB2:  ABSD      7(R0)          ;TEST INSTRUCTION.
4138
4139 016206 170205          STFPS      R5          ;GET FPS,
4140 016210 012701 016266    MOV      #JJBFB0,R1          ;CHECK RESULT,

```

```

4141 016214 012702 000004          MOV    #4,R2
4142 016220 005721          1$:   TST    (R1)+
4143 016222 001047          BNE    JJB15          ;BRANCH IF INCORRECT.
4144 016224 077203          SOB    R2,1$
4145 016226 020027 016257          CMP    R0,#JJBFF0-7  ;IS R0 CORRECT?
4146 016232 001043          BNE    JJB15          ;BRANCH IF INCORRECT.
4147 016234 022705 000204          CMP    #204,R5      ;IS THE FPS CORRECT?
4148 016240 001053          BNE    JJB20          ;BRANCH IF INCORRECT.
4149 016242 000467          BR     JJBDONE
4150
4151          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4152 016244 000177          JJBTP1: 177
4153 016246 161524          161524
4154 016250 131273          131273
4155 016252 107174 000000          107174,
4156 016256 000000          JJBTP2: 0
4157 016260 000000          0
4158 016262 000000          0
4159 016264 000000          0
4160 016266 177777          JJBBF0: -1
4161 016270 177777          -1
4162 016272 177777          -1
4163 016274 177777          -1
4164 016276 177777          JJBBF1: -1
4165 016300 177777          -1
4166 016302 177777          -1
4167 016304 177777          -1
4168
4169          ;IF A TRAP TO 4 OCCURS COME HERE.
4170 016306 011602          JJB10: MOV    (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
4171 016310 020227 016204          CMP    R2,#JJB2+2
4172 016314 001405          BEQ    1$          ;BRANCH IF YES.
4173 016316 020227 016206          CMP    R2,#JJB2+4
4174 016322 001402          BEQ    1$          ;BRANCH IF YES.
4175 016324 000137 042620          JMP    @CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,
4176          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4,
4177 016330 022626          1$:   CMP    (SP)+,(SP)+
4178 016332 010237 001236          MOV    R2,@$TMP2
4179 016336 104117          2$:   ERROR  117      ;ODD ADRES
4180 016340 000430          BR     JJBDONE      ;BUT FDSTX IN ST 771
4181
4182          ;REPORT RESULT INCORRECT:
4183 016342 012737 016256 001240          JJB15: MOV    #JJBTP2,@$TMP3
4184 016350 012737 016244 001242          MOV    #JJBTP1,@$TMP4
4185 016356 012737 016266 001244          MOV    #JJBBF0,@$TMP5
4186 016364 104120          1$:   ERROR  120      ;BAD DATA X11*0 ST 3127
4187 016366 000415          BR     JJBDONE
4188
4189          ;REPORT R0 INCORRECT:
4190 016370 012737 016257 001240          JJB20: MOV    #JJBBF0-7,@$TMP3
4191 016376 010037 001242          MOV    R0,@$TMP4
4192 016402 104124          1$:   ERROR  124      ;R0 BADX
4193 016404 000406          BR     JJBDONE
4194          ;REPORT FPS INCORRECT:
4195 016406 010537 001240          JJB25: MOV    R5,@$TMP3
4196 016412 012737 000204 001244          MOV    #204,@$TMP5

```

4197 016420 104122  
 4198 016422  
 4199 016422 104412  
 4200  
 4201  
 4202  
 4203  
 4204  
 4205  
 4206  
 4207  
 4208  
 4209  
 4210  
 4211  
 4212 016424 000004  
 4213  
 4214 016426  
 4215 016426 104413  
 4216 016430 012700 016540  
 4217 016434 012701 016570  
 4218 016440 012702 000010  
 4219 016444 012021  
 4220 016446 077202  
 4221 016450 012700 000200  
 4222 016454 170100  
 4223 016456 012700 016571  
 4224 016462 012737 016476 001236  
 4225 016470 012737 016610 000004  
 4226  
 4227 016476 170770 000007  
 4228  
 4229 016502 170205  
 4230 016504 012701 016570  
 4231 016510 012702 000004  
 4232 016514 005721  
 4233 016516 001052  
 4234 016520 077203  
 4235 016522 020027 016571  
 4236 016526 001061  
 4237 016530 022705 000204  
 4238 016534 001056  
 4239 016536 000472  
 4240  
 4241  
 4242 016540 000177  
 4243 016542 167574  
 4244 016544 137271  
 4245 016546 107675 016570 177777  
 4246 016554 177777 177777  
 4247 016560 000000  
 4248 016562 000000  
 4249 016564 000000  
 4250 016566 000000  
 4251 016570 177777  
 4252 016572 177777

```

1$: ERROR 122 ;FPSX
JJB 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?).
;*****
;*TEST 27 NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
;#
;*THIS IS A TEST THE NEGF, ABSF AND TSTF
;*SOURCE FLOWS. THE ABSD INSTRUCTION
;*IS USED TO TEST MODE 6
;#
;*****
TST27: SCOPE

KKB1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #KKBTP1,R0 ;SET UP THE DATA BUFFER.
MOV #KKB BF0,R1
MOV #10,R2
1$: MOV (R0)+,(R1)+
SOB R2,1$
MOV #200,R0 ;SET FD.
LDFPS R0
MOV #KKB BF1-7,R0 ;SET UP THE OPERAND ADDRESS.
MOV #KKB2,@#STMP2
MOV #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

KKB2: NEG D @7(R0) ;TEST INSTRUCTION.

STFPS R5 ;GET FPS.
MOV #KKB BF0,R1 ;CHECK RESULT.
MOV #4,R2
1$: TST (R1)+
BNE KKB15 ;BRANCH IF INCORRECT.
SOB R2,1$
CMP R0,#KKB BF1-7 ;IS R0 CORRECT?
BNE KKB20 ;BRANCH IF INCORRECT.
CMP #204,R5 ;IS THE FPS CORRECT?
BNE KKB20 ;BRANCH IF INCORRECT.
BR KKB DONE

;THESE ARE TEST DATA TABLES AND DATA BUFFER.
KKBTP1: 177
167574
137271
107675,KKB BF0,-1,-1,-1

KKBTP2: 0
0
0

KKB BF0: -1
-1
  
```

```

4253 016574 177777 -1
4254 016576 177777 -1
4255 016600 177777 KKBBF1: -1
4256 016602 177777 -1
4257 016604 177777 -1
4258 016606 177777 -1
4259
4260 ;IF A TRAP TO 4 OCCURS COME HERE.
4261 016610 011602 KKB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
4262 016612 020227 016500 CMP R2,#KKB2+2
4263 016616 001405 BEQ 1$ ;BRANCH IF YES.
4264 016620 020227 016502 CMP R2,#KKB2+4
4265 016624 001402 BEQ 1$ ;BRANCH IF YES.
4266 016626 000137 042620 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4,
4267 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
4268 016632 022626 1$: CMP (SP)+,(SP)+
4269 016634 010237 001236 MOV R2,@#STMP2
4270 016640 104123 2$: ERROR 123 ;ODD ADRES
4271 016642 000430 BR KKBDONE ;BUT FDSTX IN ST 771
4272
4273 ;REPORT RESULT INCORRECT:
4274 016644 012737 016560 001240 KKB15: MOV #KKBTP2,@#STMP3
4275 016652 012737 016540 001242 MOV #KKBTP1,@#STMP4
4276 016660 012737 016570 001244 MOV #KKBBF0,@#STMP5
4277 016666 104124 1$: ERROR 124 ;BAD DATA X11*0 ST 3127
4278 016670 000415 BR KKBDONE
4279
4280 ;REPORT R0 INCORRECT:
4281 016672 012737 016571 001240 KKB20: MOV #KKBBF1-7,@#STMP3
4282 016700 010037 001242 MOV R0,@#STMP4
4283 016704 104125 1$: ERROR 125 ;R0 BADX
4284 016706 000406 BR KKBDONE
4285 ;REPORT FPS INCORRECT:
4286 016710 010537 001240 KKB25: MOV R5,@#STMP3
4287 016714 012737 000204 001244 MOV #204,@#STMP5
4288 016722 104126 1$: ERROR 126 ;FPSX
4289
4290 KKBDONE:
4291 016724 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4292 ;SEE IF THE USER HAS EXPRESSED
4293 ;THE DESIRE TO CHANGE THE SOFTWARE
4294 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4295 ;THE USER TYPED CONTROL G?),
4296 ;*****
4297 ;*TEST 30 NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
4298 ;*
4299 ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
4300 ;*SOURCE FLOWS. THE NEGD INSTRUCTION
4301 ;*IS USED TO TEST MODE 6
4302 ;*
4303 ;*****
4304 016726 000004 TST30: SCOPE
4305 016730 LLB1:
4306 016730 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
4307 016732 012700 017030 MOV #LLBTP1,R0 ;SET UP THE DATA BUFFER.
4308 016736 012701 017050 MOV #LLBBF0,R1

```

```

4309 016742 012702 000004      MOV      #4,R2
4310 016746 012021      1$:     MOV      (R0)+,(R1)+
4311 016750 077202      SOB      R2,1$
4312 016752 012700 000200      MOV      #200,R0      ;SET FD.
4313 016756 170100      LDFPS   R0
4314 016760 012737 016774 001236      MOV      #LLB2,@$TMP2
4315 016766 012737 017070 000004      MOV      #LLB10,@$ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
4316
4317 016774 170767 000050      LLB2:   NEG     LLBBF0      ;TEST INSTRUCTION.
4318
4319 017000 170205      STFPS   R5      ;GET FPS.
4320 017002 012701 017050      MOV      #LLBBF0,R1      ;CHECK RESULT.
4321 017006 012702 000004      MOV      #4,R2
4322 017012 005721      1$:     TST      (R1)+
4323 017014 001043      BNE     LLB15      ;BRANCH IF INCORRECT.
4324 017016 077203      SOB      R2,1$
4325 017020 022705 000204      CMP      #204,R5      ;IS THE FPS CORRECT?
4326 017024 001052      BNE     LLB25      ;BRANCH IF INCORRECT.
4327 017026 000457      BR      LLBDONE
4328
4329      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
4330 017030 000127      LLBTP1: 127
4331 017032 137475      137475
4332 017034 147372      147372
4333 017036 117057      117057
4334 017040 000000      LLBTP2: 0
4335 017042 000000      0
4336 017044 000000      0
4337 017046 000000      0
4338 017050 177777      LLBBF0: -1
4339 017052 177777      -1
4340 017054 177777      -1
4341 017056 177777      -1
4342 017060 177777      LLBBF1: -1
4343 017062 177777      -1
4344 017064 177777      -1
4345 017066 177777      -1
4346
4347      ;IF A TRAP TO 4 OCCURS COME HERE.
4348 017070 011602      LLB10:  MOV     (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,
4349 017072 020227 016776      CMP      R2,#LLB2+2
4350 017076 001405      BEQ      1$      ;BRANCH IF YES.
4351 017100 020227 017000      CMP      R2,#LLB2+4
4352 017104 001402      BEQ      1$      ;BRANCH IF YES.
4353 017106 000137 042620      JMP      @#CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
4354      ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
4355 017112 022626      1$:     CMP      (SP)+,(SP)+
4356 017114 010237 001236      MOV      R2,@$TMP2
4357 017120 104127      2$:     ERROR   127      ;ODD ADRES
4358 017122 000421      BR      LLBDONE      ;BUT FDSTX IN ST 771
4359
4360      ;REPORT RESULT INCORRECT:
4361 017124 012737 017040 001240      LLB15:  MOV      #LLBTP2,@$TMP3
4362 017132 012737 017030 001242      MOV      #LLBTP1,@$TMP4
4363 017140 012737 017050 001244      MOV      #LLBBF0,@$TMP5
4364 017146 104130      1$:     ERROR   130      ;BAD DATA X11*0 ST 3127

```

4365 017150 000406

BR LLBDONE

4366 4367 017152 010537 001240

;REPORT FPS INCORRECT:

4368 017156 012737 000204 001244

LLB25: MOV R5,0#STMP3

4369 017164 104131

MOV #204,0#STMP5

4370

1\$: ERROR 131 ;FPSX

4371 017166

LLBDONE:

4372 017166 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?).

4373

4374

4375

4376

4377

4378

;\*\*\*\*\*

4379

\*TEST 31 NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST

4380

;

4381

\*THIS IS A TEST THE NEGF, ABSF AND TSTF

4382

\*SOURCE FLOWS. THE ABSD INSTRUCTION

4383

\*IS USED TO TEST MODE 7

4384

;

4385 017170 000004

;\*\*\*\*\*

4386

TST31: SCOPE

4387 017172

MMB1:

4388 017172 104413

LPERR

;SET UP THE LOOP ON ERROR ADDRESS.

4389 017174 012700 017272

MOV #MMBTP1,R0

;SET UP THE DATA BUFFER,

4390 017200 012701 017322

MOV #MMBBF0,R1

4391 017204 012702 000010

MOV #10,R2

4392 017210 012021

1\$: MOV (R0)+,(R1)+

4393 017212 077202

SOB R2,1\$

4394 017214 012700 000200

MOV #200,R0

;SET FD.

4395 017220 170100

LDFPS R0

4396 017222 012737 017236 001236

MOV #MMB2,0#STMP2

4397 017230 012737 017342 000004

MOV #MMB10,0#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.

4398

4399 017236 170677 000070

MMB2: ABSD @MMBBF1 ;TEST INSTRUCTION.

4400

4401 017242 170205

STFPS R5

;GET FPS.

4402 017244 012701 017322

MOV #MMBBF0,R1

;CHECK RESULT.

4403 017250 012702 000004

MOV #4,R2

4404 017254 005721

1\$: TST (R1)+

4405 017256 001047

BNE MMB15 ;BRANCH IF INCORRECT.

4406 017260 077203

SOB R2,1\$

4407 017262 022705 000204

CMP #204,R5 ;IS THE FPS CORRECT?

4408 017266 001056

BNE MMB25 ;BRANCH IF INCORRECT.

4409 017270 000463

BR MMBDONE

4410

;THESE ARE TEST DATA TABLES AND DATA BUFFER.

4411

MMBTP1: 137

4412 017272 000137

045607

4413 017274 045607

101230

4414 017276 101230

45607,MMBBF0,-1,-1,-1

4415 017300 045607 017322 177777

4416 017306 177777 177777

4417 017312 000000

MMBTP2: 0

4418 017314 000000

0

4419 017316 000000

0

4420 017320 000000

0



4421 017322 177777  
 4422 017324 177777  
 4423 017326 177777  
 4424 017330 177777  
 4425 017332 177777  
 4426 017334 177777  
 4427 017336 177777  
 4428 017340 177777  
 4429  
 4430

MMBBF0: -1  
 -1  
 -1  
 -1  
 MMBBF1: -1  
 -1  
 -1  
 -1

4431 017342 011602  
 4432 017344 020227 017240  
 4433 017350 001405  
 4434 017352 020227 017242  
 4435 017356 001402  
 4436 017360 000137 042620  
 4437  
 4438 017364 022626  
 4439 017366 010237 001236  
 4440 017372 104132  
 4441 017374 000421  
 4442

;IF A TRAP TO 4 OCCURS COME HERE.  
 MMB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION,  
 CMP R2,#MMB2+2  
 BEQ 1\$ ;BRANCH IF YES.  
 CMP R2,#MMB2+4  
 BEQ 1\$ ;BRANCH IF YES.  
 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.  
 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.  
 1\$: CMP (SP)+,(SP)+  
 MOV R2,@#STMP2  
 2\$: ERROR 132 ;ODD ADRES  
 BR MMBDONE ;BUT FDSTX IN ST 771

4443  
 4444 017376 012737 017312 001240  
 4445 017404 012737 017272 001242  
 4446 017412 012737 017322 001244  
 4447 017420 104133  
 4448 017422 000406  
 4449  
 4450 017424 010537 001240  
 4451 017430 012737 000204 001244  
 4452 017436 104134  
 4453  
 4454 017440  
 4455 017440 104412  
 4456  
 4457  
 4458  
 4459  
 4460

;REPORT RESULT INCORRECT:  
 MMB15: MOV #MMBTP2,@#STMP3  
 MOV #MMBTP1,@#STMP4  
 MOV #MMBBF0,@#STMP5  
 1\$: ERROR 133 ;BAD DATA X11\*0 ST 3127  
 BR MMBDONE  
 ;REPORT FPS INCORRECT:  
 MMB25: MOV R5,@#STMP3  
 MOV #204,@#STMP5  
 1\$: ERROR 134 ;FPSX

4454 017440  
 4455 017440 104412  
 4456

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

4457  
 4458  
 4459  
 4460  
 4461  
 4462  
 4463  
 4464  
 4465  
 4466  
 4467 017442 000004  
 4468  
 4469 017444  
 4470 017444 104413  
 4471 017446 012700 000200  
 4472 017452 170100  
 4473 017454 012700 017542  
 4474 017460 172410  
 4475 017462 012737 017470 001236  
 4476

;\*\*\*\*\*  
 ;\*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  
 NNB1:  
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 MOV #200,R0 ;SET FD.  
 LDFPS R0  
 MOV #NNBTP1,R0 ;SET UP AC0.  
 LDD (R0),AC0  
 MOV #NNB2,@#STMP2

```

4477 017470 170700 NNB2: NEG D AC0 ;TEST INSTRUCTION.
4478
4479 017472 170205 STFPS R5 ;GET FPS.
4480 017474 012700 000200 MOV #200,R0 ;SET FD.
4481 017500 170100 LDFPS R0
4482 017502 012700 017562 MOV #NNBBF0,R0 ;GET THE RESULT.
4483 017506 174010 STD AC0,(R0)
4484 017510 012700 017562 MOV #NNBBF0,R0 ;IS THE RESULT CORRECT?
4485 017514 012701 017552 MOV #NNBTP2,R1
4486 017520 012702 000004 MOV #4,R2
4487 017524 022021 1s: CMP (R0)+,(R1)+
4488 017526 001021 BNE NNB10 ;BRANCH IF INCORRECT.
4489 017530 077203 SOB R2,1s
4490 017532 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4491 017536 001033 BNE NNB15 ;BRANCH IF INCORRECT.
4492 017540 000440 BR NNB DONE
4493
4494 ;THESE ARE DATA TABLES AND A DATA BUFFER.
4495 017542 013572 NNBTP1: 013572
4496 017544 046013 46013
4497 017546 057246 57246
4498 017550 013570 013570
4499 017552 113572 NNBTP2: 113572
4500 017554 046013 46013
4501 017556 057246 57246
4502 017560 013570 013570
4503 017562 000000 NNB BF0: 0
4504 017564 000000 0
4505 017566 000000 0
4506 017570 000000 0
4507
4508 ;REPORT RESULT INCORRECT:
4509 017572 012737 017562 001240 NNB10: MOV #NNBBF0,@$TMP3
4510 017600 012737 017552 001242 MOV #NNBTP2,@$TMP4
4511 017606 023737 017542 017562 CMP @NNBTP1,@NNBBF0
4512 017614 001002 BNE NNB11
4513 017616 104135 1s: ERROR 135 ;E10*200X ST 336
4514 017620 000410 BR NNB DONE
4515
4516 ;REPORT RESULT INCORRECT:
4517 017622 NNB11:
4518 017622 104136 1s: ERROR 136 ;BAD DATA NEGF
4519 017624 000406 BR NNB DONE
4520
4521 ;REPORT FPS INCORRECT:
4522 017626 010537 001242 NNB15: MOV R5,@$TMP4
4523 017632 012737 000210 001240 MOV #210,@$TMP3
4524 017640 104137 1s: ERROR 137 ;FPSX
4525
4526 NNB DONE:
4527 017642 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4528 ;SEE IF THE USER HAS EXPRESSED
4529 ;THE DESIRE TO CHANGE THE SOFTWARE
4530 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4531 ;THE USER TYPED CONTROL G?),
4532 ;*****

```

```

4533 ;*TEST 33 SPECIAL DEST, MODE 1, TEST
4534 ;*
4535 ;*THIS IS A TEST OF THE NEGJ ABSF AND TSTF DESTINATION FLOWS
4536 ;*MODE 1 USING THE NEGJ INSTR.
4537 ;*
4538 ;/*****
4539 017644 000004 TST33: SCOPE
4540
4541 017646 OOB1:
4542 017646 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
4543 017650 012701 017760 MOV #OOBTP1,R1 ;SET UP THE DATA BUFFER,
4544 017654 012700 017770 MOV #OOBTP2,R0
4545 017660 012702 000004 MOV #4,R2
4546 017664 012021 1$: MOV (R0)+,(R1)+
4547 017666 077202 SOB R2,1$
4548 017670 012700 017760 MOV #OOBTP1,R0
4549 017674 042710 100000 BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
4550 017700 012737 017714 001236 MOV #OOB2,@$STMP2
4551 017706 012701 000200 MOV #200,R1 ;SET FD.
4552 017712 170101 LDFPS R1
4553
4554 017714 170710 OOB2: NEGJ (R0) ;TEST INSTRUCTION,
4555 017716 170205 STFPS R5 ;GET FPS,
4556 017720 012701 017760 MOV #OOBTP1,R1 ;IS THE RESULT CORRECT,
4557 017724 012702 017770 MOV #OOBTP2,R2
4558 017730 012703 000004 MOV #4,R3
4559 017734 022122 1$: CMP (R1)+,(R2)+
4560 017736 001020 BNE OOB10 ;BRANCH IF INCORRECT,
4561 017740 077303 SOB R3,1$
4562 017742 022700 017760 CMP #OOBTP1,R0 ;IS R0 CORRECT,
4563 017746 001024 BNE OOB15 ;BRANCH IF INCORRECT,
4564 017750 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4565 017754 001030 BNE OOB20 ;BRANCH IF INCORRECT,
4566 017756 000435 BR OOBDONE
4567
4568 ;THESE ARE DATA TABLES AND A DATA BUFFER,
4569 017760 023245 OOBTP1: 023245
4570 017762 026720 26720
4571 017764 122324 122324
4572 017766 052672 52672
4573 017770 123245 OOBTP2: 123245
4574 017772 026720 26720
4575 017774 122324 122324
4576 017776 052672 52672
4577
4578 ;REPORT RESULT INCORRECT:
4579 020000 012737 017760 001240 OOB10: MOV #OOBTP1,@$STMP3
4580 020006 012737 017770 001242 MOV #OOBTP2,@$STMP4
4581 020014 104140 1$: ERROR 140 ;BAD DATA
4582 020016 000415 BR OOBDONE
4583
4584 ;REPORT R0 INCORRECT:
4585 020020 012737 017760 001240 OOB15: MOV #OOBTP1,@$STMP3
4586 020026 010037 001242 MOV R0,@$STMP4
4587 020032 104141 1$: ERROR 141 ;SPEC DESTX
4588 020034 000406 BR OOBDONE ;R0X

```

```

4589
4590 ;REPORT FPS INCORRECT:
4591 020036 012737 000210 001240 OOB20: MOV #210,@$TMP3
4592 020044 010537 001242 MOV R5,@$TMP4
4593 020050 104142 1$: ERROR 142
4594
4595 020052 OOB DONE:
4596 020052 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4597 ;SEE IF THE USER HAS EXPRESSED
4598 ;THE DESIRE TO CHANGE THE SOFTWARE
4599 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4600 ;THE USER TYPED CONTROL G?).
4601 ;*****
4602 ;*TEST 34 SPECIAL DEST, MODE 2, TEST
4603 ;*
4604 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4605 ;*MODE 2 USING THE NEGD INSTR.
4606 ;*
4607 ;*****
4608 020054 000004 TST34: SCOPE
4609 020056 PPB1:
4610 020056 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4611
4612 020060 012701 020170 MOV #PPBTP1,R1 ;SET UP THE DATA BUFFER.
4613 020064 012700 020200 MOV #PPBTP2,R0
4614 020070 012702 000004 MOV #4,R2
4615 020074 012021 1$: MOV (R0)+,(R1)+
4616 020076 077202 SOB R2,1$
4617 020100 012700 020170 MOV #PPBTP1,R0
4618 020104 042710 100000 BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
4619 020110 012737 020124 001236 MOV #PPB2,@$TMP2
4620 020116 012701 000200 MOV #200,R1 ;SET FD.
4621 020122 170101 LDFPS R1
4622
4623 020124 170720 PPB2: NEGD (R0)+ ;TEST INSTRUCTION.
4624
4625 020126 170205 STFPS R5 ;GET FPS.
4626 020130 012701 020170 MOV #PPBTP1,R1 ;IS THE RESULT CORRECT,
4627 020134 012702 020200 MOV #PPBTP2,R2
4628 020140 012703 000004 MOV #4,R3
4629 020144 022122 1$: CMP (R1)+,(R2)+
4630 020146 001020 BNE PPB10 ;BRANCH IF INCORRECT.
4631 020150 077303 SOB R3,1$
4632 020152 022700 020200 CMP #PPBTP1+10,R0 ;IS R0 CORRECT.
4633 020156 001024 BNE PPB15 ;BRANCH IF INCORRECT.
4634 020160 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4635 020164 001030 BNE PPB20 ;BRANCH IF INCORRECT.
4636 020166 000435 BR PPBDONE
4637
4638 ;THESE ARE DATA TABLES AND A DATA BUFFER.
4639 020170 023245 PPBTP1: 023245
4640 020172 026720 26720
4641 020174 122324 122324
4642 020176 052672 52672
4643 020200 123245 PPBTP2: 123245
4644 020202 026720 26720
  
```

```

4645 020204 122324 122324
4646 020206 052672 52672
4647
4648
4649 020210 012737 020170 001240 ;REPORT RESULT INCORRECT:
PPB10: MOV #PPBTP1,0##$TMP3
4650 020216 012737 020200 001242 MOV #PPBTP2,0##$TMP4
4651 020224 104143 1$: ERROR 143 ;BAD DATA
4652 020226 000415 BR PPBDONE
4653
4654
4655 020230 012737 020200 001240 ;REPORT R0 INCORRECT:
PPB15: MOV #PPBTP1+10,0##$TMP3
4656 020236 010037 001242 MOV R0,0##$TMP4
4657 020242 104144 1$: ERROR 144 ;SPEC DESTX R0X
4658 020244 000406 BR PPBDONE
4659
4660
4661 020246 012737 000210 001240 ;REPORT FPS INCORRECT:
PPB20: MOV #210,0##$TMP3
4662 020254 010537 001242 MOV R5,0##$TMP4
4663 020260 104145 1$: ERROR 145
4664
4665 020262 PPBDONE:
4666 020262 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
4667 ;SEE IF THE USER HAS EXPRESSED
4668 ;THE DESIRE TO CHANGE THE SOFTWARE
4669 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4670 ;THE USER TYPED CONTROL G?).
4671 ;)*****
4672 ;*TEST 35 SPECIAL DEST, MODE 4, TEST
4673 ;*
4674 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4675 ;*MODE 4 USING THE NEGD INSTR.
4676 ;*
4677 ;)*****
4678 020264 000004 TST35: SCOPE
4679 020266 QQB1:
4680 020266 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
4681 020270 012701 020402 MOV #QQBTP1,R1 ;SET UP THE DATA BUFFER.
4682 020274 012700 020422 MOV #QQBTP2,R0
4683 020300 012702 000004 MOV #4,R2
4684 020304 012021 1$: MOV (R0)+,(R1)+
4685 020306 077202 SOB R2,1$
4686 020310 012700 020412 MOV #QQBTP1+10,R0
4687 020314 042760 100000 177770 BIC #100000,-10(R0) ;MAKE OPERAND POSITIVE.
4688 020322 012737 020336 001236 MOV #QQB2,0##$TMP2
4689 020330 012701 000200 MOV #200,R1 ;SET FD.
4690 020334 170101 LDFPS R1
4691
4692 020336 170740 QQB2: NEGD -(R0) ;TEST INSTRUCTION.
4693
4694 020340 170205 STFPS R5 ;GET FPS.
4695 020342 012701 020402 MOV #QQBTP1,R1 ;IS THE RESULT CORRECT.
4696 020346 012702 020422 MOV #QQBTP2,R2
4697 020352 012703 000004 MOV #4,R3
4698 020356 022122 1$: CMP (R1)+,(R2)+
4699 020360 001024 BNE QQB10 ;BRANCH IF INCORRECT.
4700 020362 077303 SOB R3,1$

```

```

4701 020364 022700 020402      CMP      #QQBTP1,R0      ;IS R0 CORRECT,
4702 020370 001030      BNE      QQB15        ;BRANCH IF INCORRECT.
4703 020372 022705 000210      CMP      #210,R5      ;IS THE FPS CORRECT?
4704 020376 001034      BNE      QQB20        ;BRANCH IF INCORRECT.
4705 020400 000441      BR       QQBDONE
4706

```

```

4707 ;THESE ARE DATA TABLES AND A DATA BUFFER,

```

```

4708 020402 023245      QQBTP1: 023245
4709 020404 026720      26720
4710 020406 122324      122324
4711 020410 052672      52672
4712 020412 177777 177777 177777      .WORD   -1,-1,-1,-1
4713 020420 177777

```

```

4714 020422 123245      QQBTP2: 123245
4715 020424 026720      26720
4716 020426 122324      122324
4717 020430 052672      52672
4718

```

```

4719 ;REPORT RESULT INCORRECT:
4720 020432 012737 020402 001240      QQB10:  MOV      #QQBTP1,@#TMP3
4721 020440 012737 020422 001242      MOV      #QQBTP2,@#TMP4
4722 020446 104146      1$:      ERROR    146      ;BAD DATA
4723 020450 000415      BR       QQBDONE
4724

```

```

4725 ;REPORT R0 INCORRECT:
4726 020452 012737 020402 001240      QQB15:  MOV      #QQBTP1,@#TMP3
4727 020460 010037 001242      MOV      R0,@#TMP4
4728 020464 104147      1$:      ERROR    147      ;SPEC DESTX R0X
4729 020466 000406      BR       QQBDONE
4730

```

```

4731 ;REPORT FPS INCORRECT:
4732 020470 012737 000210 001240      QQB20:  MOV      #210,@#TMP3
4733 020476 010537 001242      MOV      R5,@#TMP4
4734 020502 104150      1$:      ERROR    150
4735

```

```

4736 QQBDONE:
4737 020504      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
4738 020504 104412      ;SEE IF THE USER HAS EXPRESSED
4739      ;THE DESIRE TO CHANGE THE SOFTWARE
4740      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4741      ;THE USER TYPED CONTROL G?).
4742

```

```

4743 ;*****
4744 ;*TEST 36      SPECIAL DEST, MODE 3, TEST
4745 ;*
4746 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4747 ;*MODE 3 USING THE NEGD INSTR.
4748 ;*
4749 ;*****

```

```

4750 TST36:  SCOPE
4751 020506 000004
4752

```

```

4753 RRB1:
4754 020510      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS,
4755 020512 012701 020630      MOV      #RRBTP1,R1      ;SET UP THE DATA BUFFER,
4756 020516 012700 020640      MOV      #RRBTP2,R0

```

```

4757 020522 012702 000004          MOV      #4,R2
4758 020526 012021          1$:     MOV      (R0)+,(R1)+
4759 020530 077202          SOB      R2,1$
4760 020532 012700 020650          MOV      #RRBTP3,R0
4761 020536 012710 020630          MOV      #RRBTP1,(R0)
4762 020542 042737 100000 020630          BIC      #100000,@#RRBTP1          ;MAKE THE OPERAND POSITIVE.
4763 020550 012737 020564 001236          MOV      #RRB2,@#TMP2
4764 020556 012701 000200          MOV      #200,R1          ;SET FD.
4765 020562 170101          LDFPS   R1
4766
4767 020564 170730          RRB2:   NEG D   0(R0)+          ;TEST INSTRUCTION.
4768
4769 020566 170205          STFPS   R5          ;GET FPS.
4770 020570 012701 020630          MOV      #RRBTP1,R1          ;IS THE RESULT CORRECT.
4771 020574 012702 020640          MOV      #RRBTP2,R2
4772 020600 012703 000004          MOV      #4,R3
4773 020604 022122          1$:     CMP      (R1)+,(R2)+
4774 020606 001021          BNE     RRB10          ;BRANCH IF INCORRECT.
4775 020610 077303          SOB      R3,1$
4776 020612 022700 020652          CMP      #RRBTP3+2,R0          ;IS R0 CORRECT.
4777 020616 001025          BNE     RRB15          ;BRANCH IF INCORRECT.
4778 020620 022705 000210          CMP      #210,R5          ;IS THE FPS CORRECT?
4779 020624 001031          BNE     RRB20          ;BRANCH IF INCORRECT.
4780 020626 000436          BR      RRB DONE
4781
4782          ;THESE ARE DATA TABLES AND A DATA BUFFER,
4783 020630 023245          RRBTP1: 023245
4784 020632 026720          26720
4785 020634 122324          122324
4786 020636 052672          52672
4787 020640 123245          RRBTP2: 123245
4788 020642 026720          26720
4789 020644 123324          123324
4790 020646 052672          52672
4791 020650 020630          RRBTP3: RRBTP1
4792
4793          ;REPORT RESULT INCORRECT:
4794 020652 012737 020630 001240          RRB10:  MOV      #RRBTP1,@#TMP3
4795 020660 012737 020640 001242          MOV      #RRBTP2,@#TMP4
4796 020666 104150          1$:     ERROR   150          ;BAD DATA
4797 020670 000415          BR      RRB DONE
4798
4799          ;REPORT R0 INCORRECT:
4800 020672 012737 020652 001240          RRB15:  MOV      #RRBTP3+2,@#TMP3
4801 020700 010037 001242          MOV      R0,@#TMP4
4802 020704 104152          1$:     ERROR   152          ;SPEC DESTX R0X
4803 020706 000406          BR      RRB DONE
4804
4805          ;REPORT FPS INCORRECT:
4806 020710 012737 000210 001240          RRB20:  MOV      #210,@#TMP3
4807 020716 010537 001242          MOV      R5,@#TMP4
4808 020722 104153          1$:     ERROR   153
4809
4810          RRB DONE:
4811 020724 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
4812          ;SEE IF THE USER HAS EXPRESSED

```

4813 ;THE DESIRE TO CHANGE THE SOFTWARE  
 4814 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
 4815 ;THE USER TYPED CONTROL G?).  
 4816

4817 ;)\*\*\*\*\*  
 4818 ;\*TEST 37 SPECIAL DEST, MODE 5, TEST  
 4819 ;\*  
 4820 ;\*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS  
 4821 ;\*MODE 5 USING THE NEGD INSTR.  
 4822 ;\*  
 4823 ;)\*\*\*\*\*

4824 020726 000004 TST37: SCOPE  
 4825 020730 SSB1:  
 4826 020730 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 4827 020732 012701 021052 MOV #SSBTP1,R1 ;SET UP THE DATA BUFFER.  
 4828 020736 012700 021062 MOV #SSBTP2,R0  
 4829 020742 012702 000004 MOV #4,R2  
 4830 020746 012021 1\$: MOV (R0)+,(R1)+  
 4831 020750 077202 SOB R2,1\$  
 4832 020752 012700 021074 MOV #SSBTP3+2,R0  
 4833 020756 012760 021052 177776 MOV #SSBTP1,-2(R0)  
 4834 020764 042737 100000 021052 BIC #100000,#SSBTP1 ;MAKE THE OPERAND POSITIVE,  
 4835 020772 012737 021006 001236 MOV #SSB2,#\$TMP2  
 4836 021000 012701 000200 MOV #200,R1 ;SET FD,  
 4837 021004 170101 LDFPS R1  
 4838  
 4839 021006 170750 SSB2: NEGD @-(R0) ;TEST INSTRUCTION,  
 4840  
 4841 021010 170205 STFPS R5 ;GET FPS,  
 4842 021012 012701 021052 MOV #SSBTP1,R1 ;IS THE RESULT CORRECT,  
 4843 021016 012702 021062 MOV #SSBTP2,R2  
 4844 021022 012703 000004 MOV #4,R3  
 4845 021026 022122 1\$: CMP (R1)+,(R2)+  
 4846 021030 001021 BNE SSB10 ;BRANCH IF INCORRECT,  
 4847 021032 077303 SOB R3,1\$  
 4848 021034 022700 021072 CMP #SSBTP3,R0 ;IS R0 CORRECT,  
 4849 021040 001025 BNE SSB15 ;BRANCH IF INCORRECT,  
 4850 021042 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?  
 4851 021046 001031 BNE SSB20 ;BRANCH IF INCORRECT,  
 4852 021050 000436 BR SSBDONE

4853 ;THESE ARE DATA TABLES AND A DATA BUFFER,  
 4854

4855 021052 023245 SSBTP1: 023245  
 4856 021054 026720 26720  
 4857 021056 122324 122324  
 4858 021060 052672 52672  
 4859 021062 123245 SSBTP2: 123245  
 4860 021064 026270 26270  
 4861 021066 122324 122324  
 4862 021070 052672 52672  
 4863 021072 021052 SSBTP3: SSBTP1

4864 ;REPORT RESULT INCORRECT:  
 4865  
 4866 021074 012737 021052 001240 SSB10: MOV #SSBTP1,#\$TMP3  
 4867 021102 012737 021062 001242 MOV #SSBTP2,#\$TMP4  
 4868 021110 104154 1\$: ERROR 154 ;BAD DATA



```

4869 021112 000415 BR SSBDONE
4870
4871 ;REPORT R0 INCORRECT:
4872 021114 012737 021072 001240 SSB15: MOV #SSBTP3,@#STMP3
4873 021122 010037 001242 MOV R0,@#STMP4
4874 021126 104155 1$: ERROR 155 ;SPEC DESTX R0X
4875 021130 000406 BR SSBDONE
4876
4877 ;REPORT FPS INCORRECT:
4878 021132 012737 000210 001240 SSB20: MOV #210,@#STMP3
4879 021140 010537 001242 MOV R5,@#STMP4
4880 021144 104156 1$: ERROR 156
4881
4882 021146 SSBDONE:
4883 021146 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK, AND
4884 ;SEE IF THE USER HAS EXPRESSED
4885 ;THE DESIRE TO CHANGE THE SOFTWARE
4886 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
4887 ;THE USER TYPED CONTROL G?).
4888 ;:*****
4889 ;*TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST
4890 ;*
4891 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
4892 ;*MODE 2 USING THE NEGF INSTR.
4893 ;*
4894 ;:*****
4895 021150 000004 TST40: SCOPE
4896 021152 TTB1:
4897 021152 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4898 021154 012701 021264 MOV #TTBTP1,R1 ;SET UP THE DATA BUFFER,
4899 021160 012700 021274 MOV #TTBTP2,R0
4900 021164 012702 000004 MOV #4,R2
4901 021170 012021 1$: MOV (R0)+,(R1)+
4902 021172 077202 SOB R2,1$
4903 021174 012700 021264 MOV #TTBTP1,R0
4904 021200 042710 100000 BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
4905 021204 012737 021220 001236 MOV #TTB2,@#STMP2
4906 021212 012701 000000 MOV #000,R1 ;SET FD.
4907 021216 170101 LDFPS R1
4908
4909 021220 170720 TTB2: NEGF (R0)+ ;TEST INSTRUCTION.
4910
4911 021222 170205 STFPS R5 ;GET FPS.
4912 021224 012701 021264 MOV #TTBTP1,R1 ;IS THE RESULT CORRECT.
4913 021230 012702 021274 MOV #TTBTP2,R2
4914 021234 012703 000004 MOV #4,R3
4915 021240 022122 1$: CMP (R1)+,(R2)+
4916 021242 001020 BNE TTB10 ;BRANCH IF INCORRECT.
4917 021244 077303 SOB R3,1$
4918 021246 022700 021270 CMP #TTBTP1+4,R0 ;IS R0 CORRECT.
4919 021252 001024 BNE TTB15 ;BRANCH IF INCORRECT.
4920 021254 022705 000010 CMP #010,R5 ;IS THE FPS CORRECT?
4921 021260 001030 BNE TTB20 ;BRANCH IF INCORRECT.
4922 021262 000435 BR TTBDONE
4923
4924 ;THESE ARE DATA TABLES AND A DATA BUFFER.
  
```

4925 021264 023245  
 4926 021266 026720  
 4927 021270 122324  
 4928 021272 052672  
 4929 021274 123245  
 4930 021276 026720  
 4931 021300 122324  
 4932 021302 052672

TTBTP1: 023245  
 26720  
 122324  
 52672  
 TTBTP2: 123245  
 26720  
 122324  
 52672

4933  
 4934  
 4935 021304 012737 021264 001240  
 4936 021312 012737 021274 001242  
 4937 021320 104150  
 4938 021322 000415

;REPORT RESULT INCORRECT:  
 TTB10: MOV #TTBTP1,0##\$TMP3  
 MOV #TTBTP2,0##\$TMP4  
 1\$: ERROR 150 ;BAD DATA  
 BR TTBDONE

4939  
 4940  
 4941 021324 012737 021270 001240  
 4942 021332 010037 001242  
 4943 021336 104160  
 4944 021340 000406

;REPORT R0 INCORRECT:  
 TTB15: MOV #TTBTP1+4,0##\$TMP3  
 MOV R0,0##\$TMP4  
 1\$: ERROR 160 ;SPEC DESTX R0X  
 BR TTBDONE

4945  
 4946  
 4947 021342 012737 000010 001240  
 4948 021350 010537 001242  
 4949 021354 104161  
 4950

;REPORT FPS INCORRECT:  
 TTB20: MOV #010,0##\$TMP3  
 MOV R5,0##\$TMP4  
 1\$: ERROR 161

4951 021356  
 4952 021356 104412  
 4953  
 4954  
 4955  
 4956

TTBDONE;  
 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?).

4957  
 4958  
 4959  
 4960  
 4961  
 4962  
 4963

;\*\*\*\*\*  
 ;\*TEST 41 SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST  
 ;\*  
 ;\*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS  
 ;\*MODE 2(IMMEDIATE) USING THE NEGD INSTR.  
 ;\*

4964 021360 000004  
 4965 021362  
 4966 021362 104413  
 4967 021364 012700 021510  
 4968 021370 012701 021436  
 4969 021374 012702 000004  
 4970 021400 012021  
 4971 021402 077202  
 4972 021404 012700 021436  
 4973 021410 042737 100000 021436  
 4974 021416 012737 021434 001236  
 4975 021424 012701 000200  
 4976 021430 170101  
 4977 021432 005001  
 4978

;\*\*\*\*\*  
 TST41: SCOPE  
 UUB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 MOV #UUBTP2,R0  
 MOV #UUBTP1,R1 ;SET UP THE DATA BUFFER,  
 MOV #4,R2  
 1\$: MOV (R0)+,(R1)+  
 SOB R2,1\$  
 MOV #UUBTP1,R0  
 BIC #100000,0##\$UUBTP1 ;MAKE THE OPERAND POSITIVE.  
 MOV #UUB2,0##\$TMP2  
 MOV #200,R1 ;SET FD,  
 LDFPS R1  
 CLR R1

4979 021434 170727  
 4980 021436 005201 005201 005201

UUB2: NEGD (R7)+ ;TEST INSTRUCTION,  
 UUBTP1: 5201,5201,5201,5201

```

4981 021444 005201
4982 ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3,
4983 021446 170205 STFPS R5 ;GET FPS,
4984 021450 012703 021436 MOV #UUBTP1,R3 ;IS THE RESULT CORRECT,
4985 021454 012702 021510 MOV #UUBTP2,R2
4986 021460 012704 000004 MOV #4,R4
4987 021464 022322 1$: CMP (R3)+,(R2)+
4988 021466 001014 BNE UUB10 ;BRANCH IF INCORRECT,
4989 021470 077403 SOB R4,1$
4990 021472 022701 000003 CMP #3,R1 ;WAS R1 INCREMENTED CORRECTLY,
4991 021476 001027 BNE UUB15 ;BRANCH IF INCORRECT,
4992 021500 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
4993 021504 001015 BNE UUB20 ;BRANCH IF INCORRECT,
4994 021506 000436 BR UUBDONE
  
```

```

4995
4996 ;THESE ARE DATA TABLE.
4997 021510 105201 UUBTP2: 105201
4998 021512 005201 5201
4999 021514 005201 5201
5000 021516 005201 5201
  
```

```

5001
5002 ;REPORT RESULT INCORRECT:
5003 021520 012737 021436 001240 UUB10: MOV #UUBTP1,@$TMP3
5004 021526 012737 021510 001242 MOV #UUBTP2,@$TMP4
5005 021534 104162 1$: ERROR 162 ;BAD DATA
5006 021536 000422 BR UUBDONE
  
```

```

5007
5008 ;REPORT FPS INCORRECT:
5009 021540 012737 000210 001240 UUB20: MOV #210,@$TMP3
5010 021546 010537 001242 MOV R5,@$TMP4
5011 021552 104163 1$: ERROR 163 ;FPS
5012 021554 000413 BR UUBDONE
  
```

```

5013
5014 ;REPORT PC INCORRECTLY INCREMENTED DURING EXECUTION,
5015 021556 162701 000003 UUB15: SUB #3,R1
5016 021562 006301 ASL R1
5017 021564 012702 021440 MOV #UUBTP1+2,R2
5018 021570 010237 001240 MOV R2,@$TMP3
5019 021574 160102 SUB R1,R2
5020 021576 010237 001242 MOV R2,@$TMP4
5021 021602 104164 1$: ERROR 164 ;PC BAD CONSTAND B GR7X
  
```

```

5022
5023 UUBDONE:
5024 021604 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK, AND
5025 ;SEE IF THE USER HAS EXPRESSED
5026 ;THE DESIRE TO CHANGE THE SOFTWARE
5027 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5028 ;THE USER TYPED CONTROL G?).
  
```

```

5029 ;*****
5030 ;*TEST 42 SPECIAL DEST, MODE 6, TEST
5031 ;*
5032 ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
5033 ;*MODE 6 USING THE NEGD INSTR.
5034 ;*
5035 ;*****
5036 021606 000004 TST42: SCOPE
  
```

```

5037 021610
5038 021610 104413
5039 021612 012701 021734
5040 021616 012700 021744
5041 021622 012702 000004
5042 021626 012021
5043 021630 077202
5044 021632 012700 014533
5045 021636 042737 100000 021734
5046 021644 012737 021662 001236
5047 021652 012701 000200
5048 021656 170101
5049
5050 021660 005001
5051 021662 170760 005201
5052
5053 021666 170205
5054 021670 005701
5055 021672 001030
5056 021674 012701 021734
5057 021700 012702 021744
5058 021704 012703 000004
5059 021710 022122
5060 021712 001030
5061 021714 077303
5062 021716 022700 014533
5063 021722 001034
5064 021724 022705 000210
5065 021730 001040
5066 021732 000445
5067
5068
5069 021734 023245
5070 021736 026720
5071 021740 122324
5072 021742 052672
5073 021744 123245
5074 021746 026720
5075 021750 122324
5076 021752 052672
5077
5078
5079
5080 021754 012737 021664 001242
5081 021762 012737 021666 001240
5082 021770 104215
5083 021772 000425
5084
5085
5086 021774 012737 021734 001240
5087 022002 012737 021744 001242
5088 022010 104216
5089 022012 000415
5090
5091
5092 022014 012737 014533 001240
  
```

```

XXB1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #XXBTP1,R1 ;SET UP THE DATA BUFFER.
MOV #XXBTP2,R0
MOV #4,R2
1$: MOV (R0)+,(R1)+
SOB R2,1$
MOV #XXBTP1-5201,R0
BIC #100000,0#XXBTP1;MAKE OPERAND POSITIVE.
MOV #XXB2,0#TMP2
MOV #200,R1 ;SET FD.
LDFPS R1

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

STFPS R5 ;GET FPS.
TST R1
BNE XXB25 ;WAS THE PC CORRECT AFTER EXECUTION?
MOV #XXBTP1,R1 ;IS THE RESULT CORRECT.
MOV #XXBTP2,R2
MOV #4,R3
1$: CMP (R1)+,(R2)+
BNE XXB10 ;BRANCH IF INCORRECT.
SOB R3,1$
CMP #XXBTP1-5201,R0 ;IS R0 CORRECT.
BNE XXB15 ;BRANCH IF INCORRECT.
CMP #210,R5 ;IS THE FPS CORRECT?
BNE XXB20 ;BRANCH IF INCORRECT.
BR XXBDONE

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

;REPORT PC INCORRECT AFTER EXECUTION.
XXB25: MOV #XXB2+2,0#TMP4
MOV #XXB2+4,0#TMP3
1$: ERROR 215 ;PC NOT INCREMENTED BY 2.
BR XXBDONE

;REPORT RESULT INCORRECT:
XXB10: MOV #XXBTP1,0#TMP3
MOV #XXBTP2,0#TMP4
1$: ERROR 216 ;BAD DATA
BR XXBDONE

;REPORT R0 INCORRECT:
XXB15: MOV #XXBTP1-5201,0#TMP3
  
```

```

5093 022022 010037 001242          MOV      R0,@#STMP4
5094 022026 104217          1$:     ERROR    217          ;SPEC DESTX R0X
5095 022030 000406          BR       XXBDONE
5096
5097
5098          ;REPORT FPS INCORRECT:
5099 022032 012737 000210 001240  XXB20:  MOV      #210,@#STMP3
5100 022040 010537 001242          MOV      R5,@#STMP4
5101 022044 104220          1$:     ERROR    220
5102
5103 022046
5104 022046 104412          XXBDONE: RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
5105
5106
5107
5108
5109
5110
5111          ;*****
5112          ;*TEST 43          SPECIAL DEST, MODE 7, TEST
5113          ;*
5114          ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
5115          ;*MODE 7 USING THE NEGD INSTR.
5116          ;*
5117 022050 000004          ;*****
5118          TST43:  SCOPE
5119
5120 022052          YYB1:          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
5121 022054 012701 022204          MOV      #YYBTP1,R1          ;SET UP THE DATA BUFFER,
5122 022060 012700 022214          MOV      #YYBTP2,R0
5123 022064 012702 000004          MOV      #4,R2
5124 022070 012021          1$:     MOV      (R0)+,(R1)+
5125 022072 077202          SOB      R2,1$
5126 022074 012700 015023          MOV      #YYBTP3-5201,R0
5127 022100 012760 022204 005201          MOV      #YYBTP1,5201(R0)
5128 022106 042737 100000 022204          BIC      #100000,@#YYBTP1          ;MAKE THE OPERAND POSITIVE,
5129 022114 012737 022132 001236          MOV      #YYB2,@#STMP2
5130 022122 012701 000200          MOV      #200,R1          ;SET FD,
5131 022126 170101          LDFPS   R1
5132
5133 022130 005001          YYB2:          CLR      R1
5134 022132 170770 005201          YYB2:          NEGD      @5201(R0)          ;TEST INSTRUCTION,
5135
5136 022136 170205          STFPS   R5          ;GET FPS,
5137 022140 005701          TST     R1          ;WAS THE PC CORRECT AFTER EXECUTION?
5138 022142 001031          BNE     YYB25
5139 022144 012701 022204          MOV      #YYBTP1,R1          ;IS THE RESULT CORRECT,
5140 022150 012702 022214          MOV      #YYBTP2,R2
5141 022154 012703 000004          MOV      #4,R3
5142 022160 022122          1$:     CMP      (R1)+,(R2)+
5143 022162 001031          BNE     YYB10          ;BRANCH IF INCORRECT,
5144 022164 077303          SOB     R3,1$
5145 022166 022700 015023          CMP      #YYBTP3-5201,R0          ;IS R0 CORRECT,
5146 022172 001035          BNE     YYB15          ;BRANCH IF INCORRECT,
5147 022174 022705 000210          CMP      #210,R5          ;IS THE FPS CORRECT?
5148 022200 001041          BNE     YYB20          ;BRANCH IF INCORRECT,
  
```

```

5149 022202 000446 BR YYBDONE
5150
5151 ;THESE ARE DATA TABLES AND A DATA BUFFER.
5152 022204 023245 YYBTP1: 023245
5153 022206 026720 26720
5154 022210 122324 122324
5155 022212 052672 52672
5156 022214 123245 YYBTP2: 123245
5157 022216 026720 26720
5158 022220 123324 123324
5159 022222 052672 52672
5160 022224 022204 YYBTP3: YYBTP1
5161
5162 ;REPORT PC INCORRECT AFTER EXECUTION.
5163 022226 016737 177702 001242 YYB25: MOV YYB2+2,@#STMP4
5164 022234 016737 177676 001240 MOV YYB2+4,@#STMP3
5165 022242 104221 1$: ERROR 221 ;PC NOT INCREMENTED BY 2.
5166 022244 000425 BR YYBDONE
5167
5168 ;REPORT RESULT INCORRECT:
5169 022246 012737 022204 001240 YYB10: MOV #YYBTP1,@#STMP3
5170 022254 012737 022214 001242 MOV #YYBTP2,@#STMP4
5171 022262 104222 1$: ERROR 222 ;BAD DATA
5172 022264 000415 BR YYBDONE
5173
5174 ;REPORT R0 INCORRECT:
5175 022266 012737 015023 001240 YYB15: MOV #YYBTP3-5201,@#STMP3
5176 022274 010037 001242 MOV R0,@#STMP4
5177 022300 104223 1$: ERROR 223 ;SPEC DESTX R0X
5178 022302 000406 BR YYBDONE
5179
5180 ;REPORT FPS INCORRECT:
5181 022304 012737 000210 001240 YYB20: MOV #210,@#STMP3
5182 022312 010537 001242 MOV R5,@#STMP4
5183 022316 104224 1$: ERROR 224
5184
5185 YYBDONE:
5186 022320 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
5187 ;SEE IF THE USER HAS EXPRESSED
5188 ;THE DESIRE TO CHANGE THE SOFTWARE
5189 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5190 ;THE USER TYPED CONTROL G?).
5191 ;*****
5192 ;*TEST 44 NEG D, ABS D AND TSTD TEST
5193 ;*
5194 ;*THIS IS A TEST OF THE NEG D ABS D AND TSTD INSTRUCTIONS,
5195 ;*
5196 ;*****
5197 022322 000004 TST44: SCOPE
5198 ;TEST NEG D WITH POS NONZERO OPERAND
5199 022324 WWB1:
5200 022324 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5201 022326 004767 000634 JSR PC,NATSUB
5202 022332 000000 1$: 0 ;FLAG=NEG D.
5203 022334 016341 2$: 16341 ;OPERAND.
5204 022336 055772 55772
  
```

5205	022340	021133		21133	
5206	022342	055447		55447	
5207	022344	116341	3s:	116341	;RESULT.
5208	022346	055772		55772	
5209	022350	021133		21133	
5210	022352	055447		55447	
5211	022354	016341	4s:	16341	;ERROR RES.
5212	022356	055772		55772	
5213	022360	021133		21133	
5214	022362	055447		55447	
5215	022364	000207	5s:	207	;FPS BEFORE EXECUTION.
5216	022366	000210		210	;FPS AFTER EXECUTION.
5217	022370	000200		200	;ERROR FPS.
5218	022372	177777		-1	;FEC
5219	022374	104200	6s:	ERROR 200	;E10<---E10*200X ST 336
5220	022376	000401		BR 7s	
5221	022400	104201		ERROR 201	;BUT ENBT ST 336X WENT TO 053 INTO 453
5222	022402		7s:		
5223			;TEST NEGD WITH NEG OPERAND.		
5224	022402		WWB2:		
5225	022402	104413		LPERR	;SET UP THE LOOP ON ERROR ADDRESS,
5226	022404	004767	000556	JSR PC,NATSUB	
5227	022410	000000	1s:	0	;FLAG=NEGD,
5228	022412	152525	2s:	152525	;OPERAND,
5229	022414	053545		53545	
5230	022416	055565		55565	
5231	022420	057505		57505	
5232	022422	052525	3s:	52525	;RESULT,
5233	022424	053545		53545	
5234	022426	055565		55565	
5235	022430	057505		57505	
5236	022432	152525	4s:	152525	;ERROR RES.
5237	022434	053545		53545	
5238	022436	055565		55565	
5239	022440	057505		57505	
5240	022442	000217	5s:	217	;FPS BEFORE EXECUTION.
5241	022444	000200		200	;FPS AFTER EXECUTION.
5242	022446	000210		210	;ERROR FPS.
5243	022450	177777		-1	;FEC
5244	022452	104200	6s:	ERROR 200	;E10<---E10*200X S336
5245	022454	000401		BR 7s	
5246	022456	104202		ERROR 202	;BUT ENBT X ST336 TO 453 INTO 053
5247	022460		7s:		
5248			;TEST ABSD WITH POSITIVE OPERAND		
5249	022460		WWB3:		
5250	022460	104413		LPERR	;SET UP THE LOOP ON ERROR ADDRESS,
5251	022462	004767	000500	JSR PC,NATSUB	
5252	022466	000001	1s:	1	;FLAG=ABSD,
5253	022470	060705	2s:	60705	;OPERAND,
5254	022472	124735		124735	
5255	022474	060124		60124	
5256	022476	073560		73560	
5257	022500	060705	3s:	60705	;RESULT,
5258	022502	124735		124735	
5259	022504	060124		60124	
5260	022506	073560		73560	

5261	022510	160705		4s:	160705		;ERROR RES.
5262	022512	124735			124735		
5263	022514	060124			60124		
5264	022516	073560			73560		
5265	022520	000217		5s:	217		;FPS BEFORE EXECUTION.
5266	022522	000200			200		;FPS AFTER EXECUTION.
5267	022524	000210			210		;ERROR FPS.
5268	022526	177777			-1		;EITHER BUT OP1B
5269	022530	104203		6s:	ERROR 203		;BUT ST 055 TO 336 INTO 335
5270	022532	000401			BR 7s		
5271	022534	104203			ERROR 203		;OR BUT ENBT ST 335 TO 452 INTO 052
5272	022536			7s:			
5273					;TEST ABSD WITH NEG. OPERAND		
5274	022536			WWB4:			
5275	022536	104413			LPERR		;SET UP THE LOOP ON ERROR ADDRESS.
5276	022540	004767	000422		JSR PC,NATSUB		
5277	022544	000001		1s:	1		;FLAG=ABSD.
5278	022546	154345		2s:	154345		;OPERAND.
5279	022550	076567			76567		
5280	022552	032123			32123		
5281	022554	043234			43234		
5282	022556	054345		3s:	54345		;RESULT.
5283	022560	076567			76567		
5284	022562	032123			32123		
5285	022564	043234			43234		
5286	022566	154345		4s:	154345		;ERROR RES.
5287	022570	076567			76567		
5288	022572	032123			32123		
5289	022574	043234			43234		
5290	022576	000217		5s:	217		;FPS BEFORE EXECUTION.
5291	022600	000200			200		;FPS AFTER EXECUTION.
5292	022602	177777			-1		;ERROR FPS.
5293	022604	177777			-1		
5294	022606	104204		6s:	ERROR 204		;E10*E10*200X ST 452
5295	022610	000401			BR 7s		
5296	022612	104171			ERROR 171		
5297	022614			7s:			
5298					;TEST WITH POSITIVE OP		
5299	022614			WWB5:			
5300	022614	104413			LPERR		;SET UP THE LOOP ON ERROR ADDRESS.
5301	022616	004767	000344		JSR PC,NATSUB		
5302	022622	000002		1s:	2		;FLAG=TSTD.
5303	022624	012321		2s:	12321		;OPERAND.
5304	022626	045654			45654		
5305	022630	070107			70107		
5306	022632	034543			34543		
5307	022634	012321		3s:	12321		;RESULT.
5308	022636	045654			45654		
5309	022640	070107			70107		
5310	022642	034543			34543		
5311	022644	112321		4s:	112321		;ERROR RES.
5312	022646	045654			45654		
5313	022650	070107			70107		
5314	022652	034543			34543		
5315	022654	000217		5s:	217		;FPS BEFORE EXECUTION.
5316	022656	000200			200		;FPS AFTER EXECUTION.



```

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

```

```

5373 ;TEST TSTD =0 OP FIUV=0
5374 023026 104413 ;WB8:
5375 023026 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
5376 023030 004767 000132 JSR PC,NATSUB
5377 023034 000002 1$: 2 ;FLAG=TSTD,
5378 023036 100123 2$: 100123 ;OPERAND,
5379 023040 021012 21012
5380 023042 034565 34565
5381 023044 043210 43210
5382 023046 100123 3$: 100123 ;RESULT,
5383 023050 021012 21012
5384 023052 034565 34565
5385 023054 043210 43210
5386 023056 000000 4$: 0 ;ERROR RES,
5387 023060 000000 0
5388 023062 000000 0
5389 023064 000000 0
5390 023066 040203 5$: 40203 ;FPS BEFORE EXECUTION,
5391 023070 040214 040214 ;FPS AFTER EXECUTION,
5392 023072 140214 140214 ;ERROR FPS,
5393 023074 177777 -1
5394 023076 104211 6$: ERROR 211 ;+
5395 023100 000401 BR 7$
5396 023102 104213 ERROR 213 ;BUT FIUV ST 257 TO 355 INTO 255
5397 023104
5398 ;TEST TSTD =0 OP FIUV=1
5399 023104 ;WB9:
5400 023104 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
5401 023106 004767 000054 JSR PC,NATSUB
5402 023112 000002 1$: 2 ;FLAG=TSTD,
5403 023114 100137 2$: 100137 ;OPERAND,
5404 023116 024613 24613
5405 023120 057024 57024
5406 023122 060137 60137
5407 023124 100137 3$: 100137 ;RESULT,
5408 023126 024613 24613
5409 023130 057024 57024
5410 023132 060137 60137
5411 023134 000000 4$: 0 ;ERROR RES,
5412 023136 000000 0
5413 023140 000000 0
5414 023142 000000 0
5415 023144 044200 5$: 44200 ;FPS BEFORE EXECUTION,
5416 023146 144214 144214 ;FPS AFTER EXECUTION,
5417 023150 044214 044214 ;ERROR FPS,
5418 023152 000014 14
5419 023154 104211 6$: ERROR 211 ;+
5420 023156 000401 BR 7$
5421 023160 104214 ERROR 214 ;BUT FIUV ST 257 TO 255 INTO 355
5422 023162
5423 023162 000167 000414 JMP WWBDONE
5424
5425 ;THIS SUBROUTINE, NATSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
5426 ;THE EITHER A TSTD, AN ABSD OR A NEGD INSTRUCTION AND CHECK THE RESULTS, A CALL
5427 ;TO IT IS MADE THUS:
5428 ;

```

```

5429 ; JSR PC,@#NATSUB
5430 ; FLAG: .WORD X ;INSTRUCTION TYPE FLAG.
5431 ; ACARG: .WORD X,X,X,X ;OPERAND
5432 ; RES: .WORD X,X,X,X ;EXPECTED RESULT
5433 ; ERRES: .WORD X,X,X,X ;ERROR RESULT
5434 ; FPSB: .WORD X ;FPS BEFORE EXECUTION
5435 ; FPSA: .WORD X ;FPS AFTER EXECUTION
5436 ; FEC: .WORD X ;EXPECTED FEC
5437 ; ERFPS: .WORD X ;ERROR FPS.
5438 ; ERR1: ERROR X ;DATA ERROR,
5439 ; BR CONT
5440 ; ERR2: ERROR X ;FPS ERROR.
5441 ; CONT: ;RETURN ADDRESS
5442 ;
5443 ;THE OPERAND IS SET UP IN NATBF1. THEN
5444 ;THE EITHER THE TSTD, NEGD OR ABSD INSTRUCTION IS EXECUTED.
5445 ;NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
5446 ;IS TO BE EXECUTED; 0 = NEGD, 1 = ABSD, 2 = TSTD.
5447 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
5448 ;COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL
5449 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
5450 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
5451 ;TO THE ERROR CALL AT ERR2, OTHERWISE NATSUB ITSELF
5452 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
5453 ;INSTRUCTION IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
5454 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
5455 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN NATSUB
5456 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
5457 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
5458 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT,

```

```

5459
5460
5461 023166 012601 NATSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS,
5462 023170 010102 MOV R1,R2 ;COPY THE OPERAND,
5463 023172 062702 000002 ADD #2,R2
5464 023176 012703 023570 MOV #NATBF1,R3
5465 023202 012704 000004 MOV #4,R4
5466 023206 012223 1$: MOV (R2)+,(R3)+
5467 023210 077402 SOB R4,1$
5468 023212 016100 000032 MOV 32(R1),R0 ;LOAD THE FPS.
5469 023216 170100 LDFPS R0
5470 023220 012700 023570 MOV #NATBF1,R0 ;SET UP THE OPERAND ADDRESS,
5471 023224 011102 MOV (R1),R2 ;GET THE FLAG TO DETERMINE WHICH
5472 023226 006302 ASL R2 ;INSTRUCTION TO EXECUTE.
5473 023230 006302 ASL R2 ;0 = NEGD, 1 = ABSD, 2 = TSTD
5474 023232 012703 023246 MOV #NATINS,R3
5475 023236 060203 ADD R2,R3
5476 023240 010337 001236 MOV R3,@$TMP2
5477 023244 000113 JMP (R3) ;GO EXECUTE THE INSTRUCTION,
5478 023246 170710 NATINS: NEGD (R0)
5479 023250 000403 BR 2$
5480 023252 170610 ABSD (R0)
5481 023254 000401 BR 2$
5482 023256 170510 TSTD (R0)
5483
5484 023260 170204 2$: STFPS R4 ;GET THE FPS.

```

```

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

```

```

5541 023464 062702 023536      ADD    #NATER2,R2
5542 023470 000112      JMP    (R2)                ;GO TO THE PROPER ERROR CALL.
5543
5544      ;REPORT THAT THE FEC WAS INCORRECT.
5545 023472 016137 000040 001256 206:  MOV    40(R1),@#STMP12
5546 023500 010537 001254      MOV    R5,@#STMP11
5547 023504 011102      MOV    (R1),R2
5548 023506 006302      ASL   R2
5549 023510 006302      ASL   R2
5550 023512 062702 023552      ADD    #NATER3,R2
5551 023516 000112      JMP    (R2)                ;GO TO THE PROPER ERROR CALL.
5552
5553      ;THESE ARE THE ERROR CALLS FOR EACH INDIVIDUAL INSTRUCTION AND CONDITION.
5554 023520 104165  NATER1: ERROR 165                ;NEG0 BAD DATA
5555 023522 000403      BR    NATRET
5556 023524 104166      ERROR 166                ;ABSD BAD DATA
5557 023526 000401      BR    NATRET
5558 023530 104167      ERROR 167                ;TSTD BAD DATA
5559 023532 000161 000050  NATRET: JMP    50(R1)
5560
5561      ;FPS INCORRECT:
5562 023536 104170  NATER2: ERROR 170                ;NEG0 FPSX
5563 023540 000774      BR    NATRET
5564 023542 104171      ERROR 171                ;ABSD FPSX
5565 023544 000772      BR    NATRET
5566 023546 104172      ERROR 172                ;TSTD FPSX
5567 023550 000770      BR    NATRET
5568
5569      ;FEC INCORRECT:
5570 023552 104173  NATER3: ERROR 173                ;NEG0 FECX
5571 023554 000766      BR    NATRET
5572 023556 104174      ERROR 174                ;ABSD FECX
5573 023560 000764      BR    NATRET
5574 023562 104175      ERROR 175                ;TSTD FECX
5575 023564 000762      BR    NATRET
5576
5577 023566 177777      .WORD -1
5578 023570 177777 177777 177777  NATBF1: .WORD -1,-1,-1,-1,-1
5579 023576 177777 177777
5580
5581 023602      WWBDONE:
5582 023602 104412      RSETUP                    ;GO INITIALIZE THE FPS AND STACK; AND
5583                                     ;SEE IF THE USER HAS EXPRESSED
5584                                     ;THE DESIRE TO CHANGE THE SOFTWARE
5585                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5586                                     ;THE USER TYPED CONTROL G?).
5587
5588
5589
5590      ;*****
5591      ;*TEST 45      SOURCE MODES, MODE 1 (FL=0), TEST
5592      ;*
5593      ;* THIS IS A TEST OF SOURCE MODE 1
5594      ;* USING THE LDFPS INSTR
5595      ;*
5596      ;*****

```

```

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

```

5653  
5654  
5655  
5656  
5657  
5658  
5659  
5660  
5661  
5662  
5663  
5664  
5665 023764 000004  
5666  
5667 023766  
5668 023766 104413  
5669  
5670 023770 012700 024046  
5671 023774 012710 145212  
5672 024000 012737 145212 001240  
5673 024006 012737 024022 001236  
5674 024014 012737 024106 000004  
5675  
5676 024022 170120  
5677  
5678 024024 170205  
5679  
5680 024026 020027 024050  
5681 024032 001007  
5682 024034 022705 145212  
5683 024040 001013  
5684 024042 000436  
5685  
5686  
5687  
5688 024044 177777  
5689 024046 177777  
5690 024050 177777  
5691  
5692  
5693  
5694 024052 012737 024050 001240  
5695 024060 010037 001242  
5696 024064 104230  
5697 024066 000424  
5698  
5699  
5700 024070 012737 145212 001240  
5701 024076 010537 001242  
5702 024102 104231  
5703 024104 000415  
5704  
5705  
5706  
5707  
5708 024106

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

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

```
BBC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
MOV #BBCTP1,R0 ;SET UP TEST DATA IN BUFFER,  
MOV #145212,(R0)  
MOV #145212,@#STMP3 ;SAVE DATA IN CASE OF ERROR,  
MOV #BBC2,@#STMP2  
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 BB CDONE
```

```
;TEST BUFFER AND DATA:  
BBC10: .WORD -1  
-1
```

```
;REPORT R0 INCORRECT.  
BBC10: MOV #BBCTP1+2,@#STMP3  
MOV R0,@#STMP4  
1$: ERROR 230 ;R0 BAD BUT FSRC FAILED  
BR BB CDONE
```

```
;REPORT FPS INCORRECT.  
BBC11: MOV #145212,@#STMP3 ;REPORT FPS INCORRECT,  
MOV R5,@#STMP4  
1$: ERROR 231  
BR BB CDONE
```

```
;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:
```

```

5709 024106 011602          MOV      (SP),R2
5710 024110 020227 024024    CMP      R2,#BBC2+2
5711 024114 001405          BEQ      1$
5712 024116 020227 024026    CMP      R2,#BBC2+4
5713 024122 001402          BEQ      1$
5714 024124 000137 042620    JMP      @#CPSPUR
5715 024130 022626          1$:     CMP      (SP)+,(SP)+
5716 024132 010237 001236    MOV      R2,@#STMP2
5717 024136 104232          2$:     ERROR   232          ;ODD ADRES
5718                                     ;BUT FDSTX IN ST 771
5719
5720 024140          BBCDONE:
5721 024140 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
5722                                     ;SEE IF THE USER HAS EXPRESSED
5723                                     ;THE DESIRE TO CHANGE THE SOFTWARE
5724                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5725                                     ;THE USER TYPED CONTROL G?).
5726
5727
5728                                     ;*****
5729                                     ;*TEST 47          SOURCE MODES, MODE 4 (FL=0), TEST
5730                                     ;*
5731                                     ;* THIS IS A TEST OF SOURCE MODE 4
5732                                     ;* USING THE LDFPS INSTR
5733                                     ;*
5734                                     ;*****
5735 024142 000004          TST47:  SCOPE
5736
5737 024144          DDC1:
5738 024144 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5739
5740 024146 012700 024236    MOV      #DDCTP1+2,R0      ;SET UP THE TEST DATA BUFFER,
5741 024152 012760 105252 177776    MOV      #105252,-2(R0)
5742 024160 012737 105252 001240    MOV      #105252,@#STMP3 ;SAVE DATA IN CASE OF ERROR,
5743 024166 012737 024202 001236    MOV      #DDC2,@#STMP2
5744 024174 012737 024302 000004    MOV      #DDC20,@#ERRVEC
5745 024202 170140          DDC2:  LDFPS   -(R0)
5746 024204 170205          STFPS   R5
5747 024206 020027 024234    CMP      R0,#DDCTP1
5748 024212 001015          BNE     DDC10
5749 024214 022705 105252    CMP      #105252,R5
5750 024220 001021          BNE     DDC11
5751 024222 000444          BR      DDCDONE
5752
5753 024224 177777 177777 177777    -1,-1,-1,-1
5754 024232 177777
5755 024234 177777          DDCTP1: -1
5756 024236 177777 177777 177777    -1,-1,-1,-1
5757 024244 177777
5758
5759 024246 012737 024234 001240    DDC10:  MOV      #DDCTP1,@#STMP3
5760 024254 010037 001242          MOV      R0,@#STMP4
5761 024260 104233          1$:     ERROR   233          ;R0 BAD BUT FSRC FAILED
5762 024262 000424          BR      DDCDONE
5763 024264 012737 105252 001240    DDC11:  MOV      #105252,@#STMP3 ;REPORT FPS INCORRECT,
5764 024272 010537 001242          MOV      R5,@#STMP4

```



```

5765 024276 104234          1$:      ERROR      234
5766 024300 000415          BR          DDCDONE
5767 024302 011602          DDC20:     MOV          (SP),R2
5768 024304 020227 024204     CMP          R2,#DDC2+2
5769 024310 001405          BEQ          1$
5770 024312 020227 024206     CMP          R2,#DDC2+4
5771 024316 001402          BEQ          1$
5772 024320 000137 042620     JMP          @#CPSPUR
5773 024324 022626          1$:      CMP          (SP)+,(SP)+
5774 024326 010237 001236     MOV          R2,@#STMP2
5775 024332 104235          2$:      ERROR      235          ;DDD ADRES
5776 024334          DDCDONE:
5777 024334 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
5778          ;SEE IF THE USER HAS EXPRESSED
5779          ;THE DESIRE TO CHANGE THE SOFTWARE
5780          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5781          ;THE USER TYPED CONTROL G?).
5782          ;)*****
5783          ;*TEST 50          SOURCE MODES, MODE 3 (FL=0), TEST
5784          ;*
5785          ;* THIS IS A TEST OF SOURCE MODE 3
5786          ;* USING THE LDFPS INSTR
5787          ;*
5788          ;)*****
5789 024336 000004          TST50:     SCOPE
5790 024340          EEC1:
5791 024340 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5792 024342 012700 024444     MOV          #EECTP2,R0
5793 024346 012710 024434     MOV          #EECTP1,(R0)
5794 024352 012767 103456 000054     MOV          #103456,EECTP1
5795 024360 012737 103456 001240     MOV          #103456,@#STMP3
5796 024366 012737 024402 001236     MOV          #EEC2,@#STMP2
5797 024374 012737 024512 000004     MOV          #EEC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
5798 024402 170130          EEC2:     LDFPS          @(R0)+          ;TEST INSTRUCTION.
5799 024404 170205          STFPS          R5          ;GET THE FPS.
5800 024406 020027 024446     CMP          R0,#EECTP2+2     ;IS R0 CORRECT?
5801 024412 001021          BNE          EEC10          ;BR IF NOT.
5802 024414 022705 103456     CMP          #103456,R5     ;IS THE FPS CORRECT?
5803 024420 001025          BNE          EEC11          ;BR IF NOT.
5804 024422 000450          BR          EECDONE
5805
5806
5807          ;TEST BUFFER AND DATA:
5808 024424 177777 177777 177777     -1,-1,-1,-1
5809 024432 177777
5810 024434 177777          EECTP1:     -1
5811 024436 177777 177777 177777     -1,-1,-1
5812 024444 024434 177777 177777     EECTP2:     EECTP1,-1,-1,-1,
5813 024452 177777 000000
5814
5815
5816          ;REPORT R0 INCORRECT.
5817 024456 012737 024446 001240     EEC10:     MOV          #EECTP2+2,@#STMP3
5818 024464 010037 001242     MOV          R0,@#STMP4
5819 024470 104236          1$:      ERROR      236          ;R0 BAD BUT FSRC FAILED
5820 024472 000424          BR          EECDONE

```

5821  
 5822  
 5823 024474 012737 103456 001240  
 5824 024502 010537 001242  
 5825 024506 104237  
 5826 024510 000415  
 5827  
 5828  
 5829  
 5830 024512 011602  
 5831 024514 020227 024404  
 5832 024520 001405  
 5833 024522 020227 024406  
 5834 024526 001402  
 5835 024530 000137 042620  
 5836 024534 022626  
 5837 024536 010237 001236  
 5838 024542 104240  
 5839 024544  
 5840 024544 104412  
 5841  
 5842  
 5843  
 5844  
 5845  
 5846  
 5847  
 5848  
 5849  
 5850  
 5851  
 5852 024546 000004  
 5853 024550  
 5854 024550 104413  
 5855 024552 012700 024652  
 5856 024556 012760 024640 177776  
 5857 024564 012737 045412 024640  
 5858 024572 012737 045412 001240  
 5859 024600 012737 024550 001236  
 5860 024606 012737 024714 000004  
 5861 024614 170150  
 5862 024616 170205  
 5863 024620 020027 024650  
 5864 024624 001015  
 5865 024626 022705 045412  
 5866 024632 001021  
 5867 024634 000444  
 5868  
 5869  
 5870  
 5871 024636 177777  
 5872 024640 177777  
 5873 024642 177777 177777 177777  
 5874 024650 024640 177777 177777  
 5875 024656 177777  
 5876

```

;REPORT FPS INCORRECT.
EEC11: MOV #103456,@#STMP3 ;REPORT FPS INCORRECT.
        MOV R5,@#STMP4
1$:     ERROR 237
        BR EECDONE
;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
;FAILURE, OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
EEC20: MOV (SP),R2
        CMP R2,#EEC2+2
        BEQ 1$
        CMP R2,#EEC2+4
        BEQ 1$
        JMP @*CPSPUR
1$:     CMP (SP)+,(SP)+
        MOV R2,@#STMP2
2$:     ERROR 240 ;DDD ADRES
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?),
;*****
;*TEST 51 SOURCE MODES, MODE 5 (FL=0), TEST
;*
;* THIS IS A TEST OF SOURCE MODE 5
;* USING THE LDFPS INSTR
;*
;*****
TST51: SCOPE
FFC1:   LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
        MOV #FFCTP2+2,R0 ;SET UP THE TEST DATA BUFFER.
        MOV #FFCTP1,-2(R0)
        MOV #45412,@#FFCTP1
        MOV #45412,@#STMP3 ;SAVE DATA IN CASE OF ERROR.
        MOV #FFC1,@#STMP2
        MOV #FFC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
FFC2:   LDFPS @=(R0) ;TEST INSTRUCTION.
        STFPS R5 ;GET THE FPS.
        CMP R0,#FFCTP2 ;IS R0 CORRECT?
        BNE FFC10 ;BR IF NOT.
        CMP #45412,R5 ;IS THE FPS CORRECT?
        BNE FFC11 ;BR IF NOT.
        BR FFCDONE
;TEST BUFFER AND DATA:
        -1
FFCTP1: -1
        -1,-1,-1
FFCTP2: FFCTP1,-1,-1,-1
  
```

```

5877
5878 ;REPORT R0 INCORRECT.
5879 024660 012737 024650 001240 FFC10: MOV #FFCTP2,@#STMP3
5880 024666 010037 001242 MOV R0,@#STMP4
5881 024672 104241 1$: ERROR 241 ;R0 BAD BUT FSRC FAILED
5882 024674 000424 BR FFCDONE
5883
5884 ;REPORT FPS INCORRECT.
5885 024676 012737 045412 001240 FFC11: MOV #45412,@#STMP3 ;REPORT FPS INCORRECT.
5886 024704 010537 001242 MOV R5,@#STMP4
5887 024710 104242 1$: ERROR 242
5888 024712 000415 BR FFCDONE
5889 ;TRAP HERE THROUGH VECTOR FOUR, SEE IF THE TRAP WAS DURING
5890 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED, IF SO REPORT
5891 ;FAILURE, OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
5892 024714 011602 FFC20: MOV (SP),R2
5893 024716 020227 024616 CMP R2,#FFC2+2
5894 024722 001405 BEQ 1$
5895 024724 020227 024620 CMP R2,#FFC2+4
5896 024730 001402 BEQ 1$
5897 024732 000137 042620 JMP @#CPSPUR
5898 024736 022626 1$: CMP (SP)+,(SP)+
5899 024740 010237 001236 MOV R2,@#STMP2
5900 024744 104243 2$: ERROR 243 ;ODD ADRES
5901 024746 FFCDONE:
5902 024746 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
5903 ;SEE IF THE USER HAS EXPRESSED
5904 ;THE DESIRE TO CHANGE THE SOFTWARE
5905 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5906 ;THE USER TYPED CONTROL G?).
5907 ;*****
5908 ;*TEST 52 SOURCE MODES, MODE 6 (FL=0), TEST
5909 ;*
5910 ;* THIS IS A TEST OF SOURCE MODE 6
5911 ;* USING THE LDFPS INSTR
5912 ;*
5913 ;*****
5914 024750 000004 TST52: SCOPE
5915 024752 GGC1:
5916 024752 104413 LPPER ;SET UP THE LOOP ON ERROR ADDRESS.
5917 024754 012700 017643 MOV #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER,
5918 024760 012737 046543 025044 MOV #46543,@#GGCTP1
5919 024766 012737 046543 001240 MOV #46543,@#STMP3 ;SAVE DATA IN CASE OF ERROR,
5920 024774 012737 025012 001236 MOV #GGC2,@#STMP2
5921 025002 005001 CLR R1
5922 025004 012737 025132 000004 MOV #GGC20,@#ERRVECT ;SET UP FOR TRAPS TO 4,
5923 025012 170160 005201 GGC2: LDFPS 5201(R0) ;TEST INSTRUCTION,
5924 025016 170204 STFPS R4 ;GET THE FPS,
5925 025020 005701 TST R1 ;WAS PC CORRECT AFTER EXECUTION?
5926 025022 001033 BNE GGC25 ;BR IF NOT,
5927 025024 020027 017643 CMP R0,#GGCTP1-5201 ;IS R0 CORRECT?
5928 025030 001012 BNE GGC10 ;BR IF NOT,
5929 025032 022704 046543 CMP #46543,R4 ;IS THE FPS CORRECT?
5930 025036 001016 BNE GGC11 ;BR IF NOT,
5931 025040 000451 BR GGCDONE
5932

```

```

5933
5934 ;TEST BUFFER AND DATA:
5935 025042 177777 -1
5936 025044 177777 177777 177777 GGCTP1: -1,-1,-1,-1
5937 025052 177777
5938 025054 177777 -1
5939
5940 ;REPORT R0 INCORRECT.
5941 025056 012737 017643 001240 GGC10: MOV #GGCTP1-5201,@$TMP3
5942 025064 010037 001242 MOV R0,@$TMP4
5943 025070 104244 1$: ERROR 244 ;R0 BAD BUT FSRC FAILED
5944 025072 000434 BR GGCDONE
5945
5946 ;REPORT FPS INCORRECT.
5947 025074 012737 046543 001240 GGC11: MOV #46543,@$TMP3 ;REPORT FPS INCORRECT.
5948 025102 010437 001242 MOV R4,@$TMP4
5949 025106 104245 1$: ERROR 245
5950 025110 000425 BR GGCDONE
5951
5952 ;REPORT PC INCORRECT AFTER INSTRUCTION.
5953 025112 012737 025016 001240 GGC25: MOV #GGC2+4,@$TMP3
5954 025120 012737 025014 001242 MOV #GGC2+2,@$TMP4
5955 025126 104246 1$: ERROR 246 ;PC X
5956 025130 000415 BR GGCDONE
5957 ;TRAP HERE THROUGH VECTOR FOUR, SEE IF THE TRAP WAS DURING
5958 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED, IF SO REPORT
5959 ;FAILURE, OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
5960 025132 011602 GGC20: MOV (SP),R2
5961 025134 020227 025014 CMP R2,#GGC2+2
5962 025140 001405 BEQ 1$
5963 025142 020227 025016 CMP R2,#GGC2+4
5964 025146 001402 BEQ 1$
5965 025150 000137 042620 JMP @#CPSPUR
5966 025154 022626 1$: CMP (SP)+,(SP)+
5967 025156 010237 001236 MOV R2,@$TMP2
5968 025162 104247 2$: ERROR 247 ;ODD ADRES
5969 025164 GGCDONE:
5970 025164 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
5971 ;SEE IF THE USER HAS EXPRESSED
5972 ;THE DESIRE TO CHANGE THE SOFTWARE
5973 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
5974 ;THE USER TYPED CONTROL G?).
5975 ;*****
5976 ;*TEST 53 SOURCE MODES, MODE 7 (FL=0), TEST
5977 ;*
5978 ;* THIS IS A TEST OF SOURCE MODE 7
5979 ;* USING THE LDFPS INSTR
5980 ;*
5981 ;*****
5982 025166 000004 TST53: SCOPE
5983 025170 HHC1:
5984 025170 104413 LPPER ;SET UP THE LOOP ON ERROR ADDRESS,
5985 025172 012700 020077 MOV #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER,
5986 025176 012760 025270 005201 MOV #HHCTP1,5201(R0)
5987 025204 012737 004547 025270 MOV #4547,@#HHCTP1
5988 025212 012737 004547 001240 MOV #4547,@$TMP3 ;SAVE DATA IN CASE OF ERROR.
  
```

```

5989 025220 012737 025236 001236      MOV      #HHC2,@#STMP2
5990 025226 005001      CLR      R1
5991 025230 012737 025364 000004      MOV      #HHC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
5992 025236 170170 005201      HHC2:   LDFPS  05201(R0)          ;TEST INSTRUCTION,
5993 025242 170204      STFPS   R4          ;GET THE FPS.
5994 025244 005701      TST     R1          ;WAS PC CORRECT AFTER EXECUTION?
5995 025246 001036      BNE     HHC25       ;BR IF NOT.
5996 025250 020027 020077      CMP     R0,#HHCTP2-5201 ;IS R0 CORRECT?
5997 025254 001015      BNE     HHC10       ;BR IF NOT.
5998 025256 022704 004547      CMP     #4547,R4    ;IS THE FPS CORRECT?
5999 025262 001021      BNE     HHC11       ;BR IF NOT.
6000 025264 000454      BR      HHCDONE
6001
6002
6003      ;TEST BUFFER AND DATA:
6004 025266 177777      -1
6005 025270 177777 177777 177777 HHCTP1: .WORD -1,-1,-1,-1
6006 025276 177777
6007 025300 177777 177777 177777 HHCTP2: .WORD -1,-1,-1,-1
6008 025306 177777
6009
6010      ;REPORT R0 INCORRECT.
6011 025310 012737 020077 001240 HHC10:  MOV     #HHCTP2-5201,@#STMP3
6012 025316 010037 001242      MOV     R0,@#STMP4
6013 025322 104250      1$:    ERROR  250          ;R0 BAD BUT FSRC FAILED
6014 025324 000434      BR      HHCDONE
6015
6016      ;REPORT FPS INCORRECT.
6017 025326 012737 004547 001240 HHC11:  MOV     #4547,@#STMP3 ;REPORT FPS INCORRECT.
6018 025334 010437 001242      MOV     R4,@#STMP4
6019 025340 104251      1$:    ERROR  251
6020 025342 000425      BR      HHCDONE
6021
6022      ;REPORT PC INCORRECT AFTER INSTRUCTION.
6023 025344 012737 025242 001240 HHC25:  MOV     #HHC2+4,@#STMP3
6024 025352 012737 025240 001242      MOV     #HHC2+2,@#STMP4
6025 025360 104252      1$:    ERROR  252          ;PC X
6026 025362 000415      BR      HHCDONE
6027      ;TRAP HERE THROUGH VECTOR FOUR, SEE IF THE TRAP WAS DURING
6028      ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED, IF SO REPORT
6029      ;FAILURE, OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING,
6030 025364 011602      HHC20:  MOV     (SP),R2
6031 025366 020227 025240      CMP     R2,#HHC2+2
6032 025372 001405      BEQ     1$
6033 025374 020227 025242      CMP     R2,#HHC2+4
6034 025400 001402      BEQ     1$
6035 025402 000137 042620      JMP     @#CPSPUR
6036 025406 022626      1$:    CMP     (SP)+,(SP)+
6037 025410 010237 001236      MOV     R2,@#STMP2
6038 025414 104253      2$:    ERROR  253          ;DDD ADDRESS
6039 025416      HHCDONE:
6040 025416 104412      RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
6041      ;SEE IF THE USER HAS EXPRESSED
6042      ;THE DESIRE TO CHANGE THE SOFTWARE
6043      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
6044      ;THE USER TYPED CONTROL G?).

```

6045  
 6046  
 6047  
 6048  
 6049  
 6050  
 6051  
 6052  
 6053  
 6054  
 6055 025420 000004  
 6056  
 6057 025422  
 6058 025422 104413  
 6059 025424 012737 025450 001236  
 6060 025432 012737 025522 000004  
 6061 025440 012700 000300  
 6062 025444 170100  
 6063 025446 005001  
 6064  
 6065 025450 177027  
 6066 025452 005201  
 6067 025454 005201  
 6068 025456 005201  
 6069 025460 005201  
 6070  
 6071 025462 020127 000003  
 6072 025466 001421  
 6073  
 6074  
 6075  
 6076 025470 012704 025454  
 6077 025474 162701 000003  
 6078 025500 006301  
 6079 025502 160104  
 6080 025504 010437 001242  
 6081 025510 012737 025454 001240  
 6082 025516 104254  
 6083 025520 000404  
 6084  
 6085  
 6086  
 6087 025522 011637 001236  
 6088 025526 022626  
 6089 025530 104255  
 6090  
 6091 025532  
 6092 025532 104412  
 6093  
 6094  
 6095  
 6096  
 6097  
 6098  
 6099  
 6100

```

;*****
;*TEST 54 SOURCE MODES, MODE 2 GR7 (FL=1), TEST
;*
;* THIS IS A TEST OF THE LDCLD WITH
;* IMMEDIATE ADDRESSING MODE
;*
;*****
TST54: SCOPE

IIC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #IIC2,@$TMP2 ;SAVE DATA IN CASE OF ERROR.
MOV #IIC20,@$ERRVECT ;SET UP FOR TRAPS TO 4.
MOV #300,R0
LDFPS R0
CLR R1

IIC2: LDCLD (R7)+,AC0 ;TEST INSTRUCTION.
5201
5201
5201
5201

CMP R1,#3 ;WAS PC CORRECT AFTER EXECUTION?
BEQ IICDONE ;BR IF YES.

;REPORT PC INCORRECT AFTER INSTRUCTION.
IIC3: MOV #IIC2+4,R4
SUB #3,R1
ASL R1
SUB R1,R4
MOV R4,@$TMP4
MOV #IIC2+4,@$TMP3
1s: ERROR 254 ;BAD CONSTANT
BR IICDONE

;TRAP HERE THROUGH VECTOR FOUR, SEE IF THE TRAP WAS DURING
;EXECUTION OF THE FPS INSTRUCTION BEING TESTED, IF SO REPORT
;FAILURE, OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
IIC20: MOV (SP),@$TMP2
CMP (SP)+,(SP)+
1s: ERROR 255 ;BAD CONSTANT ODD ADD

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

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

```

6101 ;*
6102 ;* THIS IS A TEST OF THE LDCLD INSTR
6103 ;* WITH MODE 2.
6104 ;*
6105 ;)*****
6106 025534 000004 TST55: SCOPE
6107
6108 025536 TCC1:
6109 025536 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
6110 025540 016737 000014 001236 MOV TCC2,@#TMP2 ;SAVE DATA IN CASE OF ERROR.
6111 025546 012700 000300 MOV #300,R0
6112 025552 170100 LDFPS R0
6113 025554 012700 025650 MOV #TCCBF0,R0 ;SET UP THE TEST DATA BUFFER.
6114 025560 177020 TCC2: LDCLD (R0)+,AC0 ;TEST INSTRUCTION,
6115
6116 025562 170204 STFPS R4 ;GET THE FPS.
6117 025564 012701 025660 MOV #TCCBF1,R1 ;GET THE RESULT.
6118 025570 012702 000200 MOV #200,R2
6119 025574 170102 LDFPS R2
6120 025576 174011 STD AC0,(R1)
6121 025600 020027 025654 CMP R0,#TCCBF0+4 ;IS R0 CORRECT?
6122 025604 001407 BEQ TCC3
6123 ;REPORT R0 INCORRECT.
6124 025606 010037 001242 MOV R0,@#TMP4
6125 025612 012737 025654 001240 MOV #TCCBF0+4,@#TMP3
6126 025620 104256 1s: ERROR 256 ;BAD CONST
6127 025622 000422 BR TCCDONE
6128
6129 025624 022704 000300 TCC3: CMP #300,R4 ;IS THE FPS CORRECT?
6130 025630 001417 BEQ TCCDONE
6131
6132 ;REPORT FPS INCORRECT.
6133 025632 010437 001242 MOV R4,@#TMP4
6134 025636 012737 000300 001240 MOV #300,@#TMP3
6135 025644 104257 1s: ERROR 257 ;FPS X
6136 025646 000410 BR TCCDONE
6137
6138
6139 ;TEST BUFFER AND DATA:
6140 025650 001234 067076 054321 TCCBF0: ,WORD 01234,67076,54321,012345
6141 025656 012345
6142 025660 177777 177777 177777 TCCBF1: -1,-1,-1,-1
6143 025666 177777
6144
6145 TCCDONE:
6146 025670 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
6147 ;SEE IF THE USER HAS EXPRESSED
6148 ;THE DESIRE TO CHANGE THE SOFTWARE
6149 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
6150 ;THE USER TYPED CONTROL G?).
6151
6152
6153
6154 ;)*****
6155 ;*TEST 56 LDCIF AND LDCLF TEST
6156 ;*

```

```

6157 ;* THIS IS A TEST OF THE LDCIF AND
6158 ;* THE LDCLF INSTRUCTIONS.
6159 ;*
6160 ;*****
6161 025672 000004 TST56: SCOPE
6162
6163
6164 ;ZERO OPERAND FL=0
6165
6166 025674 KKC1:
6167 025674 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6168 025676 004737 027026 JSR PC,#LDCFSUB ;GO EXECUTE INSTRUCTION,
6169
6170 025702 000000 000000 1$: .WORD 0,0 ;FSRC OPERAND.
6171 025706 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
6172 025712 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6173 025716 000000 4$: 0 ;FPS BEFORE EXECUTION.
6174 025720 000004 4 ;FPS AFTER EXECUTION.
6175 025722 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
6176 025724 104260 5$: ERROR 260 ;REPORT RESULT INCORRECT.
6177 025726 000401 BR 6$
6178 025730 104261 ERROR 261 ;REPORT FPS INCORRECT.
6179 025732 6$:
6180 ;ZERO OPERAND FL=0
6181
6182 025732 KKC2:
6183 025732 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6184 025734 004737 027026 JSR PC,#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6185
6186 025740 000000 177777 1$: .WORD 0,-1 ;FSRC OPERAND.
6187 025744 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
6188 025750 004177 177400 3$: 4177,177400 ;ANTICIPATED ERRONEOUS RESULT.
6189 025754 000000 4$: 0 ;FPS BEFORE EXECUTION.
6190 025756 000004 4 ;FPS AFTER EXECUTION.
6191 025760 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
6192 025762 104262 5$: ERROR 262 ;(BUT FL) ST
6193 025764 000401 BR 6$ ;277 TO 300
6194 025766 104261 ERROR 261 ;INTO 301
6195 025770 6$:
6196 ;ZERO OPERAND FL=1
6197
6198 025770 KKC3:
6199 025770 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6200 025772 004737 027026 JSR PC,#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
6201
6202 025776 000000 000000 1$: .WORD 0,0 ;FSRC OPERAND.
6203 026002 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
6204 026006 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
6205 026012 000100 4$: 100 ;FPS BEFORE EXECUTION.
6206 026014 000104 104 ;FPS AFTER EXECUTION.
6207 026016 000004 4 ;ANTICIPATED ERRONEOUS FPS.
6208 026020 104260 5$: ERROR 260 ;REPORT RESULT INCORRECT.
6209 026022 000401 BR 6$
6210 026024 104263 ERROR 263 ;FL WAS CLR'ED
6211 026026 6$:
6212 ;OPERAND POSITIVE FL=0

```



```

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

```

KKC4:

```

LPERR
JSR PC,#LDCFSUB
1$: .WORD 40000,0
2$: .WORD 43600,0
3$: .WORD 47600,0
4$: 17
-1
5$: ERROR 264 ;ST 107
BR 6$
ERROR 261
6$:
;OPERAND=1, FL=0

```

KKC5:

```

LPERR
JSR PC,#LDCFSUB
1$: .WORD 1,0
2$: .WORD 40200,0
3$: .WORD 44200,0
4$: 17
-1
5$: ERROR 264
BR 6$
ERROR 261
6$:
;OPERAND= PATTERN FL=0

```

KKC6:

```

LPERR
JSR PC,#LDCFSUB
1$: .WORD 252,0
2$: .WORD 42052,0
3$: .WORD 46052,0
4$: 0
-1
5$: ERROR 264
BR 6$
ERROR 261
6$:
;OPERAND=-40000 FL=0

```

KKC7:

```

LPERR
JSR PC,#LDCFSUB
1$: .WORD -40000,0
2$: .WORD 143600,0
3$: .WORD 43600,0
4$: 7
10
-1
5$: ERROR 265
BR 6$

```

;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,  
BAD  
;CONSTANT 231 INSD  
;215

;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,  
;(SET SIGN) ST 146

6269 026214 104261  
 6270 026216  
 6271  
 6272  
 6273 026216  
 6274 026216 104413  
 6275 026220 004737 027026  
 6276 026224 177777 000000  
 6277 026230 140200 000000  
 6278 026234 144000 000400  
 6279 026240 000000  
 6280 026242 000010  
 6281 026244 177777  
 6282 026246 104266  
 6283 026250 000401  
 6284 026252 104261  
 6285 026254  
 6286  
 6287  
 6288 026254  
 6289 026254 104413  
 6290 026256 004737 027026  
 6291 026262 125252 000000  
 6292 026266 143652 126000  
 6293 026272 043652 126000  
 6294 026276 000007  
 6295 026300 000010  
 6296 026302 177777  
 6297 026304 104265  
 6298 026306 000401  
 6299 026310 104261  
 6300 026312  
 6301  
 6302  
 6303 026312  
 6304 026312 104413  
 6305 026314 004737 027026  
 6306 026320 040000 000000  
 6307 026324 047600 000000  
 6308 026330 043600 000000  
 6309 026334 000117  
 6310 026336 000100  
 6311 026340 177777  
 6312 026342 104267  
 6313 026344 000401  
 6314 026346 104261  
 6315 026350  
 6316  
 6317  
 6318 026350  
 6319 026350 104413  
 6320 026352 004737 027026  
 6321 026356 000000 000001  
 6322 026362 040200 000000  
 6323 026366 034200 000000  
 6324 026372 000100

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 40000,0  
 2\$: .WORD 47600,0  
 3\$: .WORD 43600,0  
 4\$: 117  
 100  
 -1  
 5\$: ERROR 267 ;ST 107  
 BR 6\$  
 ERROR 261  
 6\$:  
 ;OPERAND=1 FL=1  
 KKC11:  
 LPERR  
 JSR PC,#LDCFSUB  
 1\$: .WORD 0,1  
 2\$: .WORD 40200,0  
 3\$: .WORD 34200,0  
 4\$: 100

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

```

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

```

6381	026546	125252	125252	1\$:	.WORD	125252,125252		;FSRC OPERAND.
6382	026552	147652	125253	2\$:	.WORD	147652,125253		;EXPECTED RESULT.
6383	026556	047652	125253	3\$:	.WORD	47652,125253		;ANTICIPATED ERRONEOUS RESULT.
6384	026562	000105		4\$:	105			;FPS BEFORE EXECUTION.
6385	026564	000110			110			;FPS AFTER EXECUTION.
6386	026566	177777			-1			;ANTICIPATED ERRONEOUS FPS.
6387	026570	104265		5\$:	ERROR	265		;REPORT RESULT INCORRECT.
6388	026572	000401			BR	6\$		
6389	026574	104261			ERROR	261		;REPORT FPS INCORRECT.
6390	026576			6\$:				
6391								
6392								
6393	026576							
6394	026576	104413			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
6395	026600	004737	027026		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
6396	026604	077777	177500	1\$:	.WORD	77777,177500		;FSRC OPERAND.
6397	026610	047777	177777	2\$:	.WORD	47777,177777		;EXPECTED RESULT.
6398	026614	047777	177776	3\$:	.WORD	47777,177776		;ANTICIPATED ERRONEOUS RESULT.
6399	026620	000117		4\$:	117			;FPS BEFORE EXECUTION.
6400	026622	000100			100			;FPS AFTER EXECUTION.
6401	026624	177777			-1			;ANTICIPATED ERRONEOUS FPS.
6402	026626	104270		5\$:	ERROR	270		;ST 631 INTO RND
6403	026630	000401			BR	6\$		
6404	026632	104261			ERROR	261		;REPORT FPS INCORRECT.
6405	026634			6\$:				
6406								
6407								
6408	026634							
6409	026634	104413			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
6410	026636	004737	027026		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
6411	026642	040000	000100	1\$:	.WORD	40000,100		;FSRC OPERAND.
6412	026646	047600	000001	2\$:	.WORD	47600,1		;EXPECTED RESULT.
6413	026652	047600	000000	3\$:	.WORD	47600,0		;ANTICIPATED ERRONEOUS RESULT.
6414	026656	000102		4\$:	102			;FPS BEFORE EXECUTION.
6415	026660	000100			100			;FPS AFTER EXECUTION.
6416	026662	177777			-1			;ANTICIPATED ERRONEOUS FPS.
6417	026664	104270		5\$:	ERROR	270		;REPORT RESULT INCORRECT.
6418	026666	000401			BR	6\$		
6419	026670	104261			ERROR	261		;REPORT FPS INCORRECT.
6420	026672			6\$:				
6421								
6422								
6423	026672							
6424	026672	104413			LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
6425	026674	004737	027026		JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
6426	026700	040000	000100	1\$:	.WORD	40000,100		;FSRC OPERAND.
6427	026704	047600	000000	2\$:	.WORD	47600,0		;EXPECTED RESULT.
6428	026710	047600	000001	3\$:	.WORD	47600,1		;ANTICIPATED ERRONEOUS RESULT.
6429	026714	000157		4\$:	157			;FPS BEFORE EXECUTION.
6430	026716	000140			140			;FPS AFTER EXECUTION.
6431	026720	177777			-1			;ANTICIPATED ERRONEOUS FPS.
6432	026722	104271		5\$:	ERROR	271		;ST 631 ... INTO TRNC
6433	026724	000401			BR	6\$		
6434	026726	104261			ERROR	261		;REPORT FPS INCORRECT.
6435	026730			6\$:				
6436								

;OPERAND=100000,0 (MOST NEG #) FL=0

6437 026730  
 6438 026730 104413  
 6439 026732 004737 027026  
 6440 026736 100000 000000  
 6441 026742 144000 000000  
 6442 026746 143600 000000  
 6443 026752 000007  
 6444 026754 000010  
 6445 026756 177777  
 6446 026760 104272  
 6447 026762 000401  
 6448 026764 104261  
 6449 026766

KKC19:

```

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

```

6450  
 6451  
 6452 026766  
 6453 026766 104413  
 6454 026770 004737 027026  
 6455 026774 100000 000000  
 6456 027000 150000 000000  
 6457 027004 147600 000000  
 6458 027010 000107  
 6459 027012 000110  
 6460 027014 177777  
 6461 027016 104272  
 6462 027020 000401  
 6463 027022 104261  
 6464 027024 000506  
 6465  
 6466  
 6467  
 6468  
 6469  
 6470  
 6471  
 6472  
 6473  
 6474  
 6475  
 6476  
 6477  
 6478  
 6479  
 6480  
 6481  
 6482  
 6483  
 6484  
 6485  
 6486  
 6487  
 6488  
 6489  
 6490  
 6491  
 6492

;OPERAND=100000,0 FL=1  
 KKC20:

```

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

```

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

```

;
; JSR PC,@#LDCFSUB
; ACARG: .WORD X,X ;AC OPERAND
; RES: .WORD X,X ;EXPECTED RESULT
; ERRES: .WORD X,X ;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 AC0 AS THE ACCUMULATOR), THEN  
 ;THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED,  
 ;THE RESULT IS CHECKED AGAINST RES, IF THE RESULT IS CORRECT THEN THE FPS IS  
 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL  
 ;TO THE CALLING ROUTINE AT CONT, IF THE FPS IS BAD LDCFSUB WILL  
 ;COMPARE IT TO ERROR FPS, IF THIS MATCHES THEN LDCFSUB WILL RETURN  
 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF  
 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT, IF THE RESULT OF THE  
 ;LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE  
 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN  
 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB

```

6493 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
6494 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
6495 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT,
6496
6497 027026 012601 LDCFSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS,
6498 027030 016100 000014 MOV 14(R1),R0 ;SET THE FPS.
6499 027034 170100 LDFPS R0
6500 027036 012737 027046 001236 MOV #1$,@#TMP2
6501 027044 010100 MOV R1,R0
6502
6503 027046 177010 1$: LDCIF (R0),AC0 ;TEST INSTRUCTION LDCIF OR LDCLF.
6504
6505 027050 170204 STFPS R4 ;GET FPS.
6506 027052 012700 027232 MOV #LDCT,R0 ;GET THE RESULT.
6507 027056 012702 000200 MOV #200,R2
6508 027062 170102 LDFPS R2
6509 027064 174010 STD AC0,(R0)
6510
6511 027066 012702 027232 MOV #LDCT,R2 ;SEE IF THE RESULT WAS CORRECT.
6512 027072 010237 001242 MOV R2,@#TMP4
6513 027076 010137 001240 MOV R1,@#TMP3
6514 027102 010103 MOV R1,R3
6515 027104 062703 000004 ADD #4,R3
6516 027110 010337 001244 MOV R3,@#TMP5
6517 027114 010437 001250 MOV R4,@#TMP7
6518 027120 016137 000016 001252 MOV 16(R1),@#TMP10
6519 027126 010100 MOV R1,R0
6520 027130 062700 000004 ADD #4,R0
6521 027134 012703 000002 MOV #2,R3
6522 027140 022022 2$: CMP (R0)+,(R2)+
6523 027142 001006 BNE 10$ ;BR IF INCORRECT.
6524 027144 077303 SOB R3,2$
6525
6526 027146 026104 000016 CMP 16(R1),R4 ;SEE IF THE FPS WAS CORRECT.
6527 027152 001020 BNE 15$ ;BR IF INCORRECT.
6528 027154 000161 000030 3$: JMP 30(R1) ;RETURN.
6529
6530 ;RESULT IN CORRECT SO SEE IF THE FAILURE WAS ANTICIPATED,
6531 027160 012702 027232 10$: MOV #LDCT,R2
6532 027164 010100 MOV R1,R0
6533 027166 062700 000010 ADD #10,R0
6534 027172 012703 000002 MOV #2,R3
6535 027176 022022 11$: CMP (R0)+,(R2)+
6536 027200 001003 BNE 13$
6537 027202 077303 SOB R3,11$
6538 027204 000161 000022 JMP 22(R1)
6539
6540 ;THE FAILURE WAS NOT ANTICIPATED SO REPORT THE ERROR HERE,
6541 027210 13$:
6542
6543 027210 104260 14$: ERROR 260 ;BAD RES
6544 027212 000760 BR 3$
6545
6546
6547 ;THE FPS WAS INCORRECT SO SEE IF IT WAS ANTICIPATED,
6548 027214 026104 000020 15$: CMP 20(R1),R4

```

6549 027220 001002  
 6550 027222 000161 000026  
 6551  
 6552  
 6553 027226  
 6554 027226 104261  
 6555 027230 000751  
 6556

BNE 16s  
 JMP 26(R1)  
 ;FPS ERROR NOT ANTICIPATED SO REPORT IT HERE.  
 16s:  
 17s: ERROR 261 ;BAD FPS  
 BR 3s

6557  
 6558 027232 000000 000000 000000  
 6559 027240 000000  
 6560  
 6561 027242  
 6562 027242 104412  
 6563  
 6564  
 6565  
 6566  
 6567  
 6568

;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?).

6569  
 6570  
 6571  
 6572  
 6573  
 6574

;\*\*\*\*\*  
 ;\*TEST 57 LDCID AND LDCLD TEST  
 ;\*  
 ;\* THIS IS A TEST OF LDCID AND LDCLD  
 ;\*

6575 027244 000004

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

6576  
 6577 027246  
 6578 027246 104413  
 6579 027250 004737 030044  
 6580 027254 000000 000000  
 6581 027260 000000 000000 000000  
 6582 027266 000000  
 6583 027270 177777 177777 177777  
 6584 027276 177777  
 6585 027300 000213  
 6586 027302 000204  
 6587 027304 177777  
 6588 027306 104273  
 6589 027310 000401  
 6590 027312 104274  
 6591 027314  
 6592  
 6593 027314  
 6594 027314 104413  
 6595 027316 004737 030044  
 6596 027322 000000 177777  
 6597 027326 000000 000000 000000  
 6598 027334 000000  
 6599 027336 004177 177400 000000  
 6600 027344 000000  
 6601 027346 000200  
 6602 027350 000204  
 6603 027352 177777  
 6604 027354 104275

LLC1:  
 LPERR  
 JSR PC,#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS,  
 1s: .WORD 0,0 ;GO EXECUTE THE INSTRUCTION,  
 2s: .WORD 0,0,0,0 ;FSRC OPERAND,  
 3s: .WORD -1,-1,-1,-1 ;EXPECTED RESULT,  
 ;ANTICIPATED ERRONEOUS RESULT.  
 4s: 213 ;FPS BEFORE EXECUTION,  
 204 ;FPS AFTER EXECUTION,  
 -1 ;ANTICIPATED ERRONEOUS FPS,  
 5s: ERROR 273 ;REPORT RESULT INCORRECT.  
 BR 6s  
 ERROR 274 ;REPORT FPS INCORRECT,  
 6s:  
 ;OPERAND=0 FL=0, FD=1

LLC2:  
 LPERR  
 JSR PC,#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS,  
 1s: .WORD 0,-1 ;GO EXECUTE THE INSTRUCTION,  
 2s: .WORD 0,0,0,0 ;FSRC OPERAND,  
 3s: .WORD 4177,177400,0,0 ;EXPECTED RESULT,  
 ;ANTICIPATED ERRONEOUS RESULT.  
 4s: 200 ;FPS BEFORE EXECUTION,  
 204 ;FPS AFTER EXECUTION,  
 -1 ;ANTICIPATED ERRONEOUS FPS,  
 5s: ERROR 275 ;(BUT FL)S+277

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



```

6661 027544          LLC6:
6662 027544 104413          LPERR
6663 027546 004737 030044          JSR      PC,#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS,
6664 027552 040000 000000          1$:      .WORD  40000,0 ;GO EXECUTE THE INSTRUCTION,
6665 027556 047600 000000 000000 2$:      .WORD  47600,0,0,0 ;FSRC OPERAND,
6666 027564 000000          ;EXPECTED RESULT,
6667 027566 043600 000000 000000 3$:      .WORD  43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT,
6668 027574 000000
6669 027576 000317          317 ;FPS BEFORE EXECUTION,
6670 027600 000300          300 ;FPS AFTER EXECUTION,
6671 027602 177777          -1 ;ANTICIPATED ERRONEOUS FPS,
6672 027604 104300          5$:      ERROR  300 ;ST 107 BAD CONS
6673 027606 000401          BR      6$
6674 027610 104274          ERROR  274 ;REPORT FPS INCORRECT,
6675 027612          6$:
6676
6677          ;OPERAND=0,1 FL=1 FD=1
6678 027612          LLC7:
6679 027612 104413          LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6680 027614 004737 030044          JSR      PC,#LDCDSUB ;GO EXECUTE THE INSTRUCTION,
6681 027620 000000 000001          1$:      .WORD  0,1 ;FSRC OPERAND,
6682 027624 040200 000000 000000 2$:      .WORD  40200,0,0,0 ;EXPECTED RESULT,
6683 027632 000000
6684 027634 034200 000000 000000 3$:      .WORD  34200,0,0,0 ;ANTICIPATED ERRONEOUS RESULT,
6685 027642 000000
6686 027644 000300          4$:      300 ;FPS BEFORE EXECUTION,
6687 027646 000300          300 ;FPS AFTER EXECUTION,
6688 027650 177777          -1 ;ANTICIPATED ERRONEOUS FPS,
6689 027652 104300          5$:      ERROR  300 ;REPORT FPS INCORRECT,
6690 027654 000401          BR      6$
6691 027656 104274          ERROR  274 ;REPORT FPS INCORRECT,
6692 027660          6$:
6693
6694          ;OPERAND=77777,177777 FL=1 FD=1
6695 027660          LLC8:
6696 027660 104413          LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6697 027662 004737 030044          JSR      PC,#LDCDSUB ;GO EXECUTE THE INSTRUCTION,
6698 027666 077777 177777          1$:      .WORD  77777,177777 ;FSRC OPERAND,
6699 027672 047777 177777 177000 2$:      .WORD  47777,177777,177000,0 ;EXPECTED RESULT,
6700 027700 000000
6701 027702 177777 177777 177777 3$:      .WORD  -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT,
6702 027710 177777
6703 027712 000317          4$:      317 ;FPS BEFORE EXECUTION,
6704 027714 000300          300 ;FPS AFTER EXECUTION,
6705 027716 177777          -1 ;ANTICIPATED ERRONEOUS FPS,
6706 027720 104273          5$:      ERROR  273 ;REPORT RESULT INCORRECT,
6707 027722 000401          BR      6$
6708 027724 104274          ERROR  274 ;REPORT FPS INCORRECT,
6709 027726          6$:
6710
6711          ;OPERAND=-PATTERN FL=1 FD=1
6712
6713 027726          LLC9:
6714 027726 104413          LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6715 027730 004767 000110          JSR      PC,LDCDSUB ;GO EXECUTE THE INSTRUCTION,
6716 027734 177777 177526          1$:      .WORD  -1,-252 ;FSRC OPERAND,
  
```

```

6717 027740 142052 000000 000000 2s: .WORD 142052,0,0,0 ;EXPECTED RESULT,
6718 027746 000000
6719 027750 136052 000000 000000 3s: .WORD 136052,0,0,0 ;ANTICIPATED ERRONEOUS RESULT,
6720 027756 000000
6721 027760 000307 4s: 307 ;FPS BEFORE EXECUTION,
6722 027762 000310 310 ;FPS AFTER EXECUTION,
6723 027764 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
6724 027766 104300 5s: ERROR 300 ;REPORT RESULT INCORRECT,
6725 027770 000401 BR 6s 6s ;REPORT FPS INCORRECT,
6726 027772 104274 ERROR 274
6727 027774 6s:
6728

```

```

6729 ;OPERAND=PATTERN FL=1 FD=1 FT=1
6730 027774 LLC10:
6731 027774 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6732 027776 004767 000042 JSR PC,LDCDSUB ;GO EXECUTE THE INSTRUCTION,
6733 030002 012345 067012 1s: .WORD 12345,67012 ;FSRC OPERAND,
6734 030006 047247 025560 050000 2s: .WORD 47247,025560,050000,0 ;EXPECTED RESULT,
6735 030014 000000
6736 030016 177777 177777 3s: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT,
6737 030024 177777

```

```

6738 030026 000352 4s: 352 ;FPS BEFORE EXECUTION,
6739 030030 000340 340 ;FPS AFTER EXECUTION,
6740 030032 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
6741 030034 104273 5s: ERROR 273 ;REPORT RESULT INCORRECT,
6742 030036 000401 BR 6s 6s ;REPORT FPS INCORRECT,
6743 030040 104274 ERROR 274
6744 030042 000502 6s: BR LLCDONE
6745

```

```

6746 ;THIS SUBROUTINE, LDCDSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
6747 ;THE LDCID OR LDCLD INSTRUCTION AND CHECK THE RESULTS, A CALL
6748 ;TO IT IS MADE THUS:
6749 ;

```

```

6750 ; JSR PC,@#LDCDSUB
6751 ; ACARG: .WORD X,X ;AC OPERAND
6752 ; RES: .WORD X,X,X,X ;EXPECTED RESULT
6753 ; ERRES: .WORD X,X,X,X ;ERROR RESULT
6754 ; FPSB: .WORD X ;FPS BEFORE EXECUTION
6755 ; FPSA: .WORD X ;FPS AFTER EXECUTION
6756 ; ERFPS: .WORD X ;ERROR FPS,
6757 ; ERR1: ERROR X ;DATA ERROR,
6758 ; BR CONT
6759 ; ERR2: ERROR X ;FPS ERROR,
6760 ; CONT: ;RETURN ADDRESS
6761 ;

```

```

6762 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
6763 ;THE LDCID OR LDCLD INSTRUCTION IS EXECUTED.
6764 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
6765 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCDSUB RETURNS CONTROL
6766 ;TO THE CALLING ROUTINE AT CONT, IF THE FPS IS BAD LDCDSUB
6767 ;COMPARE IT TO ERROR FPS, IF THIS MATCHES THEN LDCDSUB WILL RETURN
6768 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCDSUB ITSELF
6769 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT, IF THE RESULT OF THE
6770 ;LDCID OR LDCLD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
6771 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN
6772 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCDSUB

```

```

6773 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
6774 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCDSUB WILL
6775 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
6776
6777 030044 012601 LDCDSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS,
6778 030046 016100 000024 MOV 24(R1),R0 ;SET THE FPS.
6779 030052 170100 LDFPS R0
6780 030054 012737 030064 001236 MOV #1$,@#STMP2
6781 030062 010100 MOV R1,R0
6782 030064 177010 1$: LDCID (R0),AC0 ;TEST INSTRUCTION, LDCID OR LDCLD,
6783
6784 030066 170204 STFPS R4 ;GET FPS.
6785 030070 012700 027232 MOV #LDCT,R0 ;GET THE RESULT.
6786 030074 012702 000200 MOV #200,R2
6787 030100 170102 LDFPS R2
6788 030102 174010 STD AC0,(R0)
6789
6790 ;SEE IF THE RESULT IS CORRECT.
6791 030104 012702 027232 MOV #LDCT,R2
6792 030110 010237 001242 MOV R2,@#STMP4
6793 030114 010137 001240 MOV R1,@#STMP3
6794 030120 010103 MOV R1,R3
6795 030122 062703 000004 ADD #4,R3
6796 030126 010337 001244 MOV R3,@#STMP5
6797 030132 010437 001250 MOV R4,@#STMP7
6798 030136 016137 000026 001252 MOV 26(R1),@#STMP10
6799 030144 010100 MOV R1,R0
6800 030146 062700 000004 ADD #4,R0
6801 030152 012703 000002 MOV #2,R3
6802 030156 022022 2$: CMP (R0)+,(R2)+
6803 030160 001006 BNE 10$ ;BR IF INCORRECT.
6804 030162 077303 SOB R3,2$
6805
6806 030164 026104 000026 CMP 26(R1),R4 ;IS THE FPS CORRECT?
6807 030170 001020 BNE 15$ ;BR IF INCORRECT.
6808 030172 000161 000040 3$: JMP 40(R1) ;RETURN.
6809
6810 ;THE RESULT WAS INCORRECT SO SEE IF THE ERROR WAS ANTICIPATED,
6811 030176 012702 027232 10$: MOV #LDCT,R2
6812 030202 010100 MOV R1,R0
6813 030204 062700 000014 ADD #14,R0
6814 030210 012703 000002 MOV #2,R3
6815 030214 022022 11$: CMP (R0)+,(R2)+
6816 030216 001003 BNE 13$
6817 030220 077303 SOB R3,11$
6818 030222 000161 000032 JMP 32(R1)
6819 030226 13$:
6820 ;ERROR NOT ANTICIPATED SO REPORT RESULT INCORRECT HERE,
6821 030226 104273 14$: ERROR 273 ;BAD RES
6822 030230 000760 BR 3$
6823
6824 ;THE FPS WAS INCORRECT, SEE IF FAILURE WAS ANTICIPATED,
6825 030232 026104 000030 15$: CMP 30(R1),R4
6826 030236 001002 BNE 16$
6827 030240 000161 000036 JMP 36(R1)
6828 ;FPS ERROR WAS NOT ANTICIPATED SO REPORT FAILURE HERE,

```

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

16s:  
 17s: ERROR 274 ;BAD FPS  
 BR 3s  
 LLCDONE:  
 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
 ;SEE IF THE USER HAS EXPRESSED  
 ;THE DESIRE TO CHANGE THE SOFTWARE  
 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
 ;THE USER TYPED CONTROL G?).  
 ;\*\*\*\*\*  
 ;\*TEST 60 LDEXP TEST  
 ;\*  
 ;\* THIS IS A TEST OF THE LDEXP INST  
 ;\* A SUBROUTINE IS USED TO SET UP  
 ;\* OPERANDS, EXECUTE THE LDEXP INST AND  
 ;\* CHECK THE RESULTS.  
 ;\*  
 ;\*\*\*\*\*  
 TST60: SCOPE  
 ; NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10  
 MMC1:  
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 JSR PC,LDXSUB ;GO EXECUTE THE INSTRUCTION,  
 1s: .WORD 12345,67012,34567,012345 ;AC0 OPERAND,  
 2s: .WORD 10 ;EXPONENT OPERAND,  
 3s: .WORD 42145,67012,34567,012345 ;EXPECTED RESULT,  
 4s: .WORD 2145,67012,34567,012345 ;ANTICIPATED ERRONEOUS RESULT,  
 5s: 47217 ;FPS BEFORE EXECUTION,  
 47200 ;FPS AFTER EXECUTION,  
 147200 ;ANTICIPATED ERRONEOUS FPS,  
 -1 ;EXPECTED FEC,  
 6s: ERROR 304 ;E12+E12+200 BAD  
 BR 7s ;ST 624  
 7s: ERROR 305 ;REPORT FPS INCORRECT,  
 ;ST 625 INTO 304  
 ;NON-ZERO RES NEG.  
 MMC2:  
 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 JSR PC,@LDXSUB ;EXPON=377  
 1s: .WORD 123456,70123,45670,123456 ;AC0 OPERAND,  
 2s: .WORD 177 ;EXPONENT OPERAND,  
 3s: .WORD 177656,70123,45670,123456 ;EXPECTED RESULT,  
 4s: .WORD 137656,70123,45670,123456 ;ANTICIPATED ERRONEOUS RESULT,  
 5s: 47207 ;FPS BEFORE EXECUTION,  
 47210 ;FPS AFTER EXECUTION,

```

6885 030376 147210 147210 ;ANTICIPATED ERRONEOUS FPS,
6886 030400 177777 -1 ;EXPECTED FEC,
6887 030402 104304 6s: ERROR 304 ;REPORT RESULT INCORRECT,
6888 030404 000401 BR 7s
6889 030406 104305 ERROR 305 ;REPORT FPS INCORRECT,
6890 030410 7s:
6891
6892 ;NON-ZERO RES, EXP=256=(56)REAL
6893 030410 MMC3:
6894 030410 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6895 030412 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
6896 030416 073261 057645 043323 1s: .WORD 73261,057645,43323,101760 ;AC0 OPERAND,
6897 030424 101760
6898 030426 000056 2s: .WORD 56 ;EXPONENT OPERAND,
6899 030430 053461 057645 043323 3s: .WORD 53461,057645,43323,101760 ;EXPECTED RESULT,
6900 030436 101760
6901 030440 177777 177777 4s: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT,
6902 030446 177777
6903 030450 047200 5s: 47200 ;FPS BEFORE EXECUTION,
6904 030452 047200 47200 ;FPS AFTER EXECUTION,
6905 030454 147200 147200 ;ANTICIPATED ERRONEOUS FPS,
6906 030456 177777 -1 ;EXPECTED FEC,
6907 030460 104301 6s: ERROR 301 ;REPORT RESULT INCORRECT,
6908 030462 000401 BR 7s
6909 030464 104305 ERROR 305 ;REPORT FPS INCORRECT,
6910 030466 7s:
6911
6912 ;EXP=27 (EXCESS 200)=-151 (OCT)
6913 030466 MMC4:
6914 030466 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6915 030470 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
6916 030474 012223 024252 062720 1s: .WORD 12223,24252,62720,21222 ;AC0 OPERAND,
6917 030502 021222
6918 030504 177627 2s: .WORD -151 ;EXPONENT OPERAND,
6919 030506 005623 024252 062720 3s: .WORD 5623,24252,62720,21222 ;EXPECTED RESULT,
6920 030514 021222
6921 030516 177777 177777 4s: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT,
6922 030524 177777
6923 030526 047200 5s: 47200 ;FPS BEFORE EXECUTION,
6924 030530 047200 47200 ;FPS AFTER EXECUTION,
6925 030532 147200 147200 ;ANTICIPATED ERRONEOUS FPS,
6926 030534 177777 -1 ;EXPECTED FEC,
6927 030536 104301 6s: ERROR 301 ;REPORT RESULT INCORRECT,
6928 030540 000401 BR 7s
6929 030542 104306 ERROR 306 ;(BUT EZBT) ST 544 TO 504 INTO 704 0 (BUT EXBT) ST 704 I
6930 030544 7s:
6931
6932 ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
6933 ; FIV=1
6934 030544 MMC5:
6935 030544 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
6936 030546 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
6937 030552 030131 032334 035363 1s: .WORD 30131,32334,35363,73031 ;AC0 OPERAND,
6938 030560 073031
6939 030562 177600 2s: .WORD -200 ;EXPONENT OPERAND,
6940 030564 000131 032334 035363 3s: .WORD 00131,32334,35363,73031 ;EXPECTED RESULT,
  
```

6941	030572	073031									
6942	030574	000000	000000	000000	4s:	.WORD	0,0,0,0				;ANTICIPATED ERRONEOUS RESULT.
6943	030602	000000									
6944	030604	042200			5s:		42200				;FPS BEFORE EXECUTION.
6945	030606	142204					142204				;FPS AFTER EXECUTION.
6946	030610	042202					42202				;ANTICIPATED ERRONEOUS FPS.
6947	030612	000012					12				;EXPECTED FEC.
6948	030614	104307			6s:	ERROR	307				;(BUT EXBT) ST 704 TO 64 INST 264
6949	030616	000401				BR	7s				
6950	030620	104310				ERROR	310				;(BUT FIU) ST 264 X
6951	030622				7s:						
6952											
6953											;EXP=0 (EXCESS 200)=-200 (OCT), NEG FRACT, FIU=1
6954	030622				MMC6:						
6955	030622	104413				LPERR					;SET UP THE LOOP ON ERROR ADDRESS.
6956	030624	004737	031616			JSR	PC,0#LDXSUB				;GO EXECUTE THE INSTRUCTION.
6957	030630	140414	024344	045464	1s:	.WORD	140414,24344,45464,74045				;AC0 OPERAND.
6958	030636	074045									
6959	030640	177600			2s:	.WORD	-200				;EXPONENT OPERAND.
6960	030642	100014	024344	045464	3s:	.WORD	100014,24344,45464,74045				;=-0 ;EXPECTED RESULT.
6961	030650	074045									
6962	030652	000000	000000	000000	4s:	.WORD	0,0,0,0				;ANTICIPATED ERRONEOUS RESULT.
6963	030660	000000									
6964	030662	042200			5s:		42200				;FPS BEFORE EXECUTION.
6965	030664	142214					142214				;FPS AFTER EXECUTION.
6966	030666	042214					42214				;ANTICIPATED ERRONEOUS FPS.
6967	030670	000012					12				;EXPECTED FEC.
6968	030672	104307			6s:	ERROR	307				;REPORT RESULT INCORRECT.
6969	030674	000401				BR	7s				
6970	030676	104310				ERROR	310				;REPORT FPS INCORRECT.
6971	030700				7s:						
6972											
6973											;EXP=0 (EXCESS 200)=-200 (OCT), POS FRAC, FIU=0
6974											
6975	030700				MMC7:						
6976	030700	104413				LPERR					;SET UP THE LOOP ON ERROR ADDRESS.
6977	030702	004737	031616			JSR	PC,0#LDXSUB				;GO EXECUTE THE INSTRUCTION.
6978	030706	051525	035455	005675	1s:	.WORD	51525,35455,5675,05152				;AC0 OPERAND.
6979	030714	005152									
6980	030716	177600			2s:	.WORD	-200				;EXPONENT OPERAND.
6981	030720	000000	000000	000000	3s:	.WORD	0,0,0,0				;EXPECTED RESULT.
6982	030726	000000									
6983	030730	000125	035455	005675	4s:	.WORD	00125,35455,5675,05152				;ANTICIPATED ERRONEOUS RESULT.
6984	030736	005152									
6985	030740	045200					45200				;FPS BEFORE EXECUTION.
6986	030742	045204					45204				;FPS AFTER EXECUTION.
6987	030744	145204					145204				;ANTICIPATED ERRONEOUS FPS.
6988	030746	177777					-1				;EXPECTED FEC.
6989	030750	104311			6s:	ERROR	311				;(BUT FIU) ST 264 X ;REPORT RESULT INCORRECT
6990	030752	000401				BR	7s				
6991	030754	104302				ERROR	302				;REPORT FPS INCORRECT.
6992	030756				7s:						
6993											
6994											;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
6995	030756				MMC8:						
6996	030756	104413				LPERR					;SET UP THE LOOP ON ERROR ADDRESS.



```

7053 ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
7054 031170 MMC11:
7055 031170 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7056 031172 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7057 031176 012131 014151 016171 1s: .WORD 12131,14151,16171,10111 ;AC0 OPERAND.
7058 031204 010111
7059 031206 001006 2s: .WORD 1006 ;EXPONENT OPERAND.
7060 031210 041531 014151 016171 3s: .WORD 41531,14151,16171,10111 ;EXPECTED RESULT.
7061 031216 010111
7062 031220 000000 000000 000000 4s: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7063 031226 000000
7064 031230 041200 5s: 41200 ;FPS BEFORE EXECUTION,
7065 031232 141202 141202 ;FPS AFTER EXECUTION,
7066 031234 041204 41204 ;ANTICIPATED ERRONEOUS FPS,
7067 031236 000010 10 ;EXPECTED FEC,
7068 031240 104314 6s: ERROR 314 ;(BUT FIV) ST 104
7069 031242 000401 BR 7s
7070 031244 104302 ERROR 302 ;REPORT FPS INCORRECT.
7071 031246
7072
7073 ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
7074 031246 MMC12:
7075 031246 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7076 031250 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7077 031254 027262 025242 023222 1s: .WORD 27262,25242,23222,21202 ;AC0 OPERAND.
7078 031262 021202
7079 031264 016115 2s: .WORD 16115 ;EXPONENT OPERAND.
7080 031266 000000 000000 000000 3s: .WORD 0,0,0,0 ;EXPECTED RESULT.
7081 031274 000000
7082 031276 063262 025242 023222 4s: .WORD 63262,25242,23222,21202 ;ANTICIPATED ERRONEOUS RESULT.
7083 031304 021202
7084 031306 046200 5s: 46200 ;FPS BEFORE EXECUTION,
7085 031310 046206 46206 ;FPS AFTER EXECUTION,
7086 031312 146202 146202 ;ANTICIPATED ERRONEOUS FPS,
7087 031314 177777 -1 ;EXPECTED FEC,
7088 031316 104315 6s: ERROR 315 ;(BUT FIV) ST 104
7089 031320 000401 BR 7s
7090 031322 104302 ERROR 302 ;REPORT FPS INCORRECT.
7091 031324
7092
7093 ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1
7094
7095 031324 MMC13:
7096 031324 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7097 031326 004737 031616 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7098 031332 030313 032333 034353 1s: .WORD 30313,32333,34353,36373 ;AC0 OPERAND.
7099 031340 036373
7100 031342 010611 2s: .WORD 10611 ;EXPONENT OPERAND.
7101 031344 002313 032333 034353 3s: .WORD 2313,32333,34353,36373 ;EXPECTED RESULT.
7102 031352 036373
7103 031354 000000 000000 000000 4s: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7104 031362 000000
7105 031364 041200 5s: 41200 ;FPS BEFORE EXECUTION,
7106 031366 141202 141202 ;FPS AFTER EXECUTION,
7107 031370 041204 41204 ;ANTICIPATED ERRONEOUS FPS,
7108 031372 000010 10 ;EXPECTED FEC.
  
```



```

7109 031374 104316 6$: ERROR 316 ;(BUT FIV) ST 144
7110 031376 000401 BR 7$
7111 031400 104302 ERROR 302 ;REPORT FPS INCORRECT.
7112 031402 7$:
7113
7114 ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
7115
7116 031402 MMC14:
7117 031402 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7118 031404 004737 031616 JSR PC,#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7119 031410 040414 042434 044454 1$: .WORD 40414,42434,44454,46474 ;AC0 OPERAND.
7120 031416 046474
7121 031420 016723 2$: .WORD 16723 ;EXPONENT OPERAND.
7122 031422 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
7123 031430 000000
7124 031432 024614 042434 044454 4$: .WORD 24614,42434,44454,46474 ;ANTICIPATED ERRONEOUS RESULT.
7125 031440 046474
7126 031442 046200 5$: 46200 ;FPS BEFORE EXECUTION.
7127 031444 046206 46206 ;FPS AFTER EXECUTION.
7128 031446 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
7129 031450 177777 =1 ;EXPECTED FEC.
7130 031452 104317 6$: ERROR 317 ;(BUT FIV) ST 144
7131 031454 000401 BR 7$
7132 031456 104302 ERROR 302 ;REPORT FPS INCORRECT.
7133 031460 7$:
7134
7135 ;EXP= 254 (OCT)= 454 (EXCESS 200) FIV=1
7136
7137 031460 MMC15:
7138 031460 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7139 031462 004737 031616 JSR PC,#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7140 031466 050515 052535 054555 1$: .WORD 50515,52535,54555,56575 ;AC0 OPERAND.
7141 031474 056575
7142 031476 000254 2$: .WORD 254 ;EXPONENT OPERAND.
7143 031500 013115 052535 054555 3$: .WORD 13115,52535,54555,56575 ;EXPECTED RESULT.
7144 031506 056575
7145 031510 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
7146 031516 000000
7147 031520 041200 5$: 41200 ;FPS BEFORE EXECUTION.
7148 031522 141202 141202 ;FPS AFTER EXECUTION.
7149 031524 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
7150 031526 000010 10 ;EXPECTED FEC.
7151 031530 104320 6$: ERROR 320 ;(BUT FIV) ST344
7152 031532 000401 BR 7$
7153 031534 104302 ERROR 302 ;REPORT FPS INCORRECT.
7154 031536 7$:
7155
7156 ;EXP= 313 (OCT)= 513(EXCESS 200) FIV=0
7157
7158 031536 MMC16:
7159 031536 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7160 031540 004737 031616 JSR PC,#LDXSUB ;GO EXECUTE THE INSTRUCTION,
7161 031544 060616 062636 064656 1$: .WORD 60616,62636,64656,66676 ;AC0 OPERAND.
7162 031552 066676
7163 031554 000313 2$: .WORD 313 ;EXPONENT OPERAND.
7164 031556 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
  
```

```

7165 031564 000000
7166 031566 022616 062636 064656 4s: .WORD 22616,62636,64656,66676 ;ANTICIPATED ERRONEOUS RESULT.
7167 031574 066676
7168 031576 046200 5s: 46200 ;FPS BEFORE EXECUTION.
7169 031600 046206 46206 ;FPS AFTER EXECUTION.
7170 031602 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
7171 031604 177777 -1 ;EXPECTED FEC.
7172 031606 104321 6s: ERROR 321 ;(BUT FIV) ST 344
7173 031610 000401 BR 7s
7174 031612 104302 ERROR 302 ;REPORT FPS INCORRECT.
7175 031614 7s:
7176 031614 000540 BR MMCDONE
  
```

```

7177
7178 ;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
7179 ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
7180 ;TO IT IS MADE THUS:
  
```

```

7181 ;
7182 ; JSR PC,@#LDXSUB
7183 ; ACARG: .WORD X,X,X,X ;AC OPERAND
7184 ; EXP: .WORD X ;EXPONENT
7185 ; RES: .WORD X,X,X,X ;EXPECTED RESULT
7186 ; ERRES: .WORD X,X,X,X ;ERROR RESULT
7187 ; FPSB: .WORD X ;FPS BEFORE EXECUTION
7188 ; FPSA: .WORD X ;FPS AFTER EXECUTION
7189 ; ERFPS: .WORD X ;ERROR FPS.
7190 ; FEC: .WORD X ;EXPECTED FEC
7191 ; ERR1: ERROR X ;DATA ERROR.
7192 ; BR CONT
7193 ; ERR2: ERROR X ;FPS ERROR.
7194 ; CONT: ;RETURN ADDRESS
  
```

```

7195 ;
7196 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
7197 ;THE LDEXP INSTRUCTION IS EXECUTED.
7198 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
7199 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDXSUB RETURNS CONTROL
7200 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDXSUB
7201 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDXSUB WILL RETURN
7202 ;TO THE ERROR CALL AT ERR2, OTHERWISE LDXSUB ITSELF
7203 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
7204 ;LDEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
7205 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN
7206 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDXSUB
7207 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
7208 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDXSUB WILL
7209 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
  
```

```

7210
7211 031616 012601 LDXSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
7212 031620 012700 000200 MOV #200,R0 ;LOAD THE AC0 OPERAND.
7213 031624 170100 LDFPS R0
7214 031626 010100 MOV R1,R0
7215 031630 172410 LDD (R0),AC0
7216 031632 012737 031654 001236 MOV #1s,@#TMP2
7217 031640 016100 000032 MOV 32(R1),R0 ;SET UP THE FPS.
7218 031644 170100 LDFPS R0
7219 031646 010100 MOV R1,R0
7220 031650 062700 000010 ADD #10,R0
  
```

```

7221
7222 031654 176410      1s:  LDEXP  (R0),AC0      ;TEST INSTRUCTION.
7223
7224 031656 170204      STFPS  R4              ;GET THE FPS.
7225 031660 170305      STST   R5              ;GET THE FEC.
7226 031662 012700 000200  MOV   #200,R0         ;GET THE RESULT,
7227 031666 170100      LDFPS  R0
7228 031670 012700 032106  MOV   #LDXT,R0
7229 031674 174010      STD    AC0,(R0)
7230 031676 010437 001250  MOV   R4,##$TMP7
7231 031702 016137 000034 001252  MOV   34(R1),##$TMP10
7232 031710 010537 001254      MOV   R5,##$TMP11
7233 031714 016137 000040 001256  MOV   40(R1),##$TMP12
7234 031722 010102      MOV   R1,R2
7235 031724 010237 001240      MOV   R2,##$TMP3
7236 031730 062702 000010      ADD   #10,R2
7237 031734 011237 001242      MOV   (R2),##$TMP4
7238 031740 062702 000002      ADD   #2,R2
7239 031744 010237 001244      MOV   R2,##$TMP5
7240 031750 012737 032106 001246  MOV   #LDXT,##$TMP6
7241 031756 012702 032106      MOV   #LDXT,R2      ;SEE IF THE RESULT WAS CORRECT,
7242 031762 010103      MOV   R1,R3
7243 031764 062703 000012      ADD   #12,R3
7244 031770 012700 000004      MOV   #4,R0
7245 031774 022223      2s:  CMP   (R2)+,(R3)+
7246 031776 001014      BNE   10s           ;BRANCH IF NOT CORRECT.
7247 032000 077003      SOB   R0,2s
7248 032002 020461 000034      CMP   R4,34(R1)     ;SEE IF THE FPS WAS CORRECT,
7249 032006 001026      BNE   15s           ;BRANCH IF NOT CORRECT,
7250 032010 005761 000034      TST   34(R1)
7251 032014 100003      BPL   3s
7252 032016 020561 000040      CMP   R5,40(R1)     ;SEE IF THE FEC WAS CORRECT,
7253 032022 001027      BNE   20s           ;BRANCH IF NOT CORRECT.
7254
7255 032024 000161 000050      3s:  JMP   50(R1)        ;RETURN,
7256
7257      ;THE RESULT WAS INCORRECT SO SEE IF THE FAILURE WAS ANTICIPATED,
7258 032030 012702 032106      10s: MOV   #LDXT,R2
7259 032034 010103      MOV   R1,R3
7260 032036 062703 000022      ADD   #22,R3
7261 032042 012700 000004      MOV   #4,R0
7262 032046 022223      11s: CMP   (R2)+,(R3)+
7263 032050 001003      BNE   12s
7264 032052 077003      SOB   R0,11s
7265 032054 000161 000042      JMP   42(R1)
7266
7267      ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE,
7268 032060      12s:
7269 032060 104301      13s: ERROR 301      ;BAD RES
7270 032062 000760      BR    3s
7271
7272      ;SEE IF THE FPS ERROR WAS ANTICIPATED,
7273 032064 026104 000036      15s: CMP   36(R1),R4
7274 032070 001002      BNE   16s
7275 032072 000161 000046      JMP   46(R1)
7276 032076      16s:
    
```

```

7277 ;THE FPS WAS NOT ANTICIPATED SO REPORT IT HERE.
7278 032076 104302 17$: ERROR 302 ;BAD FPS
7279 032100 000751 BR 3$ ;BUT EZBTY8
7280 ;ST 063
7281
7282 032102 20$:
7283 ;REPORT FEC INCORRECT.
7284 032102 104303 21$: ERROR 303 ;BAD FEC
7285 032104 000747 BR 3$
7286
7287 ;DATA BUFFER:
7288 032106 000000 000000 000000 LDXT: .WORD 0,0,0,0
7289 032114 000000
7290
7291 032116 MMCDONE:
7292 032116 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
7293 ;SEE IF THE USER HAS EXPRESSED
7294 ;THE DESIRE TO CHANGE THE SOFTWARE
7295 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7296 ;THE USER TYPED CONTROL G?).
7297
7298
7299
7300 ;*****
7301 ;*TEST 61 DESTINATION MODES, MODE 1 (FL=0), TEST
7302 ;*
7303 ;* THIS IS A TEST OF DESTINATION MODE 1 USING
7304 ;* THE STFPS INSTRUCTION
7305 ;*
7306 ;*****
7307 032120 000004 TST61: SCOPE
7308
7309
7310 NNC1:
7311 032122 104413 LPPER ;SET UP THE LOOP ON ERROR ADDRESS.
7312 032124 012700 032222 MOV #NNCTB0,R0 ;SET UP THE DATA BUFFER.
7313 032130 012701 000006 MOV #6,R1
7314 032134 012720 177777 1$: MOV #-1,(R0)+
7315 032140 077103 SOB R1,1$
7316 032142 012700 102345 MOV #102345,R0
7317 032146 012737 032170 001236 MOV #NNC2,@$TMP2
7318 032154 012737 032322 000004 MOV #NNC25,@$ERRVECT ;SET UP FOR TRAPS TO 4,
7319 032162 170100 LDFPS R0 ;SET UP FPS.
7320 032164 012700 032226 MOV #NNCTB1,R0
7321
7322 NNC2: STFPS (R0) ;TEST INSTRUCTION,
7323 032172 020027 032226 CMP R0,#NNCTB1 ;IS R0 CORRECT?
7324 032176 001017 BNE NNC10 ;BRANCH IF NOT CORRECT,
7325 032200 023727 032226 102345 CMP @#NNCTB1,#102345 ;IS RESULT CORRECT?
7326 032206 001023 BNE NNC15 ;BRANCH IF NOT CORRECT,
7327 032210 023727 032230 177777 CMP @#NNCTB1+2,#-1 ;IS THE RESULT CORRECT?
7328 032216 001030 BNE NNC20 ;BRANCH IF NOT CORRECT.
7329 032220 000453 BR NNCDONE
7330
7331 ;TEST DATA BUFFER:
7332 032222 177777 177777 NNCTB0: .WORD -1,-1
  
```

```

7333 032226 177777 177777 177777 NNCTB1: .WORD -1,-1,-1,-1
7334 032234 177777
7335
7336 ;REPORT R0 INCORRECT.
7337 032236 010037 001242 NNC10: MOV R0,@$TMP4
7338 032242 012737 032226 001240 MOV #NNCTB1,@$TMP3
7339 032250 1s:
7340 032250 104377 ERROR 377
7341 032252 000001 .WORD 1
7342 ;R0 BAD (BUT
7343 032254 000435 BR NNCDONE ; FDST)X
7344
7345 ;REPORT RESULT INCORRECT.
7346 032256 012737 102345 001240 NNC15: MOV #102345,@$TMP3 ; ST 634
7347 032264 013737 032226 001242 MOV @NNCTB1,@$TMP4
7348 032272 1s:
7349 032272 104377 ERROR 377
7350 032274 000002 .WORD 2
7351 ;BAD DATA
7352 032276 000424 BR NNCDONE
7353
7354
7355 ;REPORT RESULT INCORRECT.
7356 032300 012737 177777 001240 NNC20: MOV #-1,@$TMP3
7357 032306 013737 032230 001242 MOV @NNCTB1+2,@$TMP4
7358 032314 1s:
7359 032314 104377 ERROR 377
7360 032316 000003 .WORD 3
7361 ;(BUT GR7,FL)
7362 032320 000413 BR NNCDONE ;ST 357 TO 416
7363 ;INTO 417
7364
7365 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7366 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7367 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7368 032322 011604 NNC25: MOV (SP),R4
7369 032324 020427 032172 CMP R4,#NNC2+2
7370 032330 001402 BEQ 1s
7371 032332 000137 042620 JMP @CPSPUR
7372
7373 032336 011637 001236 1s: MOV (SP),@$TMP2
7374 032342 022626 CMP (SP)+,(SP)+
7375 032344 2s:
7376 032344 104377 ERROR 377
7377 032346 000004 .WORD 4
7378 ;(BUT FDST)+ ST634
7379
7380 NNCDONE;
7381 032350 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
7382 104412 ;SEE IF THE USER HAS EXPRESSED
7383 ;THE DESIRE TO CHANGE THE SOFTWARE
7384 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7385 ;THE USER TYPED CONTROL G?).
7386
7387
7388 ;*****

```

```

7389 ;*TEST 62 DESTINATION MODES, MODE 2 (FL=0), TEST
7390 ;*
7391 ;* THIS IS A TEST OF DESTINATION MODE 2 USING
7392 ;* THE STFPS INSTRUCTION
7393 ;*
7394 ;*****
7395 032352 000004 TST62; SCOPE
7396
7397
7398 032354 OOC1:
7399 032354 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7400 032356 012700 032454 MOV #0OCTB0,R0 ;SET UP THE DATA BUFFER,
7401 032362 012701 000006 MOV #6,R1
7402 032366 012720 177777 1$: MOV #-1,(R0)+
7403 032372 077103 SOB R1,1$
7404 032374 012700 105412 MOV #105412,R0
7405 032400 012737 032422 001236 MOV #OOC2,@#$TMP2
7406 032406 012737 032554 000004 MOV #OOC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
7407 032414 170100 LDFPS R0 ;SET UP FPS.
7408 032416 012700 032460 MOV #0OCTB1,R0
7409
7410 032422 170220 OOC2: STFPS (R0)+ ;TEST INSTRUCTION.
7411 032424 020027 032462 CMP R0,#0OCTB1+2 ;IS R0 CORRECT?
7412 032430 001017 BNE OOC10 ;BRANCH IF NOT CORRECT.
7413 032432 023727 032460 105412 CMP @#0OCTB1,#105412 ;IS THE RESULT CORRECT?
7414 032440 001023 BNE OOC15 ;BRANCH IF NOT CORRECT.
7415 032442 023727 032462 177777 CMP @#0OCTB1+2,#-1 ;IS THE RESULT CORRECT?
7416 032450 001030 BNE OOC20 ;BRANCH IF NOT CORRECT.
7417 032452 000453 BR OOCDONE
7418
7419 ;TEST DATA BUFFER:
7420 032454 177777 177777 OOCB0: ,WORD -1,-1
7421 032460 177777 177777 177777 OOCB1: ,WORD -1,-1,-1,-1
7422 032466 177777
7423
7424 ;REPORT R0 INCORRECT.
7425 032470 010037 001242 OOC10: MOV R0,@#$TMP4
7426 032474 012737 032462 001240 MOV #0OCTB1+2,@#$TMP3
7427 032502 1$:
7428 032502 104377 ERROR 377
7429 032504 000005 ,WORD 5
7430 ;R0 BAD (BUT
7431 032506 000435 BR OOCDONE ; FDST)X
7432
7433 ;REPORT RESULT INCORRECT.
7434 032510 012737 105412 001240 OOC15: MOV #105412,@#$TMP3 ; ST 634
7435 032516 013737 032460 001242 MOV @#0OCTB1,@#$TMP4
7436 032524 1$:
7437 032524 104377 ERROR 377
7438 032526 000006 ,WORD 6
7439 ;BAD DATA
7440 032530 000424 BR OOCDONE
7441
7442
7443 ;REPORT RESULT INCORRECT.
7444 032532 012737 177777 001240 OOC20: MOV #-1,@#$TMP3
  
```

```

7445 032540 013737 032462 001242      MOV    @#0OCTB1+2,@#$TMP4
7446 032546      1$:
7447 032546 104377      ERROR  377
7448 032550 000007      .WORD  7
7449      ;(BUT GR7,FL)
7450 032552 000413      BR     OOCDONE      ;ST 357 TO 416
7451      ;INTO 417
7452
7453      ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7454      ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7455      ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7456 032554 011604      OOC25; MOV    (SP),R4
7457 032556 020427 032424      CMP    R4,#OOC2+2
7458 032562 001402      BEQ    1$
7459 032564 000137 042620      JMP    @#CPSPUR
7460
7461 032570 011637 001236      1$:    MOV    (SP),@#$TMP2
7462 032574 022626      CMP    (SP)+,(SP)+
7463 032576      2$:
7464 032576 104377      ERROR  377
7465 032600 000010      .WORD  10
7466      ;(BUT FDST)+ ST634
7467
7468 032602      OOCDONE:
7469 032602 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
7470      ;SEE IF THE USER HAS EXPRESSED
7471      ;THE DESIRE TO CHANGE THE SOFTWARE
7472      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7473      ;THE USER TYPED CONTROL G?).
7474
7475
7476
7477      ;*****
7478      ;*TEST 63      DESTINATION MODES, MODE 4 (FL=0), TEST
7479      ;*
7480      ;* THIS IS A TEST OF DESTINATION MODE 4 USING
7481      ;* THE STFPS INSTRUCTION
7482      ;*
7483      ;*****
7484 032604 000004      TST63; SCOPE
7485
7486 032606      PPC1:
7487 032606 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
7488 032610 012700 032706      MOV    #PPCTB0,R0      ;SET UP THE DATA BUFFER,
7489 032614 012701 000006      MOV    #6,R1
7490 032620 012720 177777      1$:    MOV    #-1,(R0)+
7491 032624 077103      SOB    R1,1$
7492 032626 012700 105555      MOV    #105555,R0
7493 032632 012737 032654 001236      MOV    #PPC2,@#$TMP2
7494 032640 012737 033006 000004      MOV    #PPC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4,
7495 032646 170100      LDFPS  R0      ;SET UP FPS,
7496 032650 012700 032714      MOV    #PPCTB1+2,R0
7497
7498 032654 170240      PPC2:    STFPS  -(R0)      ;TEST INSTRUCTION.
7499 032656 020027 032712      CMP    R0,#PPCTB1      ;IS R0 CORRECT?
7500 032662 001017      BNE    PPC10          ;BRANCH IF NOT CORRECT.

```

7501 032664 023727 032712 105555 CMP @#PPCTB1,#105555 ;IS THE RESULT CORRECT?  
 7502 032672 001023 BNE PPC15 ;BRANCH IF NOT CORRECT.  
 7503 032674 023727 032714 177777 CMP @#PPCTB1+2,#-1 ;IS THE RESULT CORRECT?  
 7504 032702 001030 BNE PPC20 ;BRANCH IF NOT CORRECT.  
 7505 032704 000453 BR PPCDONE

7506  
 7507 ;TEST DATA BUFFER:  
 7508 032706 177777 177777 PPCTB0: .WORD -1,-1  
 7509 032712 177777 177777 177777 PPCTB1: .WORD -1,-1,-1,-1  
 7510 032720 177777

7511  
 7512 ;REPORT R0 INCORRECT.  
 7513 032722 010037 001242 PPC10: MOV R0,@#STMP4  
 7514 032726 012737 032712 001240 MOV #PPCTB1,@#STMP3  
 7515 032734 1\$:  
 7516 032734 104377 ERROR 377  
 7517 032736 000011 .WORD 11

7518  
 7519 032740 000435 BR PPCDONE ;R0 BAD (BUT  
 7520 ; FDST)X

7521 ;REPORT RESULT INCORRECT.  
 7522 032742 012737 105555 001240 PPC15: MOV #105555,@#STMP3 ; ST 634  
 7523 032750 013737 032712 001242 MOV @#PPCTB1,@#STMP4  
 7524 032756 1\$:  
 7525 032756 104377 ERROR 377  
 7526 032760 000012 .WORD 12

7527 ;BAD DATA  
 7528 032762 000424 BR PPCDONE

7529  
 7530  
 7531 ;REPORT RESULT INCORRECT.  
 7532 032764 012737 177777 001240 PPC20: MOV #-1,@#STMP3  
 7533 032772 013737 032714 001242 MOV @#PPCTB1+2,@#STMP4  
 7534 033000 1\$:  
 7535 033000 104377 ERROR 377  
 7536 033002 000013 .WORD 13

7537 ;(BUT GR7,FL)  
 7538 033004 000413 BR PPCDONE ;ST 357 TO 416  
 7539 ;INTO 417  
 7540

7541 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED  
 7542 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO  
 7543 ;TO THE SPURIOUS TRAP TO 4 HANDLER.

7544 033006 011604 PPC25: MOV (SP),R4  
 7545 033010 020427 032656 CMP R4,#PPC2+2  
 7546 033014 001402 BEQ 1\$  
 7547 033016 000137 042620 JMP @#CPSPUR

7548  
 7549 033022 011637 001236 1\$: MOV (SP),@#STMP2  
 7550 033026 022626 CMP (SP)+,(SP)+  
 7551 033030 2\$:  
 7552 033030 104377 ERROR 377  
 7553 033032 000014 .WORD 14

7554 ;(BUT FDST)+ ST634  
 7555

7556 033034 PPCDONE:



7557 033034 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
 7558 ;SEE IF THE USER HAS EXPRESSED  
 7559 ;THE DESIRE TO CHANGE THE SOFTWARE  
 7560 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
 7561 ;THE USER TYPED CONTROL G?).  
 7562  
 7563  
 7564

7565 ;\*\*\*\*\*  
 7566 ;\*TEST 64 DESTINATION MODES, MODE 3 (FL=0), TEST  
 7567 ;\*  
 7568 ;\* THIS IS A TEST OF DESTINATION MODE 3 USING  
 7569 ;\* THE STFPS INSTRUCTION  
 7570 ;\*  
 7571 ;\*\*\*\*\*

7572 033036 000004 TST64: SCOPE  
 7573  
 7574 033040 QQC1:  
 7575 033040 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 7576 033042 012700 033144 MOV #QQCTB0,R0 ;SET UP THE DATA BUFFER,  
 7577 033046 012701 000010 MOV #10,R1  
 7578 033052 012720 177777 1s: MOV #-1,(R0)+  
 7579 033056 077103 SOB R1,1s  
 7580 033060 012700 106653 MOV #106653,R0  
 7581 033064 012737 033112 001236 MOV #QQC2,@#TMP2  
 7582 033072 012737 033250 000004 MOV #QQC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4,  
 7583 033100 170100 LDFPS R0 ;SET UP FPS,  
 7584 033102 012700 033160 MOV #QQCTB2,R0  
 7585 033106 012710 033150 MOV #QQCTB1,(R0)  
 7586

7587 033112 170230 QQC2: STFPS @(R0)+ ;TEST INSTRUCTION,  
 7588 033114 020027 033162 CMP R0,#QQCTB2+2 ;IS R0 CORRECT?  
 7589 033120 001021 BNE QQC10 ;BRANCH IF NOT CORRECT,  
 7590 033122 023727 033150 106653 CMP @#QQCTB1,#106653 ;IS THE RESULT CORRECT?  
 7591 033130 001025 BNE QQC15 ;BRANCH IF NOT CORRECT,  
 7592 033132 023727 033160 033150 CMP @#QQCTB2,#QQCTB1 ;IS THE RESULT CORRECT?  
 7593 033140 001032 BNE QQC20 ;BRANCH IF NOT CORRECT,  
 7594 033142 000455 BR QQCDONE  
 7595

7596 ;TEST DATA BUFFER:  
 7597 033144 177777 177777 QQC10: .WORD -1,-1  
 7598 033150 177777 177777 177777 QQC11: .WORD -1,-1,-1,-1  
 7599 033156 177777  
 7600 033160 177777 177777 QQC12: .WORD -1,-1  
 7601  
 7602

7603 033164 010037 001242 ;REPORT R0 INCORRECT.  
 7604 033170 012737 033162 001240 QQC10: MOV R0,@#TMP4  
 7605 033176 1s: MOV #QQCTB2+2,@#TMP3  
 7606 033176 104377 ERROR 377  
 7607 033200 000015 .WORD 15  
 7608 ;R0 BAD (BUT  
 7609 033202 000435 BR QQCDONE ; FDST)X  
 7610

7611 ;REPORT RESULT INCORRECT,  
 7612 033204 012737 106653 001240 QQC15: MOV #106653,@#TMP3 ; ST 634

```

7613 033212 013737 033150 001242      MOV      @#QQCTB1,@#STMP4
7614 033220                               1$:
7615 033220 104377      ERROR      377
7616 033222 000016      .WORD      16
7617                               ;BAD DATA
7618 033224 000424      BR        QQCDONE
7619
7620
7621                               ;REPORT RESULT INCORRECT.
7622 033226 012737 033160 001240 QQC20:  MOV      #QQCTB2,@#STMP3           ;(BUT FDST)
7623 033234 013737 033152 001242      MOV      @#QQCTB1+2,@#STMP4
7624 033242                               1$:
7625 033242 104377      ERROR      377
7626 033244 000017      .WORD      17
7627 033246 000413      BR        QQCDONE
7628
7629
7630                               ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7631                               ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7632                               ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7633 033250 011604 QQC25:  MOV      (SP),R4
7634 033252 020427 033114      CMP      R4,#QQC2+2
7635 033256 001402      BEQ      1$
7636 033260 000137 042620      JMP      @#CPSPUR
7637
7638 033264 011637 001236      1$:      MOV      (SP),@#STMP2
7639 033270 022626      CMP      (SP)+,(SP)+
7640 033272                               2$:
7641 033272 104377      ERROR      377
7642 033274 000020      .WORD      20
7643                               ;(BUT FDST)+ ST634
7644
7645                               QQCDONE:
7646 033276 104412      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
7647                               ;SEE IF THE USER HAS EXPRESSED
7648                               ;THE DESIRE TO CHANGE THE SOFTWARE
7649                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7650                               ;THE USER TYPED CONTROL G?).
7651
7652
7653
7654                               ;*****
7655                               ;*TEST 65      DESTINATION MODES, MODE 5 (FL=0), TEST
7656                               ;*
7657                               ;* THIS IS A TEST OF DESTINATION MODE 5 USING
7658                               ;* THE STFPS INSTRUCTION
7659                               ;*
7660                               ;*****
7661 033300 000004      TST65:  SCOPE
7662
7663
7664                               RRC1:
7665 033302 104413      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
7666 033304 012700 033410      MOV      #RRTB0,R0      ;SET UP THE DATA BUFFER.
7667 033310 012701 000006      MOV      #6,R1
7668 033314 012720 177777      1$:      MOV      #-1,(R0)+
  
```

```

7669 033320 077103 SOB R1,18
7670 033322 012700 004301 MOV #004301,R0
7671 033326 012737 033356 001236 MOV #RRC2,@#TMP2
7672 033334 012737 033514 000004 MOV #RRC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4,
7673 033342 170100 LDFPS R0 ;SET UP FPS.
7674 033344 012700 033426 MOV #RRCTB2+2,R0
7675 033350 012760 033414 177776 MOV #RRCTB1,-2(R0)
7676
7677 033356 170250 RRC2: STFPS @-(R0) ;TEST INSTRUCTION.
7678 033360 020027 033424 CMP R0,#RRCTB2 ;IS R0 CORRECT?
7679 033364 001021 BNE RRC10 ;BRANCH IF NOT CORRECT.
7680 033366 023727 033414 004301 CMP @#RRCTB1,#004301 ;IS THE RESULT CORRECT?
7681 033374 001025 BNE RRC15 ;BRANCH IF NOT CORRECT.
7682 033376 023727 033424 033414 CMP @#RRCTB2,#RRCTB1 ;IS THE RESULT CORRECT?
7683 033404 001032 BNE RRC20 ;BRANCH IF NOT CORRECT.
7684 033406 000455 BR RRCDONE
7685
7686 ;TEST DATA BUFFER:
7687 033410 177777 177777 RRC10: MOV R0,#RRCTB0 ;WORD -1,-1
7688 033414 177777 177777 177777 RRC15: MOV R0,#RRCTB1 ;WORD -1,-1,-1,-1
7689 033422 177777
7690 033424 177777 177777 RRC20: MOV R0,#RRCTB2 ;WORD -1,-1
7691
7692 ;REPORT R0 INCORRECT.
7693 033430 010037 001242 RRC10: MOV R0,@#TMP4
7694 033434 012737 033424 001240 MOV #RRCTB2,@#TMP3
7695 033442
7696 033442 104377
7697 033444 000021
7698
7699 033446 000435 BR RRCDONE ;R0 BAD (BUT
; FDST)X
7700
7701 ;REPORT RESULT INCORRECT.
7702 033450 012737 004301 001240 RRC15: MOV #004301,@#TMP3 ; ST 634
7703 033456 013737 033414 001242 MOV @#RRCTB1,@#TMP4
7704 033464
7705 033464 104377
7706 033466 000022
7707
7708 033470 000424 BR RRCDONE ;BAD DATA
7709
7710
7711 ;REPORT RESULT INCORRECT.
7712 033472 012737 033424 001240 RRC20: MOV #RRCTB2,@#TMP3 ;BUT FDST)
7713 033500 013737 033416 001242 MOV @#RRCTB1+2,@#TMP4
7714 033506
7715 033506 104377
7716 033510 000023
7717
7718 033512 000413 BR RRCDONE ;(BUT GR7,FL)
;ST 357 TO 416
;INTO 417
7719
7720
7721 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
;TO THE SPURIOUS TRAP TO 4 HANDLER.
7722
7723
7724 033514 011604 RRC25: MOV (SP),R4
    
```

7725 033516 020427 033360  
 7726 033522 001402  
 7727 033524 000137 042620  
 7728  
 7729 033530 011637 001236  
 7730 033534 022626  
 7731 033536  
 7732 033536 104377  
 7733 033540 000024  
 7734

CMP R4,#RRC2+2  
 BEQ 1\$  
 JMP @#CPSPUR  
 1\$: MOV (SP),@#STMP2  
 CMP (SP)+,(SP)+  
 2\$:  
 ERROR 377  
 .WORD 24

;(BUT FDST)+ ST634

7735  
 7736 033542  
 7737 033542 104412  
 7738  
 7739  
 7740  
 7741  
 7742  
 7743  
 7744

RRCDONE:  
 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?).

7745  
 7746  
 7747  
 7748  
 7749  
 7750

;;\*\*\*\*\*

\*TEST 66 DESTINATION MODES, MODE 6 (FL=0), TEST

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

;;\*\*\*\*\*

7751 033544 000004  
 7752

TST66: SCOPE

7753  
 7754 033546  
 7755 033546 104413  
 7756 033550 012700 033660  
 7757 033554 012701 000006  
 7758 033560 012720 177777  
 7759 033564 077103  
 7760 033566 012700 102514  
 7761 033572 012737 033616 001236  
 7762 033600 012737 033760 000004  
 7763 033606 170100  
 7764 033610 005001  
 7765 033612 012700 026463  
 7766

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,@#STMP2  
 MOV #SSC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4,  
 LDFPS R0 ;SET UP FPS,  
 CLR R1  
 MOV #SSCTB1-5201,R0

7767 033616 170260 005201  
 7768 033622 020127 000000  
 7769 033626 001070  
 7770 033630 020027 026463  
 7771 033634 001017  
 7772 033636 023727 033664 102514  
 7773 033644 001023  
 7774 033646 023727 033666 177777  
 7775 033654 001030  
 7776 033656 000456

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

7777  
 7778  
 7779 033660 177777 177777  
 7780 033664 177777 177777 177777

;TEST DATA BUFFER:

SSCTB0: .WORD -1,-1  
 SSCTB1: .WORD -1,-1,-1,-1

```

7781 033672 177777
7782
7783 ;REPORT R0 INCORRECT.
7784 033674 010037 001242 SSC10: MOV R0,@@$TMP4
7785 033700 012737 026463 001240 MOV #SSCTB1-5201,@@$TMP3
7786 033706 1s:
7787 033706 104377 ERROR 377
7788 033710 000025 .WORD 25
7789 ;R0 BAD
7790 033712 000440 BR SSCDONE
7791
7792 ;REPORT RESULT INCORRECT.
7793 033714 012737 102534 001240 SSC15: MOV #102534,@@$TMP3
7794 033722 013737 033664 001242 MOV @SSCTB1,@@$TMP4
7795 033730 1s:
7796 033730 104377 ERROR 377
7797 033732 000026 .WORD 26
7798 ;BAD DATA
7799 033734 000427 BR SSCDONE
7800
7801
7802 ;REPORT RESULT INCORRECT.
7803 033736 012737 177777 001240 SSC20: MOV #-1,@@$TMP3
7804 033744 013737 033666 001242 MOV @SSCTB1+2,@@$TMP4
7805 033752 1s:
7806 033752 104377 ERROR 377
7807 033754 000027 .WORD 27
7808 ;(BUT GR7,FL)
7809 033756 000416 BR SSCDONE ;ST 357 TO 416
7810 ;INTO 417
7811
7812 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7813 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7814 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7815 033760 011604 SSC25: MOV (SP),R4
7816 033762 020427 033620 CMP R4,#SSC2+2
7817 033766 001402 BEQ 1s
7818 033770 000137 042620 JMP @CPSPUR
7819
7820 033774 011637 001236 1s: MOV (SP),@@$TMP2
7821 034000 022626 CMP (SP)+,(SP)+
7822 034002 2s:
7823 034002 104377 ERROR 377
7824 034004 000030 .WORD 30
7825 ;(BUT FDST)+ ST634
7826 034006 000402 BR SSCDONE
7827
7828 ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
7829 034010 SSC30:
7830 034010 1s:
7831 034010 104377 ERROR 377
7832 034012 000031 .WORD 31
7833 ;PC NOT
7834 ;INCREMENTED
7835 ;BY 2
7836
    
```

```

7837 034014          SSCDONE:
7838 034014 104412          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
7839                                     ;SEE IF THE USER HAS EXPRESSED
7840                                     ;THE DESIRE TO CHANGE THE SOFTWARE
7841                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7842                                     ;THE USER TYPED CONTROL G?).
7843
7844
7845 ;*****
7846 ;*TEST 67          DESTINATION MODES, MODE 7 (FL=0), TEST
7847 ;*
7848 ;* THIS IS A TEST OF DESTINATION MODE 7 USING
7849 ;* THE STFPS INSTRUCTION
7850 ;*
7851 ;*****
7852 034016 000004          TST67:  SCOPE
7853
7854 034020          TTC1:
7855 034020 104413          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS,
7856 034022 012700 034140          MOV          #TTCTB0,R0          ;SET UP THE DATA BUFFER,
7857 034026 012701 000010          MOV          #10,R1
7858 034032 012720 177777          1$:          MOV          #-1,(R0)+
7859 034036 077103          SOB          R1,1$
7860 034040 012700 103747          MOV          #103747,R0
7861 034044 012737 034076 001236          MOV          #TTC2,@$TMP2
7862 034052 012737 034244 000004          MOV          #TTC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4,
7863 034060 170100          LDFPS          R0          ;SET UP FPS,
7864 034062 005001          CLR          R1
7865 034064 012700 026753          MOV          #TTCTB2-5201,R0
7866 034070 012760 034144 005201          MOV          #TTCTB1,5201(R0)
7867
7868 034076 170270 005201          TTC2:  STFPS          05201(R0)          ;TEST INSTRUCTION,
7869 034102 022701 000000          CMP          #0,R1          ;WAS PC CORRECT AFTER EXECUTION?
7870 034106 001072          BNE          TTC30          ;BRANCH IF NOT CORRECT,
7871 034110 020027 026753          CMP          R0,#TTCTB2-5201 ;IS R0 CORRECT?
7872 034114 001021          BNE          TTC10          ;BRANCH IF NOT CORRECT,
7873 034116 023727 034144 103747          CMP          @#TTCTB1,#103747 ;IS THE RESULT CORRECT?
7874 034124 001025          BNE          TTC15          ;BRANCH IF NOT CORRECT,
7875 034126 023727 034146 177777          CMP          @#TTCTB1+2,#-1 ;IS THE RESULT CORRECT?
7876 034134 001032          BNE          TTC20          ;BRANCH IF NOT CORRECT,
7877 034136 000460          BR          TTCDONE
7878
7879 ;TEST DATA BUFFER:
7880 034140 177777 177777          TTCTB0: .WORD  -1,-1
7881 034144 177777 177777 177777          TTCTB1: .WORD  -1,-1,-1,-1
7882 034152 177777
7883 034154 177777 177777          TTCTB2: .WORD  -1,-1
7884
7885 ;REPORT R0 INCORRECT.
7886 034160 010037 001242          TTC10:  MOV          R0,@$TMP4
7887 034164 012737 026753 001240          MOV          #TTCTB2-5201,@$TMP3
7888 034172
7889 034172 104377          1$:          ERROR          377
7890 034174 000032          .WORD          32
7891                                     ;R0 BAD
7892 034176 000440          BR          TTCDONE

```

```

7893
7894
7895 ;REPORT RESULT INCORRECT.
7896 034200 012737 103747 001240 TTC15: MOV #103747,@#STMP3
7897 034206 013737 034144 001242 MOV @#TTCTB1,@#STMP4
7898 034214 1s:
7899 034214 104377 ERROR 377
7900 034216 000033 .WORD 33
7901 ;BAD DATA
7902 034220 000427 BR TTCDONE
7903
7904
7905 ;REPORT RESULT INCORRECT.
7906 034222 012737 177777 001240 TTC20: MOV #-1,@#STMP3
7907 034230 013737 034146 001242 MOV @#TTCTB1+2,@#STMP4
7908 034236 1s:
7909 034236 104377 ERROR 377
7910 034240 000034 .WORD 34
7911 ;(BUT GR7,FL)
7912 034242 000416 BR TTCDONE ;ST 357 TO 416
7913 ;INTO 417
7914
7915 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
7916 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
7917 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
7918 034244 011604 TTC25: MOV (SP),R4
7919 034246 020427 034100 CMP R4,#TTC2+2
7920 034252 001402 BEQ 1s
7921 034254 000137 042620 JMP @#CPSPUR
7922 034260 011637 001236 1s: MOV (SP),@#STMP2
7923 034264 022626 CMP (SP)+,(SP)+
7924 034266 2s:
7925 034266 104377 ERROR 377
7926 034270 000035 .WORD 35
7927 ;(BUT FSDT)+ ST634
7928 034272 000402 BR TTCDONE
7929
7930 ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
7931 034274 TTC30:
7932 034274 1s:
7933 034274 104377 ERROR 377
7934 034276 000036 .WORD 36
7935 ;PC NOT
7936 ;INCREMENTED
7937 034300 TTCDONE:
7938 034300 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
7939 ;SEE IF THE USER HAS EXPRESSED
7940 ;THE DESIRE TO CHANGE THE SOFTWARE
7941 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7942 ;THE USER TYPED CONTROL G?).
7943
7944 ;*****
7945 ;*TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST
7946 ;*
7947 ;* THIS IS A TEST OF DESTINATION MODE
7948 ;* 2 USING STCOL WITH REGISTER 0
  
```

```

7949
7950 ;*
7951 034302 000004 ;*****
TST70: SCOPE
7952 034304 UUC1:
7953 034304 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7954 034306 012700 000300 MOV #300,R0 ;SET UP FPS,
7955 034312 170100 LDFPS R0
7956 034314 012700 034364 MOV #UUCTP1,R0 ;SET UP THE AC0 OPERAND,
7957 034320 172410 LDD (R0),AC0
7958 034322 012737 034334 001236 MOV #UUC2,@$TMP2
7959 034330 012700 034376 MOV #UUCBF0,R0
7960
7961 034334 175420 UUC2: STCDL AC0,(R0)+ ;TEST INSTRUCTION,
7962
7963 034336 020027 034402 CMP R0,#UUCBF0+4 ;IS R0 CORRECT?
7964 034342 001420 BEQ UUCDONE ;BRANCH IF CORRECT,
7965
7966 ;REPORT R0 INCORRECT,
7967 034344 010037 001242 UUC3: MOV R0,@$TMP4
7968 034350 012737 034402 001240 MOV #UUCBF0+4,@$TMP3
7969 034356
7970 034356 104377
7971 034360 000037
7972
7973 034362 000410
7974 ;TEST DATA BUFFER:
7975 034364 000000 000000 000000 UUCTP1: .WORD 0,0,0,0
7976 034372 000000
7977 034374 177777
7978 034376 177777 177777 177777 UUCBF0: .WORD -1,-1,-1
7979
7980 034404
7981 034404 104412 UUCDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
7982 ;SEE IF THE USER HAS EXPRESSED
7983 ;THE DESIRE TO CHANGE THE SOFTWARE
7984 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
7985 ;THE USER TYPED CONTROL G?).
7986
7987 ;*****
7988 ;*TEST 71 DESTINATION MODES, MODE 4 (FL=1), TEST
7989 ;*
7990 ;* THIS IS A TEST OF DESTINATION MODE
7991 ;* 4 USING STCDL WITH REGISTER 0
7992 ;*
7993 ;*****
7994 034406 000004 TST71: SCOPE
7995
7996 034410 VVC1:
7997 034410 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
7998 034412 012700 000300 MOV #300,R0 ;SET UP FPS,
7999 034416 170100 LDFPS R0
8000 034420 012700 034470 MOV #VVCTP1,R0 ;SET UP THE AC0 OPERAND,
8001 034424 172410 LDD (R0),AC0
8002 034426 012737 034440 001236 MOV #VVC2,@$TMP2
8003 034434 012700 034506 MOV #VVCBF0+4,R0
8004

```



```

8005 034440 175440 VVC2: STCDL AC0,-(R0) ;TEST INSTRUCTION,
8006
8007 034442 020027 034502 CMP R0,#VVCBF0 ;IS R0 CORRECT?
8008 034446 001420 BEQ VVCDONE
8009
8010 ;REPORT R0 INCORRECT.
8011 034450 010037 001242 VVC3: MOV R0,#$TMP4
8012 034454 012737 034502 001240 MOV #VVCBF0,#$TMP3
8013 034462 16:
8014 034462 104377 ERROR 377
8015 034464 000040 .WORD 40
8016 ;R0 NOT DECR BY 4
8017 034466 000410 BR VVCDONE
8018 ;TEST DATA BUFFER;
8019 034470 000000 000000 000000 VVCTP1: .WORD 0,0,0,0
8020 034476 000000
8021 034500 177777 -1
8022 034502 177777 177777 177777 VVCBF0: .WORD -1,-1,-1
8023
8024 034510 VVCDONE:
8025 034510 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
8026 ;SEE IF THE USER HAS EXPRESSED
8027 ;THE DESIRE TO CHANGE THE SOFTWARE
8028 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8029 ;THE USER TYPED CONTROL G?).
8030
8031 ;*****
8032 ;*TEST 72 STCDI AND STCDL TEST
8033 ;*
8034 ;* THIS IS A TEST OF THE STCDI AND
8035 ;* STCDL INSTRUCTIONS, NOTE THAT A
8036 ;* SUBROUTINE, STCSUB, IS USED TO
8037 ;* SET UP THE OPERANDS, EXECUTE THE STC
8038 ;* INSTRUCTION AND CHECK THE RESULT.
8039 ;*
8040 ;*****
8041 034512 000004 TST72: SCOPE
8042
8043 ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
8044 034514 WWC1:
8045 034514 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8046 034516 004737 035662 JSR PC,#STCSUB ;GO EXECUTE THE INSTRUCTION,
8047 034522 020000 000000 000000 16: .WORD 20000,0,0,0 ;AC0 OPERAND.
8048 034530 000000
8049 034532 000000 000000 26: .WORD 0,0 ;EXPECTED RESULT.
8050 034536 177777 177777 36: .WORD -1,-1 ;ERROR RES.
8051 034542 040300 46: 40300 ;FPS BEFORE EXECUTION.
8052 034544 040304 40304 ;FPS AFTER EXECUTION.
8053 034546 140304 140304 ;ANTICIPATED ERRONEOUS FPS.
8054 034550 177777 -1 ;REPORT RESULT INCORRECT.
8055 034552 104322 56: ERROR 322 ;RESULT INCORP.
8056 034554 000401 BR 66
8057 034556 104325 ERROR 325 ;EITHER (BUT FLAG)
8058 034560 66: ;ST 662
8059 ;OR CLEAR FLAG
8060 ;ST 774

```

```

8061
8062 ;EXP=0 (OCT) FL=1 FIC=0
8063 034560 WWC2:
8064 034560 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8065 034562 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8066 034566 040000 000000 000000 1$: .WORD 40000,0,0,0 ;AC ;AC0 OPERAND,
8067 034574 000000
8068 034576 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT,
8069 034602 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT,
8070 034606 040313 4$: 40313 ;FPS BEFORE EXECUTION,
8071 034610 040304 40304 ;FPS AFTER EXECUTION,
8072 034612 140304 140304 ;ANTICIPATED ERRONEOUS FPS,
8073 034614 177777 -1 ;EXPECTED FEC,
8074 034616 104322 5$: ERROR 322 ;REPORT RESULT INCORRECT,
8075 034620 000401 BR 6$
8076 034622 104326 ERROR 326 ;REPORT FPS INCORRECT,
8077 034624 6$:
8078
8079 ;EXP=37 (OCT) FL=1 FIC=1
8080 034624 WWC4:
8081 034624 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8082 034626 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8083 034632 047667 075757 157737 1$: .WORD 47667,75757,157737,167773 ;AC0 OPERAND,
8084 034640 167773
8085 034642 055675 173757 2$: .WORD 55675,173757 ;EXPECTED RESULT,
8086 034646 122102 004021 3$: .WORD 122102,004021 ;ANTICIPATED ERRONEOUS RESULT,
8087 034652 040717 4$: 40717 ;FPS BEFORE EXECUTION,
8088 034654 040700 40700 ;FPS AFTER EXECUTION,
8089 034656 140705 140705 ;ANTICIPATED ERRONEOUS FPS,
8090 034660 177777 -1 ;EXPECTED FEC,
8091 034662 104327 5$: ERROR 327 ;(BUT ENBT) ST 632
8092 034664 000401 BR 6$
8093 034666 104326 ERROR 326 ;REPORT FPS INCORRECT,
8094 034670 6$:
8095
8096 ;EXP=40 (OCT) FL=1 FIC=1
8097 034670 WWC5:
8098 034670 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8099 034672 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8100 034676 050000 000000 000000 1$: .WORD 50000,0,0,0 ;AC0 OPERAND,
8101 034704 000000
8102 034706 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT,
8103 034712 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT,
8104 034716 040700 4$: 40700 ;FPS BEFORE EXECUTION,
8105 034720 140705 140705 ;FPS AFTER EXECUTION,
8106 034722 040705 40705 ;ANTICIPATED ERRONEOUS FPS,
8107 034724 000006 6 ;EXPECTED FEC,
8108 034726 104322 5$: ERROR 322 ;REPORT RESULT INCORRECT,
8109 034730 000401 BR 6$
8110 034732 104330 ERROR 330 ;(BUT FIC) ST 004 ;REPORT FPS INCORRECT,
8111 ;TO 305 INTO
8112 034734 6$: ;315
8113
8114 ;EXP=40 (OCT) FL=1 FIC=0
8115 034734 WWC6:
8116 034734 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,

```

8117	034736	004737	035662			JSR	PC,#STCSUB		;GO EXECUTE THE INSTRUCTION.
8118	034742	050000	000000	000000	1s:	.WORD	50000,0,0,0		;AC0 OPERAND.
8119	034750	000000							
8120	034752	000000	000000		2s:	.WORD	0,0		;EXPECTED RESULT.
8121	034756	177777	177777		3s:	.WORD	-1,-1		;ANTICIPATED ERRONEOUS RESULT.
8122	034762	040312			4s:		40312		;FPS BEFORE EXECUTION.
8123	034764	040305					40305		;FPS AFTER EXECUTION.
8124	034766	140305					140305		;ANTICIPATED ERRONEOUS FPS.
8125	034770	177777					-1		;EXPECTED FEC.
8126	034772	104322			5s:	ERROR	322		;REPORT RESULT INCORRECT.
8127	034774	000401				BR	6s		
8128	034776	104331				ERROR	331		;(BUT FIC) ST 004 TO
8129	035000				6s:				;315 INTO 305
8130									
8131									
8132	035000								
8133	035000	104413				LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
8134	035002	004737	035662			JSR	PC,#STCSUB		;GO EXECUTE THE INSTRUCTION.
8135	035006	046000	000001	000000	1s:	.WORD	46000,1,0,0		;AC0 OPERAND.
8136	035014	000000							
8137	035016	000200	000001		2s:	.WORD	200,1		;EXPECTED RESULT.
8138	035022	177777	177777		3s:	.WORD	-1,-1		;ANTICIPATED ERRONEOUS RESULT.
8139	035026	040700			4s:		40700		;FPS BEFORE EXECUTION.
8140	035030	040700					40700		;FPS AFTER EXECUTION.
8141	035032	177777					-1		;ANTICIPATED ERRONEOUS FPS.
8142	035034	177777					-1		;EXPECTED FEC.
8143	035036	104322			5s:	ERROR	322		;REPORT RESULT INCORRECT.
8144	035040	000401				BR	6s		
8145	035042	104323				ERROR	323		;REPORT FPS INCORRECT.
8146	035044				6s:				
8147									
8148									
8149	035044								
8150	035044	104413				LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
8151	035046	004737	035662			JSR	PC,#STCSUB		;GO EXECUTE THE INSTRUCTION.
8152	035052	045600	000001	000000	1s:	.WORD	45600,1,0,0		;AC0 OPERAND.
8153	035060	000000							
8154	035062	000100	000000		2s:	.WORD	100,0		;EXPECTED RESULT.
8155	035066	177777	177777		3s:	.WORD	-1,-1		;ANTICIPATED ERRONEOUS RESULT.
8156	035072	040707			4s:		40707		;FPS BEFORE EXECUTION.
8157	035074	040700					40700		;FPS AFTER EXECUTION.
8158	035076	177777					-1		;ANTICIPATED ERRONEOUS FPS.
8159	035100	177777					-1		;EXPECTED FEC.
8160	035102	104322			5s:	ERROR	322		;REPORT RESULT INCORRECT.
8161	035104	000401				BR	6s		
8162	035106	104323				ERROR	323		;REPORT FPS INCORRECT.
8163	035110				6s:				
8164									
8165									
8166	035110								
8167	035110	104413				LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
8168	035112	004737	035662			JSR	PC,#STCSUB		;GO EXECUTE THE INSTRUCTION.
8169	035116	043600	000000	000000	1s:	.WORD	43600,0,0,0		;AC0 OPERAND.
8170	035124	000000							
8171	035126	040000	177777		2s:	.WORD	40000,-1		;EXPECTED RESULT.
8172	035132	000000	177777		3s:	.WORD	0,-1		;ANTICIPATED ERRONEOUS RESULT.

```
8173 035136 040600 4$: 40600 ;FPS BEFORE EXECUTION,
8174 035140 040600 40600 ;FPS AFTER EXECUTION,
8175 035142 140604 140604 ;ANTICIPATED ERRONEOUS FPS,
8176 035144 177777 -1 ;EXPECTED FEC,
8177 035146 104332 5$: ERROR 332 ;BAD CONSTANT ST 066
8178 035150 000401 BR 6$
8179 035152 104333 ERROR 333 ;REPORT FPS INCORRECT,
8180 035154 6$:
8181
8182 ;EXP=20 (OCT) FL=0 FIC=1
8183 035154 WWC10:
8184 035154 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8185 035156 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8186 035162 044000 000000 000000 1$: .WORD 44000,0,0,0 ;AC0 OPERAND,
8187 035170 000000
8188 035172 000000 177777 2$: .WORD 0,-1 ;EXPECTED RESULT,
8189 035176 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT,
8190 035202 040600 4$: 40600 ;FPS BEFORE EXECUTION,
8191 035204 140605 140605 ;FPS AFTER EXECUTION,
8192 035206 040600 40600 ;ANTICIPATED ERRONEOUS FPS,
8193 035210 000006 6 ;EXPECTED FEC,
8194 035212 104322 5$: ERROR 322 ;REPORT RESULT INCORRECT,
8195 035214 000401 BR 6$
8196 035216 104334 ERROR 334 ;BAD CONSTANT ST 066
8197 035220 6$:
8198
8199 ;EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1
8200 035220 WWC11:
8201 035220 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8202 035222 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8203 035226 142000 000000 000000 1$: .WORD 142000,0,0,0 ;AC0 OPERAND,
8204 035234 000000
8205 035236 177600 177777 2$: .WORD 177600,-1 ;EXPECTED RESULT,
8206 035242 000200 000000 3$: .WORD 200,0 ;ANTICIPATED ERRONEOUS RESULT,
8207 035246 040600 4$: 40600 ;FPS BEFORE EXECUTION,
8208 035250 040610 40610 ;FPS AFTER EXECUTION,
8209 035252 040600 40600 ;ANTICIPATED ERRONEOUS FPS,
8210 035254 177777 -1 ;EXPECTED FEC,
8211 035256 104335 5$: ERROR 335 ;(BUT ENBT) ST 632
8212 035260 000401 BR 6$
8213 035262 104336 ERROR 336 ;(SET FN) ST 473
8214 035264 6$:
8215
8216 ;EXP=37 (OCT), FL=1, FIC=1, AC NEG,
8217 035264 WWC12:
8218 035264 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8219 035266 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8220 035272 147600 000000 000000 1$: .WORD 147600,0,0,0 ;AC0 OPERAND,
8221 035300 000000
8222 035302 140000 000000 2$: .WORD 140000,0 ;EXPECTED RESULT,
8223 035306 137777 000000 3$: .WORD 137777,0 ;ANTICIPATED ERRONEOUS RESULT,
8224 035312 040700 4$: 40700 ;FPS BEFORE EXECUTION,
8225 035314 040710 40710 ;FPS AFTER EXECUTION,
8226 035316 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
8227 035320 177777 -1 ;EXPECTED FEC,
8228 035322 104337 5$: ERROR 337 ;(BUT COUT) ST 375
```

```

8229 035324 000401 BR 6S ;ST 275 TO 074
8230 035326 104323 ERROR 323 ;INTO 274
8231 035330 6S:
8232
8233 ;EXP=37 (OCT), FL=1, FIC=1, AC NEG
8234 035330 WWC13:
8235 035330 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8236 035332 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8237 035336 147600 000000 001000 1S: .WORD 147600,0,1000,0 ;AC0 OPERAND,
8238 035344 000000
8239 035346 137777 177777 2S: .WORD 137777,177777 ;EXPECTED RESULT,
8240 035352 140000 177777 3S: .WORD 140000,177777 ;ANTICIPATED ERRONEOUS RESULT,
8241 035356 040707 4S: 40707 ;FPS BEFORE EXECUTION,
8242 035360 040710 40710 ;FPS AFTER EXECUTION,
8243 035362 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
8244 035364 177777 -1 ;EXPECTED FEC,
8245 035366 104340 5S: ERROR 340 ;(BUT COUT) ST 375
8246 035370 000401 BR 6S ;TO 274 INTO 074
8247 035372 104323 ERROR 323 ;REPORT FPS INCORRECT,
8248 035374 6S:
8249
8250 ;EXP=41 (OCT), AC NEG, FL=1, FIC=1
8251 035374 WWC14:
8252 035374 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8253 035376 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8254 035402 150200 000000 000000 1S: .WORD 150200,0,0,0 ;AC0 OPERAND,
8255 035410 000000
8256 035412 000000 000000 2S: .WORD 0,0 ;EXPECTED RESULT,
8257 035416 177777 177777 3S: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT,
8258 035422 040700 4S: 40700 ;FPS BEFORE EXECUTION,
8259 035424 140705 140705 ;FPS AFTER EXECUTION,
8260 035426 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
8261 035430 000006 6 ;EXPECTED FEC,
8262 035432 104322 5S: ERROR 322 ;REPORT RESULT INCORRECT,
8263 035434 000401 BR 6S
8264 035436 104341 ERROR 341 ;(BUT EZBT) ST 377
8265 035440 6S:
8266 ;EXP=40 (OCT), AC NEG, FL=1, FIC=1
8267 035440 WWC15:
8268 035440 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8269 035442 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8270 035446 150000 000001 000000 1S: .WORD 150000,1,0,0 ;AC0 OPERAND,
8271 035454 000000
8272 035456 000000 000000 2S: .WORD 0,0 ;EXPECTED RESULT,
8273 035462 100000 177600 3S: .WORD 100000,-200 ;ANTICIPATED ERRONEOUS RESULT,
8274 035466 040700 4S: 40700 ;FPS BEFORE EXECUTION,
8275 035470 140705 140705 ;FPS AFTER EXECUTION,
8276 035472 040700 40700 ;ANTICIPATED ERRONEOUS FPS,
8277 035474 000006 6 ;EXPECTED FEC,
8278 035476 104342 5S: ERROR 342 ;(BUT COUT) ST 360
8279 035500 000401 BR 6S ;TO 654 INTO 454
8280 035502 104323 ERROR 323 ;REPORT FPS INCORRECT,
8281 035504 6S:
8282
8283 ;EXP=40, AC NEGATIVE, FL=1, FIC=1
8284 035504 WWC16:
  
```

```

8285 035504 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8286 035506 004737 035662 JSR PC,#STCSUB ;GO EXECUTE THE INSTRUCTION,
8287 035512 150001 000000 000000 1$: .WORD 150001,0,0,0 ;AC0 OPERAND.
8288 035520 000000
8289 035522 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
8290 035526 077400 000000 3$: .WORD 77400,0 ;ANTICIPATED ERRONEOUS RESULT.
8291 035532 040700 4$: 40700 ;FPS BEFORE EXECUTION,
8292 035534 140705 140705 ;FPS AFTER EXECUTION.
8293 035536 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
8294 035540 000006 6 ;EXPECTED FEC.
8295 035542 104343 5$: ERROR 343 ;REPORT RESULT INCORRECT.
8296 035544 000401 BR 6$
8297 035546 104323 ERROR 323 ;REPORT FPS INCORRECT.
8298 035550 6$:
8299
8300
8301 ;EXP 40 (OCT), AC MOST NEG LONG INT, FL=1
8302 ;FIC=1
8303 035550 WWC17:
8304 035550 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8305 035552 004737 035662 JSR PC,#STCSUB ;GO EXECUTE THE INSTRUCTION.
8306 035556 150000 000000 000000 1$: .WORD 150000,0,0,0 ;AC0 OPERAND.
8307 035564 000000
8308 035566 100000 000000 2$: .WORD 100000,0 ;EXPECTED RESULT.
8309 035572 000000 000000 3$: .WORD 0,0 ;ANTICIPATED ERRONEOUS RESULT.
8310 035576 040700 4$: 40700 ;FPS BEFORE EXECUTION,
8311 035600 040710 40710 ;FPS AFTER EXECUTION.
8312 035602 140705 140705 ;ANTICIPATED ERRONEOUS FPS.
8313 035604 177777 -1 ;EXPECTED FEC.
8314 035606 104344 5$: ERROR 344 ;(BUT NBIT) ST 654
8315 035610 000401 BR 6$ ;OR (BUT COUT) ST 454
8316 035612 104323 ERROR 323 ;REPORT FPS INCORRECT.
8317 035614 6$:
8318
8319 ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
8320
8321 035614 WWC18:
8322 035614 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
8323 035616 004737 035662 JSR PC,#STCSUB ;GO EXECUTE THE INSTRUCTION.
8324 035622 144000 000001 000000 1$: .WORD 144000,1,0,0 ;AC0 OPERAND.
8325 035630 000000
8326 035632 100000 177777 2$: .WORD 100000,-1 ;EXPECTED RESULT.
8327 035636 100000 177400 3$: .WORD 100000,177400 ;ANTICIPATED ERRONEOUS RESULT.
8328 035642 040600 4$: 40600 ;FPS BEFORE EXECUTION,
8329 035644 040610 40610 ;FPS AFTER EXECUTION,
8330 035646 140605 140605 ;ANTICIPATED ERRONEOUS FPS.
8331 035650 177777 -1 ;EXPECTED FEC.
8332 035652 104345 5$: ERROR 345 ;(BUT FL) ST 633
8333 035654 000401 BR 6$ ;TO 655 INTO 654
8334 035656 104323 ERROR 323 ;REPORT FPS INCORRECT.
8335
8336 035660 000534 6$: BR WWCDONE
8337
8338 ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
8339 ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS, A CALL
8340 ;TO IT IS MADE THUS:

```

```

8341 ;
8342 ; JSR PC,@#STCSUB
8343 ; ACARG: .WORD X,X,X,X ;AC OPERAND
8344 ; RES: .WORD X,X ;EXPECTED RESULT
8345 ; ERRES: .WORD X,X ;ERROR RESULT
8346 ; FPSB: .WORD X ;FPS BEFORE EXECUTION
8347 ; FPSA: .WORD X ;FPS AFTER EXECUTION
8348 ; ERFPS: .WORD X ;ERROR FPS,
8349 ; FEC: .WORD X ;EXPECTED FEC
8350 ; ERR1: ERROR X ;DATA ERROR,
8351 ; BR CONT
8352 ; ERR2: ERROR X ;FPS ERROR,
8353 ; CONT: ;RETURN ADDRESS
8354 ;
8355 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR), THEN
8356 ;THE STCDI OR STCDL INSTRUCTION IS EXECUTED,
8357 ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
8358 ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
8359 ;TO THE CALLING ROUTINE AT CONT, IF THE FPS IS BAD STCSUB
8360 ;COMPARE IT TO ERROR FPS, IF THIS MATCHES THEN STCSUB WILL RETURN
8361 ;TO THE ERROR CALL AT ERR2, OTHERWISE STCSUB ITSELF
8362 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT, IF THE RESULT OF THE
8363 ;STCDI OR STCDL IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
8364 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN
8365 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCSUB
8366 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
8367 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCSUB WILL
8368 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT,
8369
8370 035662 012601 STCSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS,
8371 035664 012700 000200 MOV #200,R0 ;SET UP THE AC0 OPERAND,
8372 035670 170100 LDFPS R0
8373 035672 010100 MOV R1,R0
8374 035674 172410 LDD (R0),AC0
8375 035676 012702 036142 MOV #STCIBF,R2 ;INITIALIZE THE OUT PUT BUFFER,
8376 035702 012700 000004 MOV #4,R0
8377 035706 012722 177777 1s: MOV #-1,(R2)+
8378 035712 077003 SOB R0,1s
8379 035714 016100 000020 MOV 20(R1),R0 ;SET THE FPS,
8380 035720 170100 LDFPS R0
8381 035722 012737 035734 001236 MOV #2s,@#TMP2
8382 035730 012700 036142 MOV #STCIBF,R0
8383 035734 175410 2s: STCDL AC0,(R0) ;TEST INSTRUCTION,
8384
8385 035736 170204 STFPS R4 ;GET THE FPS,
8386 035740 170305 STST R5 ;GET THE FEC,
8387 035742 010102 MOV R1,R2
8388 035744 010237 001240 MOV R2,@#TMP3
8389 035750 062702 000010 ADD #10,R2
8390 035754 010237 001244 MOV R2,@#TMP5
8391 035760 012737 036142 001242 MOV #STCIBF,@#TMP4
8392 035766 010437 001250 MOV R4,@#TMP7
8393 035772 016137 000022 001252 MOV 22(R1),@#TMP10
8394 036000 010102 MOV R1,R2
8395 036002 062702 000010 ADD #10,R2
8396 036006 012700 036142 MOV #STCIBF,R0 ;SEE IF THE RESULT IS CORRECT,

```

```

8397 036012 012703 000002          MOV      #2,R3
8398 036016 022022          3$:     CMP      (R0)+,(R2)+
8399 036020 001014          BNE     15$
8400 036022 077303          SOB     R3,3$
8401 036024 016102 000022          MOV     22(R1),R2
8402 036030 020204          CMP     R2,R4          ;SEE IF THE FPS IS CORRECT,
8403 036032 001025          BNE     20$          ;BRANCH IF INCORRECT.
8404 036034 005702          TST     R2
8405 036036 100003          BPL     4$
8406 036040 026105 000026          CMP     26(R1),R5      ;SEE IF THE FEC IS CORRECT,
8407 036044 001027          BNE     25$          ;BRANCH IF INCORRECT.
8408
8409 036046 000161 000036          4$:     JMP     36(R1)          ;RETURN.
8410          ;DATA ERROR:
8411          ;SEE IF THE FAILURE WAS ANTICIPATED,
8412 036052 010102          15$:    MOV     R1,R2
8413 036054 062702 000014          ADD     #14,R2
8414 036060 012700 036142          MOV     #STCIBF,R0
8415 036064 012703 000002          MOV     #2,R3
8416 036070 022022          16$:    CMP     (R0)+,(R2)+
8417 036072 001003          BNE     17$
8418 036074 077303          SOB     R3,16$
8419 036076 000161 000030          JMP     30(R1)
8420 036102
8421          17$:
8422 036102 104322          ;FAILURE WAS NOT ANTICIPATED SO REPORT INCORRECT RESULT HERE,
8423 036104 000760          18$:    ERROR   322          ;DATA BAD
8424          BR     4$
8425          ;FPS INCORRECT, SO SEE IF FAILURE WAS ANTICIPATED,
8426 036106 020461 000024          20$:    CMP     R4,24(R1)
8427 036112 001002          BNE     21$
8428 036114 000161 000034          JMP     34(R1)
8429 036120
8430          21$:
8431 036120 104323          ;NOT ANTICIPATED SO REPORT BAD FPS HERE,
8432 036122 000751          22$:    ERROR   323          ;FPS BAD
8433          BR     4$
8434          ;REPORT INCORRECT FEC,
8435 036124 016137 000026 001256          25$:    MOV     26(R1),@#STMP12
8436 036132 010537 001254          MOV     R5,@#STMP11
8437 036136 104324          26$:    ERROR   324
8438 036140 000742          BR     4$
8439
8440          ;DATA BUFFER:
8441 036142 177777 177777 177777          STCIBF: .WORD  -1,-1,-1,-1
8442 036150 177777
8443
8444 036152
8445 036152 104412          WWC DONE:
8446          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
8447          ;SEE IF THE USER HAS EXPRESSED
8448          ;THE DESIRE TO CHANGE THE SOFTWARE
8449          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8450          ;THE USER TYPED CONTROL G?),
8451
8452          ;*****

```



```

8453 ;*TEST 73 STCFL AND STCFI TEST
8454 ;*
8455 ;* THIS IS A TEST OF STCFL AND STCFI. IT
8456 ;* MAKES USE OF THE SAME SUBROUTINE, STCSUB,
8457 ;* WHICH WAS USED TO TEST STCDL AND STCDI,
8458 ;*
8459 ;*****
8460 036154 000004 TST73: SCOPE
8461
8462
8463 ;EXPONENT=37, FL=1
8464 036156 XXC1:
8465 036156 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8466 036160 004737 035662 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION,
8467 036164 047777 177777 177777 1s: .WORD 47777,-1,-1,-1 ;AC0 OPERAND,
8468 036172 177777
8469 036174 077777 177600 2s: .WORD 77777,177600 ;EXPECTED RESULT,
8470 036200 077777 177777 3s: .WORD 77777,177777 ;ANTICIPATED ERRONEOUS RESULT,
8471 036204 040100 4s: 40100 ;FPS BEFORE EXECUTION,
8472 036206 040100 40100 ;FPS AFTER EXECUTION,
8473 036210 177777 -1 ;ANTICIPATED ERRONEOUS FPS,
8474 036212 177777 -1 ;EXPECTED FEC.
8475 036214 104346 5s: ERROR 346 ;X11(1,0)+0 ST 773X
8476 036216 000401 BR 6s
8477 036220 104323 ERROR 323 ;REPORT FPS INCORRECT,
8478 036222
8479
8480 036222 XXCDONE:
8481 036222 104412 RSETUP ;GO INITIALIZE THE FPS AND STACK, AND
8482 ;SEE IF THE USER HAS EXPRESSED
8483 ;THE DESIRE TO CHANGE THE SOFTWARE
8484 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
8485 ;THE USER TYPED CONTROL G?),
8486
8487
8488 ;*****
8489 ;*TEST 74 STEXP TEST
8490 ;*
8491 ;* THIS IS A TEST OF THE STEXP
8492 ;* INSTRUCTION
8493 ;*
8494 ;*****
8495 036224 000004 TST74: SCOPE
8496
8497 ; EXP = 100 (EXCESS 200)
8498 036226 YXC1:
8499 036226 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8500 036230 004737 036514 JSR PC,@#STXSUB
8501 036234 020000 000000 000000 1s: .WORD 20000,0,0,0 ;AC
8502 036242 000000
8503 036244 177700 2s: -100 ;EXP RES
8504 036246 052525 3s: 52525 ;ERROR EXP.
8505 036250 040000 4s: 40000 ;FPSB
8506 036252 040010 40010 ;FPSA
8507 036254 040000 40000 ;ERROR FPS
8508 036256 104347 5s: ERROR 347 ;BAD EXP

```

```

8509 036260 000401 BR 6$
8510 036262 104352 ERROR 352 ;+(BUT ENBT) ST 376
8511 036264 6$:
8512
8513 ; EXP = 200 (EXCESS 200)
8514 036264 YYC2:
8515 036264 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8516 036266 004737 036514 JSR PC,#STXSUB ;GO EXECUTE THE INSTRUCTION,
8517 036272 040000 000000 000000 1$: .WORD 40000,0,0,0 ;AC0 OPERAND,
8518 036300 000000
8519 036302 000000 2$: 0 ;EXPECTED EXPONENT RESULT,
8520 036304 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT,
8521 036306 040000 4$: 40000 ;FPS BEFORE EXECUTION,
8522 036310 040004 40004 ;FPS AFTER EXECUTION,
8523 036312 040000 40000 ;ANTICIPATED ERRONEOUS FPS,
8524 036314 104347 5$: ERROR 347 ;REPORT RESULT INCORRECT,
8525 036316 000401 BR 6$
8526 036320 104353 ERROR 353 ;(BUT EZBT) ST 071
8527 ;TO 072 INT 272
8528 036322 6$:
8529
8530 ; EXP = 201 (EXCESS 200)
8531
8532 036322 YYC3:
8533 036322 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8534 036324 004737 036514 JSR PC,#STXSUB ;GO EXECUTE THE INSTRUCTION,
8535 036330 040200 000000 000000 1$: .WORD 40200,0,0,0 ;AC0 OPERAND,
8536 036336 000000
8537 036340 000001 2$: 1 ;EXPECTED EXPONENT RESULT,
8538 036342 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT,
8539 036344 040000 4$: 40000 ;FPS BEFORE EXECUTION,
8540 036346 040000 40000 ;FPS AFTER EXECUTION,
8541 036350 040004 40004 ;ANTICIPATED ERRONEOUS FPS,
8542 036352 104347 5$: ERROR 347 ;REPORT RESULT INCORRECT,
8543 036354 000401 BR 6$
8544 036356 104354 ERROR 354 ;(BUT EZBT) ST 071
8545 036360 6$: ;TO 272 INTO 072
8546
8547 ; EXP = 375 (EXCESS 200)
8548
8549 036360 YYC4:
8550 036360 104413 LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
8551 036362 004737 036514 JSR PC,#STXSUB ;GO EXECUTE THE INSTRUCTION,
8552 036366 077200 000000 000000 1$: .WORD 77200,0,0,0 ;AC0 OPERAND,
8553 036374 000000
8554 036376 000175 2$: 175 ;EXPECTED EXPONENT RESULT,
8555 036400 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT,
8556 036402 040000 4$: 40000 ;FPS BEFORE EXECUTION,
8557 036404 040000 40000 ;FPS AFTER EXECUTION,
8558 036406 040010 40010 ;ANTICIPATED ERRONEOUS FPS,
8559 036410 104347 5$: ERROR 347 ;REPORT RESULT INCORRECT,
8560 036412 000401 BR 6$
8561 036414 104355 ERROR 355 ;(BUT ENBT) ST 376
8562 036416 6$: ;TO 471 INTO 071
8563
8564 ; EXP = 1 (EXCESS 200)

```

8565

8566 036416  
 8567 036416 104413  
 8568 036420 004737  
 8569 036424 000200  
 8570 036432 000000  
 8571 036434 177601  
 8572 036436 052525  
 8573 036440 040000  
 8574 036442 040010  
 8575 036444 040000  
 8576 036446 104347  
 8577 036450 000401  
 8578 036452 104352  
 8579 036454

YYC5:

LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION,  
 1\$: .WORD 200,0,0,0 ;AC0 OPERAND,  
 2\$: -177 ;EXPECTED EXPONENT RESULT,  
 3\$: 52525 ;ANTICIPATED ERRONEOUS RESULT,  
 4\$: 40000 ;FPS BEFORE EXECUTION,  
 40010 ;FPS AFTER EXECUTION,  
 40000 ;ANTICIPATED ERRONEOUS FPS,  
 5\$: ERROR 347 ;REPORT RESULT INCORRECT,  
 BR 6\$  
 ERROR 352 ;REPORT FPS INCORRECT,  
 6\$:  
 ; EXP = 156 (EXCESS 200)

8580

8581  
 8582  
 8583 036454  
 8584 036454 104413  
 8585 036456 004737  
 8586 036462 033400  
 8587 036470 000000  
 8588 036472 177756  
 8589 036474 052525  
 8590 036476 047707  
 8591 036500 047710  
 8592 036502 177777  
 8593 036504 104347  
 8594 036506 000401  
 8595 036510 104350  
 8596  
 8597 036512 000510  
 8598

YYC6:

LPERR ;SET UP THE LOOP ON ERROR ADDRESS,  
 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION,  
 1\$: .WORD 33400,0,0,0 ;AC0 OPERAND,  
 2\$: -22 ;EXPECTED EXPONENT RESULT,  
 3\$: 52525 ;ANTICIPATED ERRONEOUS RESULT,  
 4\$: 47707 ;FPS BEFORE EXECUTION,  
 47710 ;FPS AFTER EXECUTION,  
 -1 ;ANTICIPATED ERRONEOUS FPS,  
 5\$: ERROR 347 ;REPORT RESULT INCORRECT,  
 BR 6\$  
 ERROR 350 ;REPORT FPS INCORRECT,  
 6\$: BR YYCDONE

8599

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

;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE  
 ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL  
 ;TO IT IS MADE THUS:  
 ;  
 ; JSR PC,@#STXSUB  
 ; ACARG: .WORD X,X,X,X ;AC OPERAND  
 ; RES: .WORD X ;EXPECTED RESULT  
 ; ERRES: .WORD X ;ERROR RESULT  
 ; FPSB: .WORD X ;FPS BEFORE EXECUTION  
 ; FPSA: .WORD X ;FPS AFTER EXECUTION  
 ; ERFPS: .WORD X ;ERROR FPS,  
 ; ERR1: ERROR X ;DATA ERROR,  
 ; BR CONT  
 ; ERR2: ERROR X ;FPS ERROR,  
 ; CONT: ;RETURN ADDRESS  
 ;  
 ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR), THEN  
 ;THE STEXP 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

```

8621 ;TO THE ERROR CALL AT ERR2, OTHERWISE STXSUB ITSELF
8622 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
8623 ;STEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
8624 ;ANTICIPATED FAILING DATA PATTERN, ERRES, IF THE FAILURE IN
8625 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STXSUB
8626 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1, OTHERWISE THE
8627 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STXSUB WILL
8628 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
8629
8630 STXSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS,
8631 MOV R1,R2
8632 MOV R2,@#STMP3
8633 ADD #10,R2
8634 MOV (R2)+,@#STMP5
8635 MOV #10,@#STMP2
8636 MOV #123456,@#STXBF
8637 MOV #76543,@#STXBF+2
8638 MOV #200,R0
8639 LDFPS R0
8640 MOV R1,R0 ;SET UP THE AC0 OPERAND.
8641 LDD (R0),AC0
8642 MOV 16(R1),R0 ;SET THE FPS.
8643 LDFPS R0
8644 MOV #STXBF,R0
8645 10: STEXP AC0,(R0) ;TEST INSTRUCTION.
8646 STFPS R4 ;GET FPS.
8647 MOV R4,@#STMP7
8648 MOV 16(R1),@#STMP10
8649 MOV @#STXBF,@#STMP4
8650 CMP 10(R1),@#STXBF ;WAS RESULT CORRECT?
8651 BEQ 50 ;BRANCH IF CORRECT.
8652 CMP 12(R1),@#STXBF ;OTHERWISE SEE IF THE FAILURE WAS ANTICIPATED.
8653 BNE 20
8654 JMP 22(R1)
8655
8656 ;IF NOT ANTICIPATED REPORT ERROR HERE,
8657 20:
8658 30: ERROR 347 ;EXP BAD
8659 40: JMP 30(R1)
8660
8661 50: CMP R4,16(R1) ;SEE IF THE FPS IS CORRECT.
8662 BEQ 100 ;BRANCH IF CORRECT.
8663 CMP R4,20(R1) ;SEE IF THE FAILURE WAS ANTICIPATED.
8664 BNE 60
8665 JMP 26(R1)
8666
8667 ;FPS ERROR WAS NOT ANTICIPATED SO REPORT ERROR HERE,
8668 60:
8669 70: ERROR 350 ;FPS BAD
8670 BR 40
8671
8672 ;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
8673 100: CMP #76543,@#STXBF+2
8674 BEQ 40
8675 110: ERROR 351 ;FDFL+0 ST 347X
8676 BR 40

```

8677  
 8678 036720 177777  
 8679 036722 177777 177777 177777 STXBF: -1  
 8680 036730 177777 177777  
 8681  
 8682 036734  
 8683 036734 104412  
 8684  
 8685  
 8686  
 8687  
 8688  
 8689  
 8690  
 8691  
 8692  
 8693  
 8694  
 8695  
 8696  
 8697  
 8698  
 8699  
 8700 036736 000004  
 8701  
 8702 036740  
 8703 036740 104413  
 8704 036742 012700 040000  
 8705 036746 170100  
 8706  
 8707 036750 170003  
 8708  
 8709 036752 012700 037126  
 8710 036756 012710 177777  
 8711 036762 012760 177777 000002  
 8712 036770 012737 036776 001236  
 8713 036776 170310  
 8714  
 8715 037000 170204  
 8716 037002 012700 037126  
 8717 037006 011037 001240  
 8718 037012 016037 000002 001242  
 8719 037020 012737 000002 001244  
 8720 037026 012737 036750 001246  
 8721 037034 010437 001250  
 8722 037040 012737 140000 001252  
 8723  
 8724 037046 022710 000002  
 8725 037052 001010  
 8726 037054 022760 036750 000002  
 8727 037062 001006  
 8728 037064 022704 140000  
 8729 037070 001013  
 8730 037072 000422  
 8731  
 8732

```

-1
WORD -1,-1,-1,-1,-1

YYCDONE:
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 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
;*
;*****
TST75; SCOPE

ZZC1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS,
MOV #40000,R0 ;SET FPS. FID=1.
LDFPS R0

ZZC2:
WORD 170003 ;ILLEGAL FPP
;OP CODE
MOV #ZZCBF,R0 ;SET UP THE OUTPUT BUFFER.
MOV #-1,(R0)
MOV #-1,2(R0)
MOV #ZZC3,@#TMP2

ZZC3:
STST (R0) ;GET FEC AND
;FEA
STFPS R4 ;GET FPS.
MOV #ZZCBF,R0
MOV (R0),@#TMP3
MOV 2(R0),@#TMP4
MOV #2,@#TMP5
MOV #ZZC2,@#TMP6
MOV R4,@#TMP7
MOV #140000,@#TMP10

CMP #2,(R0) ;SEE IF FEC IS CORRECT.
BNE ZZC5 ;BRANCH IF INCORRECT.
CMP #ZZC2,2(R0) ;SEE IF FEA, ADDRESS, IS CORRECT.
BNE ZZC10 ;BRANCH IF INCORRECT.
CMP #140000,R4 ;SEE IF FPS IS CORRECT.
BNE ZZC15 ;BRANCH IF INCORRECT.
BR ZZCDONE

;REPORT FEC INCORRECT
  
```

DFPPCA,P11 31-OCT-76 17:16 T75 STST TEST

8733 037074  
 8734 037074 104356  
 8735 037076 000420  
 8736

ZZC5:  
 1\$: ERROR 356 ;STST BAD  
 BR ZZCDONE ;FECX

8737  
 8738 037100 022760 177777 000002  
 8739 037106 001402  
 8740 037110 104357  
 8741 037112 000412  
 8742 037114  
 8743 037114 104360  
 8744 037116 000410

;REPORT FEA INCORRECT  
 ZC10: CMP #-1,2(R0)  
 BEQ ZC12  
 1\$: ERROR 357 ;STST BAD FEA  
 BR ZZCDONE  
 ZC12:  
 1\$: ERROR 360 ;SET FD FL ST 636  
 BR ZZCDONE

8745  
 8746  
 8747 037120  
 8748 037120 104361  
 8749 037122 000406  
 8750

;REPORT FPS INCORRECT  
 ZC15:  
 1\$: ERROR 361 ;FPS X AFTER ST ST  
 BR ZZCDONE

8751  
 8752 037124 177777  
 8753 037126 177777 177777 177777  
 8754 037134 177777  
 8755 037136 177777  
 8756

;DATA BUFFER:  
 -1  
 ZC16: .WORD -1,-1,-1,-1  
 -1

8757 037140  
 8758 037140 104412  
 8759  
 8760  
 8761  
 8762

ZZCDONE:  
 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?).

8763  
 8764 037142  
 8765  
 8766  
 8767  
 8768  
 8769

TST76:

8770  
 8771  
 8772  
 8773  
 8774  
 8775  
 8776

.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

8777 037142  
 8778 037142 000004  
 8779 037144 005067 141732  
 8780 037150 005067 142126  
 8781 037154 005267 142144  
 8782 037160 042767 100000 142136  
 8783 037166 005327  
 8784 037170 000001  
 8785 037172 003074  
 8786 037174 012737  
 8787 037176 000001  
 8788 037200 037170

SEOP:  
 SCOPE  
 CLR STSTNM ;;ZERO THE TEST NUMBER  
 CLR STIMES ;;ZERO THE NUMBER OF ITERATIONS  
 INC \$PASS ;;INCREMENT THE PASS NUMBER  
 BIC #100000,\$PASS ;;DON'T ALLOW A NEG. NUMBER  
 DEC (PC)+ ;;LOOP?  
 SEOPCT: .WORD 1  
 BGT \$DOAGN ;;YES  
 MOV (PC)+,@(PC)+ ;;RESTORE COUNTER  
 SENDCT: .WORD 1  
 SEOPCT

```

8789 037202 104401 037210          TYPE      ,65$          ;;TYPE ASCIZ STRING
8790 037206 000407          BR        64$          ;;GET OVER THE ASCIZ
8791          ;;65$: .ASCIZ <12><15>/END PASS #/
8792 037226          64$:
8793 037226 016746 142072          MOV        $PASS,-(SP)      ;;SAVE $PASS FOR TYPEOUT
8794          ;;TYPE PASS NUMBER IN OCTAL
8795 037232 104403          TYPOS          ;;GO TYPE--OCTAL ASCII
8796 037234 006          .BYTE      6          ;;TYPE 6 DIGITS
8797 037235 000          .BYTE      0          ;;SUPPRESS LEADING ZEROS
8798 037236 104401 037244          TYPE      ,67$          ;;TYPE ASCIZ STRING
8799 037242 000421          BR        66$          ;;GET OVER THE ASCIZ
8800          ;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
8801 037306          66$:
8802 037306 016746 141600          MOV        $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
8803          ;;TOTAL NUMBER OF ERRORS IN OCTAL
8804 037312 104403          TYPOS          ;;GO TYPE--OCTAL ASCII
8805 037314 006          .BYTE      6          ;;TYPE 6 DIGITS
8806 037315 000          .BYTE      0          ;;SUPPRESS LEADING ZEROS
8807 037316 104401 001313          TYPE      ,SCLRF        ;;TYPE CARRIAGE RETURN, LINE FEED
8808 037322 005067 141564          CLR        $ERTTL        ;;CLEAR ERROR TOTAL
8809 037326 013700 000042          $GET42: MOV    @#42,R0      ;;GET MONITOR ADDRESS
8810 037332 001414          BEQ        $DOAGN        ;;BRANCH IF NO MONITOR
8811 037334 005046          CLR        -(SP)         ;;INSURE THE "T" BIT IS CLEAR
8812 037336 012746 037344          MOV        #$CLR,T,-(SP)  ;;SETUP FOR AN RTI OR RTT
8813 037342 000426          BR        $RTRN         ;;GO DO AN RTI OR RTT TO LOAD THE PSW
8814          ;;WITH A CLEARED "T" BIT
8815 037344          $CLR,T:
8816 037344 013700 000042          MOV        @#42,R0      ;;INSURE R0 CONTAINS THE MONITORS
8817 037350 001405          BEQ        $DOAGN        ;;RETURN ADDRESS
8818 037352 000005          RESET          ;;CLEAR THE WORLD
8819 037354 004710          $ENDAD: JSR    PC,(R0)    ;;GO TO MONITOR
8820 037356 000240          NOP          ;;SAVE ROOM
8821 037360 000240          NOP          ;;FOR
8822 037362 000240          NOP          ;;ACT11
8823 037364          $DOAGN:
8824 037364 104400          TRAP          ;;PUSH OLD PSW AND PC ON STACK
8825 037366 042716 000020          BIC        #20,(SP)     ;;CLEAR THE "T" BIT
8826 037372 032777 010000 141540          BIT        #BIT12,@SWR  ;;RUN WITH TRACE TRAP?
8827 037400 001005          BNE        1$          ;;BR IF NO
8828 037402 005167 000020          COM        $TBIT        ;;IS IT TIME FOR TRACE TRAP
8829 037406 100402          BMI        1$          ;;BR IF NO
8830 037410 052716 000020          BIS        #20,(SP)     ;;SET TRACE TRAP
8831 037414 012746 037422          1$: MOV        #$LOOP,-(SP) ;;JUMP TO START OF TEST
8832 037420 000002          $RTRN: RTI          ;;RETURN--THIS IS CHANGED TO
8833          ;;AN "RTT" IF "RTT" IS A LEGAL
8834          ;;INSTRUCTION
8835 037422          $LOOP:
8836 037422 000137          JMP        @(PC)+        ;;RETURN
8837 037424 006576          $RTNAD: .WORD    LOOP
8838 037426 000000          $TBIT: .WORD    0          ;;"T" BIT STATE INDICATOR
8839 037430 377 377 000          $ENULL: .BYTE   -1,-1,0  ;;NULL CHARACTER STRING
8840          .EVEN
8841
8842          ,SBTTL SCOPE HANDLER ROUTINE
8843
8844          ;;*****

```

```

8845 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8846 ;*AND LOAD THE TEST NUMBER($STSTNM) INTO THE DISPLAY REG,(DISPLAY<7:0>)
8847 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8848 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8849 ;*SW14=1 LOOP ON TEST
8850 ;*SW11=1 INHIBIT ITERATIONS
8851 ;*SW09=1 LOOP ON ERROR
8852 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
8853 ;*CALL
8854 ;* SCOPE ;;SCOPE=IOT
8855
8856 $SCOPE:
8857 037434 104406 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
8858 037436 032777 040000 141474 1$: BIT #BIT14,@SWR ;;LOOP ON PRESENT TEST?
8859 037444 001114 BNE $OVER ;;YES IF SW14=1
8860 ;*****START OF CODE FOR THE XOR TESTER*****
8861 037446 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
8862 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
8863 037450 013746 000004 MOV @#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
8864 037454 012737 037474 000004 MOV #5$,@#ERRVEC ;;SET FOR TIMEOUT
8865 037462 005737 177060 TST @#177060 ;;TIME OUT ON XOR?
8866 037466 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
8867 037472 000463 BR $SVLAD ;;GO TO THE NEXT TEST
8868 037474 022626 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
8869 037476 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
8870 037502 000423 BR 7$ ;;LOOP ON THE PRESENT TEST
8871 037504 6$: ;*****END OF CODE FOR THE XOR TESTER*****
8872 037504 032777 000400 141426 BIT #BIT08,@SWR ;;LOOP ON SPEC. TEST?
8873 037512 001404 BEQ 2$ ;;BR IF NO
8874 037514 127767 141420 141360 CMPB @SWR,$STSTNM ;;ON THE RIGHT TEST? SWR<7:0>
8875 037522 001465 BEQ $OVER ;;BR IF YES
8876 037524 105767 141353 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
8877 037530 001421 BEQ 3$ ;;BR IF NO
8878 037532 126767 141357 141343 CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
8879 037540 101015 BHI 3$ ;;BR IF NO
8880 037542 032777 001000 141370 BIT #BIT09,@SWR ;;LOOP ON ERROR?
8881 037550 001404 BEQ 4$ ;;BR IF NO
8882 037552 016767 141332 141326 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
8883 037560 000446 BR $OVER
8884 037562 105067 141315 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
8885 037566 005067 141510 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
8886 037572 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
8887 037574 032777 004000 141336 3$: BIT #BIT11,@SWR ;;INHIBIT ITERATIONS?
8888 037602 001011 BNE 1$ ;;BR IF YES
8889 037604 005767 141514 TST $PASS ;;IF FIRST PASS OF PROGRAM
8890 037610 001406 BEQ 1$ ;; INHIBIT ITERATIONS
8891 037612 005267 141266 INC $ICNT ;;INCREMENT ITERATION COUNT
8892 037616 026767 141460 141260 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
8893 037624 002024 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
8894 037626 012767 000001 141250 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
8895 037634 016767 000052 141440 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
8896 037642 105267 141234 $SVLAD: INCB $STSTNM ;;COUNT TEST NUMBERS
8897 037646 116767 141230 141446 MOVB $STSTNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
8898 037654 011667 141226 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
8899 037660 011667 141224 MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
8900 037664 005067 141414 CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
    
```



```

8901 037670 112767 000001 141217          MOVB    #1,$ERMAX      ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
8902 037676 016777 141200 141236 $OVER:  MOV    $STSNM,@DISPLAY ;; DISPLAY TEST NUMBER
8903 037704 016716 141176          MOV     $LPADR,(SP)   ;; FUDGE RETURN ADDRESS
8904 037710 000002          RTI                    ;; FIXES PS
8905 037712 000001          $MXCNT: 1            ;; MAX. NUMBER OF ITERATIONS
8906
8907          .SBTTL  ERROR HANDLER ROUTINE
8908
8909          ;;*****
8910          ;;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
8911          ;;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
8912          ;;*AND GO TO ERTYPE ON ERROR
8913          ;;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8914          ;;*SW15=1      HALT ON ERROR
8915          ;;*SW13=1      INHIBIT ERROR TYPEOUTS
8916          ;;*SW10=1      BELL ON ERROR
8917          ;;*SW09=1      LOOP ON ERROR
8918          ;;*CALL
8919          ;;*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
8920
8921          $ERROR:
8922 037714 104406          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
8923 037716 105267 141161 7$:      INCB    $ERFLG      ;; SET THE ERROR FLAG
8924 037722 001775          BEQ     7$          ;; DON'T LET THE FLAG GO TO ZERO
8925 037724 016777 141152 141210  MOV    $STSNM,@DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
8926 037732 032777 002000 141200  BIT    #BIT10,@SWR    ;; BELL ON ERROR?
8927 037740 001402          BEQ     1$          ;; NO - SKIP
8928 037742 104401 001306          TYPE    , $BELL      ;; RING BELL
8929 037746 005267 141140 1$:      INC    $ERTTL      ;; COUNT THE NUMBER OF ERRORS
8930 037752 011667 141140          MOV    (SP),$ERRPC  ;; GET ADDRESS OF ERROR INSTRUCTION
8931 037756 162767 000002 141132  SUB    #2,$ERRPC
8932 037764 117767 141126 141122  MOVB   @ERRPC,$ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
8933 037772 032777 020000 141140  BIT    #BIT13,@SWR  ;; SKIP TYPEOUT IF SET
8934 040000 001004          BNE    20$         ;; SKIP TYPEOUTS
8935 040002 004767 002124          JSR    PC,ERTYPE   ;; GO TO USER ERROR ROUTINE
8936 040006 104401 001313          TYPE    , $CRLF
8937 040012
8938 040012 122767 000001 141316 20$:   CMPB   #APTENV,$ENV  ;; RUNNING IN APT MODE
8939 040020 001007          BNE    2$          ;; NO, SKIP APT ERROR REPORT
8940 040022 116767 141066 000004  MOVB   $ITEMB,21$   ;; SET ITEM NUMBER AS ERROR NUMBER
8941 040030 004767 000740          JSR    PC,$ATY4    ;; REPORT FATAL ERROR TO APT
8942 040034 000
8943 040035 000
8944 040036 000777          .BYTE  0
8945 040040 005777 141074 22$:   BR     22$         ;; APT ERROR LOOP
8946 040044 100002          .BYTE  0
8947 040046 000000          TST    @SWR        ;; HALT ON ERROR
8948 040050 104406          BPL    3$          ;; SKIP IF CONTINUE
8949 040052 032777 001000 141060 3$:   HALT
8950 040060 001402          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
8951 040062 016716 141022          BIT    #BIT09,@SWR ;; LOOP ON ERROR SWITCH SET?
8952 040066 005767 141212          BEQ    4$          ;; BR IF NO
8953 040072 001402          MOV    $LPERR,(SP) ;; FUDGE RETURN FOR LOOPING
8954 040074 016716 141204          TST    $ESCAPE     ;; CHECK FOR AN ESCAPE ADDRESS
8955 040100          BEQ    5$          ;; BR IF NONE
8956 040100 022737 037354 000042  MOV    $ESCAPE,(SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
          CMP    #SENDAD,@#42 ;; ACT-11 AUTO-ACCEPT?

```

```

8957 040106 001001 BNE 6$ ;;BRANCH IF NO
8958 040110 000000 HALT ;;YES
8959 040112 6$:
8960 040112 032777 001000 141020 BIT #BIT09,@SWR
8961 040120 001013 BNE ERM10
8962 040122 011637 001162 MOV (SP),@$$REG0 ;SEE IF ERROR #377
8963 040126 062737 177776 001162 ADD #-2,@$$REG0
8964 040134 122777 000377 141020 CMPB #377,@$REG0
8965 040142 001002 BNE ERM10
8966 040144 062716 000002 ADD #2,(SP)
8967 040150 000002 ERM10: RTI
8968
8969
8970 .SBTTL SAVE AND RESTORE R0-R5 ROUTINES
8971
8972 ;*****
8973 ;*SAVE R0-R5
8974 ;*CALL:
8975 ;* SAVREG
8976 ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
8977 ;*
8978 ;*TOP---(+16)
8979 ;* +2---(+18)
8980 ;* +4---R5
8981 ;* +6---R4
8982 ;* +8---R3
8983 ;*+10---R2
8984 ;*+12---R1
8985 ;*+14---R0
8986
8987 040152 $SAVREG:
8988 040152 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
8989 040154 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
8990 040156 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
8991 040160 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
8992 040162 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
8993 040164 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
8994 040166 016646 000022 MOV 22(SP),-(SP) ;;SAVE PS OF MAIN FLOW
8995 040172 016646 000022 MOV 22(SP),-(SP) ;;SAVE PC OF MAIN FLOW
8996 040176 016646 000022 MOV 22(SP),-(SP) ;;SAVE PS OF CALL
8997 040202 016646 000022 MOV 22(SP),-(SP) ;;SAVE PC OF CALL
8998 040206 000002 RTI
8999
9000 ;*RESTORE R0-R5
9001 ;*CALL:
9002 ;* RESREG
9003 040210 $RESREG:
9004 040210 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PC OF CALL
9005 040214 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PS OF CALL
9006 040220 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PC OF MAIN FLOW
9007 040224 012666 000022 MOV (SP)+,22(SP) ;;RESTORE PS OF MAIN FLOW
9008 040230 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
9009 040232 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
9010 040234 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
9011 040236 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
9012 040240 012601 MOV (SP)+,R1 ;;POP STACK INTO R1

```

9013 040242 012600  
 9014 040244 000002

MOV (SP)+,R0 ;;POP STACK INTO R0  
 RTI

9015  
 9016  
 9017

.SBTTL TYPE ROUTINE

9018  
 9019  
 9020

;;\*\*\*\*\*  
 ;\*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,  
 ;\*

9021  
 9022  
 9023

;;\*CALL:  
 ;\*1) USING A TRAP INSTRUCTION

9024  
 9025  
 9026

;\* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING

9027  
 9028  
 9029

;\*OR  
 ;\* TYPE  
 ;\* MESADR  
 ;\*

9030  
 9031  
 9032

9033 040246 105767 140705

\$TYPE; TSTB \$TPFLG ;;IS THERE A TERMINAL?

9034 040252 100002

BPL 1\$ ;;BR IF YES

9035 040254 000000

HALT ;;HALT HERE IF NO TERMINAL

9036 040256 000430

BR 3\$ ;;LEAVE

9037 040260 010046

1\$: MOV R0,-(SP) ;;SAVE R0

9038 040262 017600 000002

MOV @2(SP),R0 ;;GET ADDRESS OF ASCIZ STRING

9039 040266 122767 000001 141042

CMPB #APTENV,\$ENV ;;RUNNING IN APT MODE

9040 040274 001011

BNE 62\$ ;;NO,GO CHECK FOR APT CONSOLE

9041 040276 132767 000100 141033

BITB #APTSPOOL,\$ENVM ;;SPOOL MESSAGE TO APT

9042 040304 001405

BEQ 62\$ ;;NO,GO CHECK FOR CONSOLE

9043 040306 010067 000004

MOV R0,61\$ ;;SETUP MESSAGE ADDRESS FOR APT

9044 040312 004767 000446

JSR PC,\$ATY3 ;;SPOOL MESSAGE TO APT

9045 040316 000000

61\$: .WORD 0 ;;MESSAGE ADDRESS

9046 040320 132767 000040 141011

62\$: BITB #APTCSUP,\$ENVM ;;APT CONSOLE SUPPRESSED

9047 040326 001003

BNE 60\$ ;;YES,SKIP TYPE OUT

9048 040330 112046

2\$: MOVB (R0)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK

9049 040332 001005

BNE 4\$ ;;BR IF IT ISN'T THE TERMINATOR

9050 040334 005726

TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK

9051 040336 012600

60\$: MOV (SP)+,R0 ;;RESTORE R0

9052 040340 062716 000002

3\$: ADD #2,(SP) ;;ADJUST RETURN PC

9053 040344 000002

RTI ;;RETURN

9054 040346 122716 000011

4\$: CMPB #HT,(SP) ;;BRANCH IF <HT>

9055 040352 001430

BEQ 8\$ ;;BRANCH IF NOT <CRLF>

9056 040354 122716 000200

CMPB #CRLF,(SP)

9057 040360 001006

BNE 5\$ ;;POP <CR><LF> EQUIV

9058 040362 005726

TST (SP)+ ;;TYPE A CR AND LF

9059 040364 104401

TYPE

9060 040366 001313

\$CRLF

9061 040370 105067 000130

CLRB \$CHARCNT ;;CLEAR CHARACTER COUNT

9062 040374 000755

BR 2\$ ;;GET NEXT CHARACTER

9063 040376 004767 000056

5\$: JSR PC,\$TYPEC ;;GO TYPE THIS CHARACTER

9064 040402 126726 140550

6\$: CMPB \$FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?

9065 040406 001350

BNE 2\$ ;;IF NO GO GET NEXT CHAR.

9066 040410 016746 140540

MOV \$NULL,-(SP) ;;GET # OF FILLER CHARS, NEEDED

9067

AND THE NULL CHAR,

9068 040414 105366 000001

7\$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?

9069	040420	002770			BLT	6s		;;BR IF NO--GO POP THE NULL OFF OF STACK
9070	040422	004767	000032		JSR	PC,\$TYPEC		;;GO TYPE A NULL
9071	040426	105367	000072		DECB	\$CHARCNT		;;DO NOT COUNT AS A COUNT
9072	040432	000770			BR	7s		;;LOOP

9073  
9074 ;HORIZONTAL TAB PROCESSOR

9075								
9076	040434	112716	000040		8s:	MOVB	#' ,(SP)	;;REPLACE TAB WITH SPACE
9077	040440	004767	000014		9s:	JSR	PC,\$TYPEC	;;TYPE A SPACE
9078	040444	132767	000007	000052		BITB	#7,\$CHARCNT	;;BRANCH IF NOT AT
9079	040452	001372				BNE	9s	;;TAB STOP
9080	040454	005726				TST	(SP)+	;;POP SPACE OFF STACK
9081	040456	000724				BR	2s	;;GET NEXT CHARACTER
9082	040460	105777	140464		\$TYPEC:	TSTB	@\$TPS	;;WAIT UNTIL PRINTER IS READY
9083	040464	100375				BPL	\$TYPEC	
9084	040466	116677	000002	140456		MOVB	2(SP),@\$TPB	;;LOAD CHAR TO BE TYPED INTO DATA REG.
9085	040474	122766	000015	000002		CMPB	#CR,2(SP)	;;IS CHARACTER A CARRIAGE RETURN?
9086	040502	001003				BNE	1s	;;BRANCH IF NO
9087	040504	105067	000014			CLRB	\$CHARCNT	;;YES--CLEAR CHARACTER COUNT
9088	040510	000406				BR	\$TYPEX	;;EXIT
9089	040512	122766	000012	000002	1s:	CMPB	#LF,2(SP)	;;IS CHARACTER A LINE FEED?
9090	040520	001402				BEQ	\$TYPEX	;;BRANCH IF YES
9091	040522	105227				INCB	(PC)+	;;COUNT THE CHARACTER
9092	040524	000000			\$CHARCNT:	WORD	0	;;CHARACTER COUNT STORAGE
9093	040526	000207			\$TYPEX:	RTS	PC	

9094  
9095  
9096 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE

9097  
9098 ;\*\*\*\*\*  
9099 ;\*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT  
9100 ;\*OCTAL (ASCII) NUMBER AND TYPE IT.  
9101 ;\*\$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE  
9102 ;\*CALL:  
9103 ;\* MOV NUM,-(SP) ;;NUMBER TO BE TYPED  
9104 ;\* TYPOS ;;CALL FOR TYPEOUT  
9105 ;\* .BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE  
9106 ;\* .BYTE M ;;M=1 OR 0  
9107 ;\* ;;1=TYPE LEADING ZEROS  
9108 ;\* ;;0=SUPPRESS LEADING ZEROS  
9109 ;\*  
9110 ;\*\$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST  
9111 ;\*\$TYPOS OR \$TYPOC  
9112 ;\*CALL:  
9113 ;\* MOV NUM,-(SP) ;;NUMBER TO BE TYPED  
9114 ;\* TYPON ;;CALL FOR TYPEOUT  
9115 ;\*  
9116 ;\*\$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER  
9117 ;\*CALL:  
9118 ;\* MOV NUM,-(SP) ;;NUMBER TO BE TYPED  
9119 ;\* TYPOC ;;CALL FOR TYPEOUT

9120								
9121	040530	017646	000000		\$TYPOS:	MOV	@(SP),-(SP)	;;PICKUP THE MODE
9122	040534	116667	000001	000211		MOVB	1(SP),s0FILL	;;LOAD ZERO FILL SWITCH
9123	040542	112667	000207			MOVB	(SP)+,\$OMODE+1	;;NUMBER OF DIGITS TO TYPE
9124	040546	062716	000002			ADD	#2,(SP)	;;ADJUST RETURN ADDRESS

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

```

9181 041002          SATYC:
9182 041002 010046      MOV     R0,-(SP)      ;;PUSH R0 ON STACK
9183 041004 010146      MOV     R1,-(SP)      ;;PUSH R1 ON STACK
9184 041006 105767 000206  TSTB   $MFLG         ;;SHOULD TYPE A MESSAGE?
9185 041012 001450      BEQ     5$            ;;IF NOT: BR
9186 041014 122767 000001 140314  CMPB   #APTENV,$ENV  ;;OPERATING UNDER APT?
9187 041022 001031      BNE     3$            ;;IF NOT: BR
9188 041024 132767 000100 140305  BITB   #APTSPool,$ENVM ;;SHOULD SPOOL MESSAGES?
9189 041032 001425      BEQ     3$            ;;IF NOT: BR
9190 041034 017600 000004      MOV     04(SP),R0    ;;GET MESSAGE ADDR.
9191 041040 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
9192 041046 005767 140244      TST    $MSGTYPE     ;;SEE IF DONE W/ LAST XMISSION?
9193 041052 001375      BNE     1$            ;;IF NOT: WAIT
9194 041054 010067 140252  MOV     R0,$MSGAD    ;;PUT ADDR IN MAILBOX
9195 041060 105720      TSTB   (R0)+         ;;FIND END OF MESSAGE
9196 041062 001376      BNE     2$
9197 041064 166700 140242  SUB     $MSGAD,R0    ;;SUB START OF MESSAGE
9198 041070 006200      ASR     R0            ;;GET MESSAGE LNTH IN WORDS
9199 041072 010067 140236  MOV     R0,$MSGLGT   ;;PUT LENGTH IN MAILBOX
9200 041076 012767 000004 140212  MOV     #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
9201 041104 000413      BR      5$
9202 041106 017667 000004 000016 3$:    MOV     04(SP),4$    ;;PUT MSG ADDR IN JSR LINKAGE
9203 041114 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDRESS
9204 041122 016746 136650      MOV     177776,-(SP) ;;PUSH 177776 ON STACK
9205 041126 004767 177114  JSR    PC,$TYPE     ;;CALL TYPE MACRO
9206 041132 000000      4$:    .WORD 0
9207 041134      5$:
9208 041134 105767 000062      10$:   TSTB   $FFLG         ;;SHOULD REPORT FATAL ERROR?
9209 041140 001416      BEQ     12$          ;;IF NOT: BR
9210 041142 005767 140170      TST    $ENV         ;;RUNNING UNDER APT?
9211 041146 001413      BEQ     12$          ;;IF NOT: BR
9212 041150 005767 140142      11$:   TST    $MSGTYPE     ;;FINISHED LAST MESSAGE?
9213 041154 001375      BNE     11$          ;;IF NOT: WAIT
9214 041156 017667 000004 140134  MOV     04(SP),$FATAL ;;GET ERROR #
9215 041164 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
9216 041172 005267 140120      INC    $MSGTYPE     ;;TELL APT TO TAKE ERROR
9217 041176 105067 000020      12$:   CLRB   $FFLG         ;;CLEAR FATAL FLAG
9218 041202 105067 000013      CLRB   $LFLG        ;;CLEAR LOG FLAG
9219 041206 105067 000006      CLRB   $MFLG        ;;CLEAR MESSAGE FLAG
9220 041212 012601      MOV     (SP)+,R1    ;;POP STACK INTO R1
9221 041214 012600      MOV     (SP)+,R0    ;;POP STACK INTO R0
9222 041216 000207      RTS    PC           ;;RETURN
9223 041220 000      $MFLG: .BYTE 0      ;;MESSG, FLAG
9224 041221 000      $LFLG: .BYTE 0      ;;LOG FLAG
9225 041222 000      $FFLG: .BYTE 0      ;;FATAL FLAG
9226 041224      .EVEN
9227 000200      APTSIZE=200
9228 000001      APTENV=001
9229 000100      APTSPool=100
9230 000040      APTCSUP=040
9231
9232      .SBTTL TTY INPUT ROUTINE
9233
9234      ;;*****
9235      .ENABL LSB
9236
    
```

```

9237 ;*****
9238 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE,
9239 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
9240 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
9241 ;*WHEN OPERATING IN TTY FLAG MODE.
9242 041224 022767 000176 137706 $CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
9243 041232 001074 BNE 15$ ;;BRANCH IF NO
9244 041234 105777 137704 TSTB @STKS ;;CHAR THERE?
9245 041240 100071 BPL 15$ ;;IF NO, DON'T WAIT AROUND
9246 041242 117746 137700 MOVB @STKB,-(SP) ;;SAVE THE CHAR
9247 041246 042716 177600 BIC #C177,(SP) ;;STRIP-OFF THE ASCII
9248 041252 022726 000007 CMP #7,(SP)+ ;;IS IT A CONTROL G?
9249 041256 001062 BNE 15$ ;;NO, RETURN TO USER
9250 041260 126727 137650 000001 CMPB $AUTOB,#1 ;;ARE WE RUNNING IN AUTO-MODE?
9251 041266 001456 BEQ 15$ ;;BRANCH IF YES
9252
9253 041270 104401 041633 TYPE ,SCNTLG ;;ECHO THE CONTROL-G (^G)
9254 041274 104401 041640 $GTSWR: TYPE ,SMSWR ;;TYPE CURRENT CONTENTS
9255 041300 016746 136672 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT
9256 041304 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
9257 041306 104401 041651 TYPE ,SMNEW ;;PROMPT FOR NEW SWR
9258 041312 005046 19$: CLR -(SP) ;;CLEAR COUNTER
9259 041314 005046 CLR -(SP) ;;THE NEW SWR
9260 041316 105777 137622 7$: TSTB @STKS ;;CHAR THERE?
9261 041322 100375 BPL 7$ ;;IF NOT TRY AGAIN
9262
9263 041324 117746 137616 MOVB @STKB,-(SP) ;;PICK UP CHAR
9264 041330 042716 177600 BIC #C177,(SP) ;;MAKE IT 7-BIT ASCII
9265
9266
9267
9268 041334 021627 000025 9$: CMP (SP),#25 ;;IS IT A CONTROL-U?
9269 041340 001005 BNE 10$ ;;BRANCH IF NOT
9270 041342 104401 041626 TYPE ,SCNTLU ;;YES, ECHO CONTROL-U (^U)
9271 041346 062706 000006 20$: ADD #6,SP ;;IGNORE PREVIOUS INPUT
9272 041352 000757 BR 19$ ;;LET'S TRY IT AGAIN
9273
9274
9275 041354 021627 000015 10$: CMP (SP),#15 ;;IS IT A <CR>?
9276 041360 001022 BNE 16$ ;;BRANCH IF NO
9277 041362 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
9278 041366 001403 BEQ 11$ ;;BRANCH IF YES
9279 041370 016677 000002 137542 MOV 2(SP),@SWR ;;SAVE NEW SWR
9280 041376 062706 000006 11$: ADD #6,SP ;;CLEAR UP STACK
9281 041402 104401 001313 14$: TYPE ,SCRLF ;;ECHO <CR> AND <LF>
9282 041406 126727 137523 000001 CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?
9283 041414 001003 BNE 15$ ;;BRANCH IF NOT
9284 041416 012777 000100 137520 MOV #100,@STKS ;;RE-ENABLE TTY KBD INTERRUPTS
9285 041424 000002 15$: RTI ;;RETURN
9286 041426 004767 177026 16$: JSR PC,$TYPEC ;;ECHO CHAR
9287 041432 021627 000060 CMP (SP),#60 ;;CHAR < 0?
9288 041436 002420 BLT 18$ ;;BRANCH IF YES
9289 041440 021627 000067 CMP (SP),#67 ;;CHAR > 7?
9290 041444 003015 BGT 18$ ;;BRANCH IF YES
9291 041446 042726 000060 BIC #60,(SP)+ ;;STRIP-OFF ASCII
9292 041452 005766 000002 TST 2(SP) ;;IS THIS THE FIRST CHAR

```

```

9293 041456 001403          BEQ      17s          ;:BRANCH IF YES
9294 041460 006316          ASL      (SP)         ;:NO. SHIFT PRESENT
9295 041462 006316          ASL      (SP)         ;:  CHAR OVER TO MAKE
9296 041464 006316          ASL      (SP)         ;:  ROOM FOR NEW ONE.
9297 041466 005266 000002    17s:    INC      2(SP)       ;:KEEP COUNT OF CHAR
9298 041472 056616 177776    BIS      =2(SP),(SP) ;:SET IN NEW CHAR
9299 041476 000707          BR       7s          ;:GET THE NEXT ONE
9300 041500 104401 001312    18s:    TYPE     ,SQUES    ;:TYPE ?<CR><LF>
9301 041504 000720          BR      20s         ;:SIMULATE CONTROL-U
9302          .DSABL  LSB
9303
9304
9305          ;:*****
9306          ;:THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
9307          ;:CALL:
9308          ;*      RDCHR          ;:INPUT A SINGLE CHARACTER FROM THE TTY
9309          ;*      RETURN HERE   ;:CHARACTER IS ON THE STACK
9310          ;*
9311          ;
9312
9313 041506 011646          SRDCHR: MOV      (SP),-(SP) ;:PUSH DOWN THE PC
9314 041510 016666 000004 000002 MOV      4(SP),2(SP) ;:SAVE THE PS
9315 041516 105777 137422    1s:    TSTB     @STKS      ;:WAIT FOR
9316 041522 100375          BPL      1s          ;:A CHARACTER
9317 041524 117766 137416 000004 MOVB     @STKB,4(SP) ;:READ THE TTY
9318 041532 042766 177600 000004 BIC      #'C<177>,4(SP) ;:GET RID OF JUNK IF ANY
9319 041540 026627 000004 000023 CMP      4(SP),#23    ;:IS IT A CONTROL-S?
9320 041546 001013          BNE      3s          ;:BRANCH IF NO
9321 041550 105777 137370    2s:    TSTB     @STKS      ;:WAIT FOR A CHARACTER
9322 041554 100375          BPL      2s          ;:LOOP UNTIL ITS THERE
9323 041556 117746 137364 MOVB     @STKB,-(SP) ;:GET CHARACTER
9324 041562 042716 177600 BIC      #'C177,(SP) ;:MAKE IT 7-BIT ASCII
9325 041566 022627 000021 CMP      (SP)+,#21    ;:IS IT A CONTROL-Q?
9326 041572 001366          BNE      2s          ;:IF NOT DISCARD IT
9327 041574 000750          BR       1s          ;:YES, RESUME
9328 041576 026627 000004 000140 3s:    CMP      4(SP),#140 ;:IS IT UPPER CASE?
9329 041604 002407          BLT      4s          ;:BRANCH IF YES
9330 041606 026627 000004 000175 CMP      4(SP),#175 ;:IS IT A SPECIAL CHAR?
9331 041614 003003          BGT      4s          ;:BRANCH IF YES
9332 041616 042766 000040 000004 BIC      #40,4(SP)   ;:MAKE IT UPPER CASE
9333 041624 000002          4s:    RTI
9334 041626 052536 005015 000      $CNTLU: ,ASCIZ /'U/<15><12> ;:CONTROL "U"
9335 041633 136 006507 000012 $CNTLG: ,ASCIZ /'G/<15><12> ;:CONTROL "G"
9336 041640 005015 053523 020122 SMSWR:  ,ASCIZ <15><12>/SWR = /
9337 041646 020075 000
9338 041651 040 047040 053505 $MNEW:  ,ASCIZ / NEW = /
9339 041656 036440 000040

```

```

9340
9341          .SBTTL  TRAP DECODER
9342

```

```

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

```



TRAP DECODER

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

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

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

9360 041704 011646  
9361 041706 016666 000004 000002  
9362 041714 000002  
9363  
9364  
9365  
9366  
9367  
9368  
9369  
9370

\$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.

9371 041716 041704  
9372 041720 040246  
9373 041722 040554  
9374 041724 040530  
9375 041726 040570  
9376  
9377 041730 041274  
9378  
9379 041732 041224  
9380 041734 041506  
9381 041736 040152  
9382 041740 040210  
9383 041742 042666  
9384 041744 042660  
9385 000030  
9386  
9387  
9388  
9389  
9390

: ROUTINE  
:-----  
\$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 TES  
\$LPER ;:CALL=LPER TRAP+13(104413) ROUTINE TO SET UP LOOP ON ERROR ADDRESS

\$TERM=.-\$TRPAD

.SBTTL POWER DOWN AND UP ROUTINES

:;\*\*\*\*\*

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

:POWER DOWN ROUTINE  
\$PWRDN: MOV #SILLUP,#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 #SPWRUP,#PWRVEC ;:SET UP VECTOR  
HALT  
BR .-2 ;:HANG UP

```

9405 ;*****
9406 ;POWER UP ROUTINE
9407 042020 012737 042124 000024 $PWRUP: MOV $SILLUP,@#PWRVEC ;SET FOR FAST DOWN
9408 042026 016706 000076 MOV $SAVR6,SP ;GET SP
9409 042032 005067 000072 CLR $SAVR6 ;WAIT LOOP FOR THE TTY
9410 042036 005267 000066 1s: INC $SAVR6 ;WAIT FOR THE INC
9411 042042 001375 BNE 1s ;OF WORD
9412 042044 012677 137070 MOV (SP)+,@SWR ;POP STACK INTO @SWR
9413 042050 012605 MOV (SP)+,R5 ;POP STACK INTO R5
9414 042052 012604 MOV (SP)+,R4 ;POP STACK INTO R4
9415 042054 012603 MOV (SP)+,R3 ;POP STACK INTO R3
9416 042056 012602 MOV (SP)+,R2 ;POP STACK INTO R2
9417 042060 012601 MOV (SP)+,R1 ;POP STACK INTO R1
9418 042062 012600 MOV (SP)+,R0 ;POP STACK INTO R0
9419 042064 012737 041746 000024 MOV $SPWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR
9420 042072 012737 000340 000026 MOV #340,@#PWRVEC+2 ;PRIO:7
9421 042100 104401 TYPE ;REPORT THE POWER FAILURE
9422 042102 042736 $PWRMG: .WORD POWERM ;POWER FAIL MESSAGE POINTER
9423 042104 012716 MOV (PC)+,(SP) ;RESTART AT START
9424 042106 006106 $PWRAD: .WORD START ;RESTART ADDRESS
9425 042110 042766 000020 000002 BIC #20,2(SP) ;CLEAR "T" BIT
9426 042116 005067 175304 CLR $TBIT ;CLEAR THE "T" BIT FLAG
9427 042122 000002 RTI
9428 042124 000000 $SILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
9429 042126 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
9430 042130 000000 $SAVR6: 0 ;PUT THE SP HERE

```

```

9431
9432
9433 ;SBTTL ERROR TYPE OUT ROUTINE
9434 ;*****
9435 ;*****
9436 ;*THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED
9437 ;*IN THE ERROR MESSAGE DATA TABLE, IT IS CALLED BY THE $ERROR ROUTINE
9438 ;*OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED
9439 ;*OUT AND THEN EXECUTING A;
9440 ;* JSR PC,ERTYPE
9441 ;*
9442 042132 104401 ERTYPE: TYPE ;TYPE A CRLF
9443 042134 001313 .WORD $CRLF
9444 042136 113737 001102 001232 MOVB @#$STSTNM,@#$TMP0
9445 042144 042737 177400 001232 BIC #177400,@#$TMP0
9446 042152 013737 001116 001234 MOV @#$ERRPC,@#$TMP1 ;GET PC OF CALL
9447 042160 010046 MOV R0,-(SP) ;SAVE R0
9448
9449 042162 113700 001114 MOVB @#$ITEMB,R0 ;GET THE ITEM NUMBER,
9450 042166 042700 177400 BIC #177400,R0
9451 042172 001005 BNE 1s
9452
9453 042174 013746 001116 MOV @#$ERRPC,-(SP) ;IF ZERO THEN JUST
9454 042200 104402 TYPOC ;PRINT THE PC
9455 042202 000137 042560 JMP @#ERT5
9456
9457 042206 022700 000377 1s: CMP #377,R0
9458 042212 001005 BNE 20s
9459 042214 016600 000004 MOV 4(SP),R0
9460 042220 011000 MOV (R0),R0

```

9461	042222	062700	000400		ADD	#400,R0	
9462	042226	005300		20s:	DEC	R0	;OTHERWISE MAKE R0 AN
9463	042230	006300			ASL	R0	;INDEX FOR THE TABLE,
9464	042232	006300			ASL	R0	
9465	042234	006300			ASL	R0	
9466	042236	062700	001442		ADD	#\$ERRTB,R0	
9467							
9468	042242	012037	042252		MOV	(R0)+,0#2\$	;PICK UP THE ADDRESS
9469	042246	001404			BEQ	3\$	;OF THE EM, ERROR MESSAGE
9470	042250	104401			TYPE		
9471	042252	000000		2s:	.WORD	0	
9472	042254	104401			TYPE		
9473	042256	001313			.WORD	\$CRLF	
9474							
9475	042260	012037	042270	3s:	MOV	(R0)+,0#4\$	;GET THE DH,DATA HEADER
9476	042264	001404			BEQ	5\$	
9477	042266	104401			TYPE		
9478	042270	000000		4s:	.WORD	0	
9479	042272	104401			TYPE		
9480	042274	001313			.WORD	\$CRLF	
9481							
9482	042276	010146		5s:	MOV	R1,-(SP)	;SAVE R1,R2 AND R3
9483	042300	010246			MOV	R2,-(SP)	
9484	042302	010346			MOV	R3,-(SP)	
9485							
9486	042304	012001			MOV	(R0)+,R1	;GET THE ADDRESS OF THE
9487							;DATA TABLE,
9488	042306	001001			BNE	6\$	
9489	042310	000516			BR	ERT4	;RETURN IF NO DATA,
9490							
9491	042312	011000		6s:	MOV	(R0),R0	;GET A POINTER TO THE DATA
9492							;FORMAT TABLE,
9493	042314	105710		ERT1:	TSTB	(R0)	;FORMAT ZERO?
9494	042316	001003			BNE	7\$	
9495							
9496	042320	013146			MOV	@(R1)+,-(SP)	;FORMAT ZERO SO TYPE
9497	042322	104402			TYPOC		;AN OCTAL NUMBER,
9498	042324	000502			BR	ERT2	
9499							
9500	042326			7s:			
9501	042326	122710	000002	8s:	CMPB	#2,(R0)	;FORMAT TWO?
9502	042332	001010			BNE	9\$	
9503							
9504	042334	013102			MOV	@(R1)+,R2	;FORMAT TWO SO TYPE TWO
9505	042336	012246			MOV	(R2)+,-(SP)	;OCTAL NUMBERS,
9506	042340	104402			TYPOC		
9507	042342	104401			TYPE		
9508	042344	043002			.WORD	SPACE	
9509	042346	011246			MOV	(R2),-(SP)	
9510	042350	104402			TYPOC		
9511	042352	000467			BR	ERT2	
9512							
9513	042354	122710	000003	9s:	CMPB	#3,(R0)	;FORMAT THREE?
9514	042360	001020			BNE	10\$	
9515							
9516	042362	013102			MOV	@(R1)+,R2	;FORMAT THREE SO TYPE

9517	042364	012246			MOV	(R2)+, -(SP)				
9518	042366	104402			TYPOC					
9519	042370	104401			TYPE					
9520	042372	043002			.WORD	SPACE				
9521	042374	012246			MOV	(R2)+, -(SP)				
9522	042376	104402			TYPOC					
9523	042400	104401			TYPE					
9524	042402	043002			.WORD	SPACE				
9525	042404	012246			MOV	(R2)+, -(SP)				
9526	042406	104402			TYPOC					
9527	042410	104401			TYPE					
9528	042412	043002			.WORD	SPACE				
9529	042414	011246			MOV	(R2), -(SP)				
9530	042416	104402			TYPOC					
9531	042420	000444			BR	ERT2				
9532										
9533	042422	122710	000004	10s:	CMPB	#4, (R0)				;FORMAT FOUR?
9534	042426	001004			BNE	11s				
9535										
9536	042430	013146			MOV	@(R1)+, -(SP)				;FORMAR FOUR SO TYPE
9537	042432	104403			TYPOS					;AN OCTAL NUMBER
9538	042434	016			.BYTE	16				;SUPPRESSING LEADING ZEROES,
9539	042435	000			.BYTE	0				
9540	042436	000435			BR	ERT2				
9541										
9542	042440	122710	000005	11s:	CMPB	#5, (R0)				;FORMAT FIVE?
9543	042444	001005			BNE	13s				
9544										
9545	042446	012137	042454		MOV	(R1)+, @#12s				;FORMAT FIVE SO TYPE AN
9546	042452	104401			TYPE					;ASCIZ STRING,
9547	042454	000000		12s:	.WORD	0				
9548	042456	000427			BR	ERT3				
9549										
9550	042460	122710	000011	13s:	CMPB	#11, (R0)				;FORMAT ELEVEN?
9551	042464	001005			BNE	15s				
9552										
9553	042466	013137	042474		MOV	@(R1)+, @#14s				;FORMAT ELEVEN SO PICK
9554	042472	104401			TYPE					;A POINTER TO AN ASCIZ
9555	042474	000000		14s:	.WORD	0				;STRING,
9556	042476	000417			BR	ERT3				
9557										
9558	042500	122710	000012	15s:	CMPB	#12, (R0)				;FORMAT TWELVE?
9559	042504	001011			BNE	17s				
9560										
9561	042506	013102			MOV	@(R1)+, R2				;FORMAT TWELVE SO TYPE
9562	042510	012703	000006		MOV	#6, R3				;TYPE SIX OCTAL NUMBERS
9563	042514	012246		16s:	MOV	(R2)+, -(SP)				
9564	042516	104402			TYPOC					
9565	042520	104401			TYPE					
9566	042522	043002			.WORD	SPACE				
9567	042524	077305			SOB	R3, 16s				
9568	042526	000401			BR	ERT2				
9569										
9570	042530	000000		17s:	HALT					;UNDEFINED FORMAT FOR DATA?????
9571										
9572	042532	104401		ERT2:	TYPE					;PRINT A TAB AFTER TYPING

```

9573 042534 043005 .WORD $TAB ;AN DATA TABLE ENTRY
9574 ;OF ALL FORMATS EXCEPT
9575 ;ASCIZ, FORMATS 5 OR 11
9576
9577 042536 005200 ERT3: INC R0 ;POINT TO THE NEXT FORMAT
9578 042540 005711 TST (R1) ;END OF DATA TABLE,
9579 042542 001401 BEQ ERT4
9580 042544 000663 BR ERT1
9581
9582 042546 104401 ERT4: TYPE ;DONE,
9583 042550 001313 .WORD $CRLF
9584 042552 012603 MOV (SP)+,R3 ;RESTORE R1,R2 AND R3
9585 042554 012602 MOV (SP)+,R2
9586 042556 012601 MOV (SP)+,R1
9587 042560 012600 ERT5: MOV (SP)+,R0 ;RESTORE R0.
9588 042562 000207 RTS PC ;AND RETURN,
9589
9590
9591
9592

```

.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.
;*
FPSPUR: MOV (SP),@#$TMP2 ;SAVE PC OF TRAP.
        CMP (SP)+,(SP)+ ;RESTORE SP,
        STFPS R0 ;GET FPS
        MOV R0,@#$TMP3
        STST R0 ;GET FEC
        MOV R0,@#$TMP4
1$: ERROR 377
        .WORD 441
        RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
                ;SEE IF THE USER HAS EXPRESSED
                ;THE DESIRE TO CHANGE THE SOFTWARE
                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                ;THE USER TYPED CONTROL G?).
        JMP @#$EOP

```

.SBTTL CPU SPURIOUS TRAP TO 4 HANDLER

```

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

```

```

9629                                     ;THE USER TYPED CONTROL G?).
9630 042634 000137 037142                JMP     @#SEOP
9631
9632
9633                                     .SBTTL CPU SPURIOUS TRAP TO 10 HANDLER
9634 ;|*****
9635 ;|*****
9636 ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
9637 ;*
9638 042640 011637 001236                CPTWO:  MOV     (SP),@#STMP2                ;SAVE PC OF TRAP.
9639 042644 022626                        CMP     (SP)+,(SP)+
9640 042646 104377                        1S:    ERROR  377
9641 042650 000443                        .WORD  443
9642 042652 104412                        RSETUP                                     ;GO INITIALIZE THE FPS AND STACK; AND
9643                                     ;SEE IF THE USER HAS EXPRESSED
9644                                     ;THE DESIRE TO CHANGE THE SOFTWARE
9645                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
9646                                     ;THE USER TYPED CONTROL G?).
9647 042654 000137 037142                JMP     @#SEOP
9648
9649
9650
9651
9652
9653                                     .SBTTL SET LOOP ON ERROR ADDRESS ROUTINE
9654 ;|*****
9655 ;|*****
9656 ;*
9657 042660 011637 001110                .LPER:  MOV     (SP),@#SLPERR
9658 042664 000002                        RTI
9659
9660                                     .SBTTL FLAG RESET AND CONSOLE TEST ROUTINE
9661 ;|*****
9662 ;|*****
9663 ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
9664 ;*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
9665 ;* CONTROL G ON THE TERMINAL, IF THE USER HAS TYPED CONTROL G AND
9666 ;*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
9667 ;*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
9668 ;*TELETYPE AND THE USER CAN MODIFY IT.
9669 ;*
9670 042666 023727 001140 177570        .RSET:  CMP     @#SWR,#177570                ;SEE IF THERE IS A PHYSICAL
9671                                     ;CONSOLE SWITCH REGISTER.
9672 042674 001001                        BNE     1S                                  ;BRANCH IF NO.
9673 042676 104406                        CKSWR                                     ;OTHERWISE TYPE THE CONTENTS
9674                                     ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
9675                                     ;AND GIVE THE USER A CHANCE TO
9676                                     ;MODIFY IT.
9677 042700 012737 042564 000244        1S:    MOV     #FPSPUR,@#FPVECT
9678 042706 012737 042620 000004        MOV     #CPSPUR,@#ERRVECT
9679 042714 012737 042640 000010        MOV     #CPTWO,@#10
9680 042722 011600                        MOV     (SP),R0                            ;SAVE RETURN ADDRESS.
9681 042724 012706 001100                MOV     #STACK,SP                          ;RESET THE STACK POINTER.
9682 042730 005004                        CLR     R4                                  ;CLEAR THE FPS.
9683 042732 170104                        LDFPS  R4
9684 042734 000110                        JMP     (R0)                                ;RETURN.

```

9685  
9686  
9687

.NLIST BEX

;THESE ARE SPECIAL MESSAGES:

042736	050200	053517	051105	POWERM:	.ASCIZ	<CRLF>'POWER FAILURE. PROGRAM RESTARTING.'
043002	020040	000		SPACE:	.ASCIZ	' '
043005	011	000		STAB:	.ASCIZ	<TAB>
043007	107	052117	051040	MS1:	.ASCIZ	'GOT RESULT:'<TAB><TAB>
043025	105	050130	041505	MS2:	.ASCIZ	'EXPECTED RESULT:'<TAB>
043047	101	020103	050117	MS3:	.ASCIZ	'AC OPERAND:'<TAB><TAB>
043065	123	052517	041522	MS4:	.ASCIZ	'SOURCE OPERAND:'<TAB>
	043047			MS10=MS3		
043107	105	050130	047117	MS11:	.ASCIZ	'EXPONENT OPERAND:'<TAB>

;THESE ARE ERROR MESSAGES:

	043132	052123	020106	026101	EM1:	.ASCIZ	'STF A,AC7 DID NOT TRAP. FID=0.'
	043171	123	043124	040440	EM2:	.ASCIZ	'STF A,AC7. FPS BAD. FID=0.'
	043224	052123	020106	026101	EM3:	.ASCIZ	'STF A,AC7. FEC BAD. FID=0.'
(0)	043257				EM4:		
(1)	043257	123	043124	040440		.ASCIZ	\STF A,(R). R0 BAD. FDST FAILED,\
(0)	043317				EM5:		
(1)	043317	123	043124	040440		.ASCII	\STF A,(R) FAILED,\
	043340	000				.BYTE	0
(0)	043341				EM6:		
(1)	043341	123	043124	040440		.ASCII	\STF A,(R). FDST FAILED,\
(1)	043370	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FD) ST 707 WENT TO 245 INSTEAD OF 244,\
(0)	043445				EM7:		
(1)	043445	123	043124	040440		.ASCIZ	\STF A,(R)+. R0 BAD. FDST FAILED,\
(0)	043506				EM10:		
(1)	043506	052123	020106	026101		.ASCII	\STF A,(R)+ FAILED,\
	043530	000				.BYTE	0
(0)	043531				EM11:		
(1)	043531	123	042124	040440		.ASCIZ	\STD A,(R)+. R0 BAD. FDST FAILED,\
(0)	043572				EM12:		
(1)	043572	052123	020104	026101		.ASCII	\STD A,(R)+ FAILED,\
	043614	000				.BYTE	0
	043615	123	042124	040440	EM13:	.ASCIZ	'STD A,#N TRAP TO 4 IN FDST.'
		043615			EM14=EM13		
(0)	043651				EM15:		
(1)	043651	123	042124	040440		.ASCII	\STD A,#N FAILED,\
	043671	000				.BYTE	0
	043672	041520	041040	042101	EM16:	.ASCIZ	'PC BAD AFTER STD A,#N.'
(0)	043721				EM17:		
(1)	043721	123	042124	040440		.ASCIZ	\STD A,-(R) TRAP TO 4 IN FDST,\
(0)	043757				EM20:		
(1)	043757	123	042124	040440		.ASCIZ	\STD A,-(R). R0 BAD. FDST FAILED,\
(0)	044020				EM21:		
(1)	044020	052123	020104	026101		.ASCII	\STD A,-(R) FAILED,\
	044042	000				.BYTE	0
		044020			EM22=EM21		
(0)	044043				EM23:		
(1)	044043	123	042124	040440		.ASCIZ	\STD A,@(R)+ TRAP TO 4 IN FDST,\

(0)	044102				EM24:		
(1)	044102	052123	020104	026101		.ASCIZ	\STD A,@(R)+, R0 BAD, FDST FAILED,\
(0)	044144				EM25:		
(1)	044144	052123	020104	026101		.ASCII	\STD A,@(R)+ FAILED,\
(0)	044167	000				.BYTE	0
(0)	044170				EM26:		
(1)	044170	052123	020104	026101		.ASCIZ	\STD A,@-(R) TRAP TO 4 IN FDST,\
(0)	044227				EM27:		
(1)	044227	123	042124	040440		.ASCIZ	\STD A,@-(R), R0 BAD, FDST FAILED,\
(0)	044271				EM30:		
(1)	044271	123	042124	040440		.ASCII	\STD A,@-(R) FAILED,\
(0)	044314	000				.BYTE	0
(0)	044315				EM31:		
(1)	044315	123	042124	040440		.ASCIZ	\STD A,N(R) TRAP TO 4 IN FDST,\
(0)	044353				EM32:		
(1)	044353	123	042124	040440		.ASCIZ	\STD A,N(R), R0 BAD, FDST FAILED,\
(0)	044414				EM33:		
(1)	044414	052123	020104	026101		.ASCII	\STD A,N(R) FAILED,\
(0)	044436	000				.BYTE	0
(0)	044437				EM34:		
(1)	044437	123	042124	040440		.ASCIZ	\STD A,@N(R) TRAP TO 4 IN FDST,\
(0)	044476				EM35:		
(1)	044476	052123	020104	026101		.ASCIZ	\STD A,@N(R), R0 BAD, FDST FAILED,\
(0)	044540				EM36:		
(1)	044540	052123	020104	026101		.ASCII	\STD A,@N(R) FAILED,\
(0)	044563	000				.BYTE	0
(0)	044564				EM37:		
(1)	044564	052123	043103	020104		.ASCII	'STCFD A,(R) FAILED,'
(0)	044607	000				.BYTE	0
(0)	044610				EM40:		
(1)	044610	052123	043103	020104		.ASCII	\STCFD A,(R), FPS BAD,\
(0)	044635	000				.BYTE	0
(0)	044636				EM41:		
(1)	044636	052123	043103	020104		.ASCII	\STCFD A,(R), FEC BAD,\
(0)	044663	000				.BYTE	0
(0)	044664				EM42:		
(1)	044664	052123	043103	020104		.ASCII	'STCFD A,(R) FAILED,'
(0)	044707	200	047111	042526		.ASCIZ	<CRLF>'INVERT FDFL ST 767 FAILED,'
(0)	044743				EM43:		
(1)	044743	123	041524	042106		.ASCII	\STCFD A,(R), FPS BAD,\
(1)	044770	024200	052502	020124		.ASCIZ	<CRLF>\(BUT EZBT) ST 560 WENT TO 061 INSTEAD OF 261,\
(0)	045047				EM44:		
(1)	045047	123	041524	042106		.ASCII	'STCFD A,(R) FAILED,'
(0)	045072	046200	053517	047440		.ASCIZ	<CRLF>'LOW ORDER BITS OF X11 DID NOT GET 0 ST 766,'
(0)	045147				EM45:		
(1)	045147	123	041524	042106		.ASCII	'STCFD A,(R) FAILED,'
(0)	045172	024200	052502	020124		.ASCIZ	<CRLF>'(BUT OPIC) ST 251 FAILED,'
(0)	045225				EM46:		
(1)	045225	123	041524	042106		.ASCII	\STCFD A,(R), FPS BAD,\
(1)	045252	024200	052502	020124		.ASCIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 262 INSTEAD OF 062,\
(0)	045331				EM47:		
(1)	045331	123	041524	042106		.ASCII	'STCFD A,(R) FAILED,'
(1)	045354	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FD) ST 113 WENT TO 415 INSTEAD OF 414,\
(0)	045431				EM50:		
(1)	045431	123	041524	042106		.ASCII	'STCFD A,(R) FAILED,'
(0)	045454	051440	043511	020116		.ASCII	' SIGN BAD,'



(1)	045466	024200	052502	020124		.ASCIZ	<CRLF>\(BUT ENBT) ST 567 WENT TO 060 INSTEAD OF 460.\
(0)	045545				EM51:		
(1)	045545	123	041524	043104		.ASCII	'STCDF A,(R) FAILED.'
(0)	045570	000				.BYTE	0
(0)	045571				EM52:		
(1)	045571	123	042124	040440		.ASCII	\STD A,(R). FPS BAD.\
(0)	045614	000				.BYTE	0
(0)	045615				EM53:		
(1)	045615	123	042124	040440		.ASCII	\STD A,(R). FEC BAD.\
(0)	045640	000				.BYTE	0
(0)	045641				EM54:		
(1)	045641	123	041524	043104		.ASCII	'STCDF A,(R) FAILED.'
(0)	045664	044600	053116	051105		.ASCIZ	<CRLF>'INVERT FDFL ST 767 FAILED.'
(0)	045720				EM55:		
(1)	045720	052123	042103	020106		.ASCII	'STCDF A,(R) FAILED.'
(0)	045743	200	047522	047125		.ASCII	<CRLF>'ROUND ERROR, OR'
(1)	045763	200	041050	052125		.ASCIZ	<CRLF>\(BUT BREAKOUT) ST 400 WENT TO 766 INSTEAD OF 767.\
(0)	046046				EM56:		
(1)	046046	052123	020104	026101		.ASCII	\STD A,(R). FPS BAD.\
(1)	046071	200	041050	052125		.ASCIZ	<CRLF>\(BUT EZBT) ST 421 WENT TO 062 INSTEAD OF 262.\
(0)	046150				EM57:		
(1)	046150	052123	020104	026101		.ASCII	\STD A,(R). FPS BAD.\
(0)	046173	040	044506	036526		.ASCII	' FIV=0.'
(1)	046202	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 123 INSTEAD OF 103.\
(0)	046260				EM60:		
(1)	046260	052123	042103	020106		.ASCII	'STCDF A,(R) FAILED.'
(0)	046303	040	044506	036526		.ASCII	' FIV=1.'
(1)	046312	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FIV) ST 262 WENT TO 103 INSTEAD OF 123.\
(0)	046370				EM61:		
(1)	046370	052123	020104	026101		.ASCII	\STD A,(R). FPS BAD.\
(1)	046413	200	041050	052125		.ASCIZ	<CRLF>\(BUT FLAG) ST 147 WENT TO 361 INSTEAD OF 365.\
(0)	046472	052123	043103	020104	EM62:	.ASCII	'STCFD A,AC6. FPS BAD.'
(1)	046517	200	041050	052125		.ASCIZ	<CRLF>\(BUT FDST) ST 767 WENT TO 567 INSTEAD OF 577.\
(0)	046576	052123	043103	020104	EM63:	.ASCIZ	'STCFD A,AC6. FEC BAD.'
(0)	046624				EM64:		
(1)	046624	046103	042122	024040		.ASCII	\CLRD (R) FAILED.\
(0)	046644	055200	051105	020117		.ASCIZ	<CRLF>'ZERO X11 AT ST 770 FAILED.'
(0)	046700				EM65:		
(1)	046700	046103	042122	024040		.ASCII	\CLRD (R). FPS BAD.\
(0)	046722	000				.BYTE	0
(0)	046723				EM66:		
(1)	046723	103	051114	020104		.ASCIZ	\CLRD (R). R0 BAD. FDST FAILED.\
(0)	046762				EM67:		
(1)	046762	046103	042122	040440		.ASCII	\CLRD AC7. FPS BAD.\
(1)	047004	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FDST) ST 770 WENT TO 607 INSTEAD OF 617.\
(0)	047063				EM70:		
(1)	047063	103	051114	020104		.ASCII	\CLRD AC7. FEC BAD.\
(0)	047105	000				.BYTE	0
(0)	047106	042516	043107	040440	EM176:	.ASCIZ	'NEGF AC7. FPS BAD.'
(0)	047131	116	043505	020106	EM177:	.ASCIZ	'NEGF AC7. FEC BAD.'
(0)	047154				EM71:		
(1)	047154	042516	043107	040440		.ASCIZ	\NEGF A FAILED.\
(0)	047173				EM72:		
(1)	047173	116	043505	020106		.ASCIZ	\NEGF A. FPS BAD.\
(0)	047214				EM107:		
(1)	047214	042516	042107	024040		.ASCIZ	\NEGD (R) TRAP TO 4 IN SRC MODE.\

(0)	047254				EM73:	
(1)	047254	042516	042107	024040		.ASCIZ \NEGD (R) FAILED.\
(0)	047275				EM74:	
(1)	047275	116	043505	020104		.ASCIZ \NEGD (R), R0 BAD.\
(0)	047317				EM75:	
(1)	047317	116	043505	020104		.ASCIZ \NEGD (R), FPS BAD.\
(0)	047342				EM76:	
(1)	047342	041101	042123	024040		.ASCIZ \ABSD (R)+ TRAP TO 4 IN SRC MODE.\
(0)	047403				EM77:	
(1)	047403	101	051502	020104		.ASCIZ \ABSD (R)+ FAILED.\
(0)	047425				EM100:	
(1)	047425	101	051502	020104		.ASCIZ \ABSD (R)+, R0 BAD.\
(0)	047450				EM101:	
(1)	047450	041101	042123	024040		.ASCIZ \ABSD (R)+, FPS BAD.\
(0)	047474				EM102:	
(1)	047474	041101	042123	026440		.ASCIZ \ABSD -(R) TRAP TO 4 IN SRC MODE.\
(0)	047535				EM103:	
(1)	047535	101	051502	020104		.ASCIZ \ABSD -(R) FAILED.\
(0)	047557				EM104:	
(1)	047557	101	051502	020104		.ASCIZ \ABSD -(R), R0 BAD.\
(0)	047602				EM105:	
(1)	047602	041101	042123	026440		.ASCIZ \ABSD -(R), FPS BAD.\
(0)	047626				EM106:	
(1)	047626	041101	042123	040040		.ASCIZ \ABSD @(R)+ TRAP TO 4 IN SRC MODE.\
(0)	047670				EM110:	
(1)	047670	041101	042123	040040		.ASCIZ \ABSD @(R)+ FAILED.\
(0)	047713				EM111:	
(1)	047713	101	051502	020104		.ASCIZ \ABSD @(R)+, R0 BAD.\
(0)	047737				EM112:	
(1)	047737	101	051502	020104		.ASCIZ \ABSD @(R)+, FPS BAD.\
(0)	047764				EM113:	
(1)	047764	042516	042107	040040		.ASCIZ \NEGD @-(R) TRAP TO 4 IN SRC MODE.\
(0)	050026				EM114:	
(1)	050026	042516	042107	040040		.ASCIZ \NEGD @-(R) FAILED.\
(0)	050051				EM115:	
(1)	050051	116	043505	020104		.ASCIZ \NEGD @-(R), R0 BAD.\
(0)	050075				EM116:	
(1)	050075	116	043505	020104		.ASCIZ \NEGD @-(R), FPS BAD.\
(0)	050122				EM117:	
(1)	050122	041101	042123	047040		.ASCIZ \ABSD N(R) TRAP TO 4 IN SRC MODE.\
(0)	050163				EM120:	
(1)	050163	101	051502	020104		.ASCIZ \ABSD N(R) FAILED.\
(0)	050205				EM121:	
(1)	050205	101	051502	020104		.ASCIZ \ABSD N(R), R0 BAD.\
(0)	050230				EM122:	
(1)	050230	041101	042123	047040		.ASCIZ \ABSD N(R), FPS BAD.\
(0)	050254				EM123:	
(1)	050254	042516	042107	040040		.ASCIZ \NEGD @N(R) TRAP TO 4 IN SRC MODE.\
(0)	050316				EM124:	
(1)	050316	042516	042107	040040		.ASCIZ \NEGD @N(R) FAILED.\
(0)	050341				EM125:	
(1)	050341	116	043505	020104		.ASCIZ \NEGD @N(R), R0 BAD.\
(0)	050365				EM126:	
(1)	050365	116	043505	020104		.ASCIZ \NEGD @N(R), FPS BAD.\
(0)	050412				EM127:	
(1)	050412	042516	042107	047040		.ASCIZ \NEGD N(R7) TRAP TO 4 IN SRC MODE.\

(0)	050454				EM130:	
(1)	050454	042516	042107	047040		.ASCIZ \NEGD N(R7) FAILED.\
(0)	050477				EM131:	
(1)	050477	116	043505	020104		.ASCIZ \NEGD N(R7), FPS BAD.\
(0)	050524				EM132:	
(1)	050524	041101	042123	040040		.ASCIZ \ABSD @N(R7) TRAP TO 4 IN SRC MODE.\
(0)	050567				EM133:	
(1)	050567	101	051502	020104		.ASCIZ \ABSD @N(R7) FAILED.\
(0)	050613				EM134:	
(1)	050613	101	051502	020104		.ASCIZ \ABSD @N(R7), FPS BAD.\
	050641	116	043505	020104	EM135:	.ASCII 'NEGD A FAILED.'
	050657	200	047530	020122		.ASCIZ <CRLF>'XOR SIGN BIT ST 336 FAILED.'
(0)	050714				EM136:	
(1)	050714	042516	042107	040440		.ASCIZ \NEGD A FAILED.\
(0)	050733				EM137:	
(2)	050733	116	043505	020104		.ASCIZ \NEGD A, FPS BAD.\
(0)	050754				EM140:	
(1)	050754	042516	042107	024040		.ASCIZ \NEGD (R) FAILED.\
(0)	050775				EM141:	
(1)	050775	116	043505	020104		.ASCIZ \NEGD (R), R0 BAD, SPECIAL DEST FAILED.\
(0)	051044				EM142:	
(2)	051044	042516	042107	024040		.ASCIZ \NEGD (R), FPS BAD.\
(0)	051067				EM143:	
(1)	051067	116	043505	020104		.ASCIZ \NEGD (R)+ FAILED.\
(0)	051111				EM144:	
(1)	051111	116	043505	020104		.ASCIZ \NEGD (R)+, R0 BAD, SPECIAL DEST FAILED.\
(0)	051161				EM145:	
(2)	051161	116	043505	020104		.ASCIZ \NEGD (R)+, FPS BAD.\
(0)	051205				EM146:	
(1)	051205	116	043505	020104		.ASCIZ \NEGD -(R) FAILED.\
(0)	051227				EM147:	
(1)	051227	116	043505	020104		.ASCIZ \NEGD -(R), R0 BAD, SPECIAL DEST FAILED.\
(0)	051277				EM150:	
(2)	051277	116	043505	020104		.ASCIZ \NEGD -(R), FPS BAD.\
(0)	051323				EM151:	
(1)	051323	116	043505	020104		.ASCIZ \NEGD @(R)+ FAILED.\
(0)	051346				EM152:	
(1)	051346	042516	042107	040040		.ASCIZ \NEGD @(R)+, R0 BAD, SPECIAL DEST FAILED.\
(0)	051417				EM153:	
(2)	051417	116	043505	020104		.ASCIZ \NEGD @(R)+, FPS BAD.\
(0)	051444				EM154:	
(1)	051444	042516	042107	040040		.ASCIZ \NEGD @-(R) FAILED.\
(0)	051467				EM155:	
(1)	051467	116	043505	020104		.ASCIZ \NEGD @-(R), R0 BAD, SPECIAL DEST FAILED.\
(0)	051540				EM156:	
(2)	051540	042516	042107	040040		.ASCIZ \NEGD @-(R), FPS BAD.\
(0)	051565				EM157:	
(1)	051565	116	043505	020106		.ASCIZ \NEGF (R)+ FAILED.\
	051607	116	043505	020106	EM160:	.ASCII 'NEGF (R)+, R0 BAD.'
	051631	102	042101	041440		.ASCIZ 'BAD CONSTANT USED, SPECIAL DEST FAILED.'
(0)	051701				EM161:	
(2)	051701	116	043505	020106		.ASCIZ \NEGF (R)+, FPS BAD.\
(0)	051725				EM162:	
(1)	051725	116	043505	020104		.ASCIZ \NEGD (R7)+ FAILED.\
(0)	051750				EM163:	
(2)	051750	042516	042107	024040		.ASCIZ \NEGD (R7)+, FPS BAD.\

(0)	051775	120	020103	040502	EM164:	.ASCIZ	*PC BAD AFTER NEGD (R7)+. BAD CONSTANT USED.*
(0)	052051				EM215:		
(1)	052051	120	020103	040502		.ASCII	\PC BAD AFTER NEGD N(R). BAD CONSTANT USED 746 746.\
(1)	052133	200	051117	024040		.ASCIZ	<CRLF>*OR (BUT FDST) IN SPECIAL DEST FAILED.*
(0)	052202				EM216:		
(1)	052202	042516	042107	047040		.ASCIZ	\NEGD N(R) FAILED.\
(0)	052224				EM217:		
(1)	052224	042516	042107	047040		.ASCIZ	\NEGD N(R). R0 BAD. SPECIAL DEST FAILED.\
(0)	052274				EM220:		
(2)	052274	042516	042107	047040		.ASCIZ	\NEGD N(R). FPS BAD.\
(0)	052320				EM221:		
(1)	052320	041520	041040	042101		.ASCII	\PC BAD AFTER NEGD @N(R). BAD CONSTANT USED 747 747.\
(1)	052403	200	051117	024040		.ASCIZ	<CRLF>*OR (BUT FDST) IN SPECIAL DEST FAILED.*
(0)	052452				EM222:		
(1)	052452	042516	042107	040040		.ASCIZ	\NEGD @N(R) FAILED.\
(0)	052475				EM223:		
(1)	052475	116	043505	020104		.ASCIZ	\NEGD @N(R). R0 BAD. SPECIAL DEST FAILED.\
(0)	052546				EM224:		
(2)	052546	042516	042107	040040		.ASCIZ	\NEGD @N(R). FPS BAD.\
(0)	052573				EM165:		
(1)	052573	116	043505	020104		.ASCIZ	\NEGD (R) FAILED.\
(0)	052614				EM166:		
(1)	052614	041101	042123	024040		.ASCIZ	\ABSD (R) FAILED.\
(0)	052635				EM167:		
(1)	052635	124	052123	020104		.ASCIZ	\TSTD (R) FAILED.\
(0)	052656				EM170:		
(1)	052656	042516	042107	024040		.ASCIZ	\NEGD (R). FPS BAD.\
(0)	052701				EM171:		
(1)	052701	101	051502	020104		.ASCIZ	\ABSD (R). FPS BAD.\
(0)	052724				EM172:		
(1)	052724	051524	042124	024040		.ASCIZ	\TSTD (R). FPS BAD.\
(0)	052747				EM173:		
(1)	052747	116	043505	020104		.ASCIZ	\NEGD (R). FEC BAD.\
(0)	052772				EM174:		
(1)	052772	041101	042123	024040		.ASCIZ	\ABSD (R). FEC BAD.\
(0)	053015				EM175:		
(1)	053015	124	052123	020104		.ASCIZ	\TSTD (R). FEC BAD.\
(0)	053040				EM200:		
(1)	053040	042516	042107	024040		.ASCII	\NEGD (R) FAILED.\
(0)	053060	054200	051117	051440		.ASCIZ	<CRLF>*XOR SIGN BIT FAILED ST 336.*
(0)	053115				EM201:		
(1)	053115	116	043505	020104		.ASCII	\NEGD (R). FPS BAD.\
(1)	053137	200	041050	052125		.ASCIZ	<CRLF>\(BUT ENBT) ST 336 WENT TO 053 INSTEAD OF 453.\
(0)	053216				EM202:		
(1)	053216	042516	042107	024040		.ASCII	\NEGD (R). FPS BAD.\
(1)	053240	024200	052502	020124		.ASCIZ	<CRLF>\(BUT ENBT) ST 336 WENT TO 453 INSTEAD OF 053.\
(0)	053317				EM203:		
(1)	053317	101	051502	020104		.ASCII	\ABSD (R) FAILED.\
(0)	053337	200	041050	052125		.ASCII	<CRLF>*(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 335, OR*
(1)	053420	024200	052502	020124		.ASCIZ	<CRLF>\(BUT ENBT) ST 335 WENT TO 452 INSTEAD OF 052.\
(0)	053477				EM204:		
(1)	053477	101	051502	020104		.ASCII	\ABSD (R) FAILED.\
(0)	053517	200	047530	020122		.ASCIZ	<CRLF>*XOR SIGN BIT FAILED ST 452.*
(0)	053554				EM205:		
(1)	053554	051524	042124	024040		.ASCII	\TSTD (R) FAILED.\
(1)	053574	024200	052502	020124		.ASCIZ	<CRLF>\(BUT OP1B) ST 055 WENT TO 336 INSTEAD OF 334.\

(0)	053653				EM206:	
(1)	053653	124	052123	020104	.ASCII	\TSTD (R). FPS BAD.\
(1)	053675	200	041050	052125	.ASCIZ	<CRLF>\(BUT ENBT) ST 334 WENT TO 453 INSTEAD OF 053.\
(0)	053754				EM207:	
(1)	053754	051524	042124	024040	.ASCII	\TSTD (R) FAILED.\
(1)	053774	024200	052502	020124	.ASCIZ	<CRLF>\(BUT OP1B) ST 057 WENT TO 335 INSTEAD OF 334.\
(0)	054053				EM210:	
(1)	054053	124	052123	020104	.ASCII	\TSTD (R) FAILED.\
(1)	054073	200	041050	052125	.ASCIZ	<CRLF>\(BUT ENBT) ST 334 WENT TO 053 INSTEAD OF 453.\
(0)	054152				EM211:	
(1)	054152	051524	042124	024040	.ASCII	\TSTD (R) FAILED.\
(1)	054172	024200	052502	020124	.ASCIZ	<CRLF>\(BUT OP1B) ST 255 WENT TO 311 OR 312 INSTEAD OF 310.\
(0)	054260				EM212:	
(1)	054260	051524	042124	024040	.ASCII	\TSTD (R). FPS BAD.\
(1)	054302	024200	052502	020124	.ASCIZ	<CRLF>\(BUT ENBT) ST 310 WENT TO 402 INSTEAD OF 002.\
(0)	054361				EM213:	
(1)	054361	124	052123	020104	.ASCII	\TSTD (R). FPS BAD.\
	054403	040	044506	053125	.ASCII	' FIUV=0, OPERAND=-0.'
(1)	054427	200	041050	052125	.ASCIZ	<CRLF>\(BUT FIUV) ST 257 WENT TO 355 INSTEAD OF 255.\
(0)	054506				EM214:	
(1)	054506	051524	042124	024040	.ASCII	\TSTD (R). FPS BAD.\
	054530	043040	052511	036526	.ASCII	' FIUV=1, OPERAND=-0.'
(1)	054554	024200	052502	020124	.ASCIZ	<CRLF>\(BUT FIUV) ST 257 WENT TO 255 INSTEAD OF 355.\
(0)	054633				EM225:	
(1)	054633	114	043104	051520	.ASCIZ	\LDFPS (R). R0 BAD.\
(0)	054656				EM226:	
(1)	054656	042114	050106	020123	.ASCIZ	\LDFPS (R). FPS BAD.\
(0)	054702				EM227:	
(1)	054702	042114	050106	020123	.ASCIZ	\LDFPS (R) TRAPPED TO 4.\
(0)	054732				EM230:	
(1)	054732	042114	050106	020123	.ASCIZ	\LDFPS (R)+. R0 BAD.\
(0)	054756				EM231:	
(1)	054756	042114	050106	020123	.ASCIZ	\LDFPS (R)+. FPS BAD.\
(0)	055003				EM232:	
(1)	055003	114	043104	051520	.ASCIZ	\LDFPS (R)+ TRAPPED TO 4.\
(0)	055034				EM233:	
(1)	055034	042114	050106	020123	.ASCIZ	\LDFPS -(R). R0 BAD.\
(0)	055060				EM234:	
(1)	055060	042114	050106	020123	.ASCIZ	\LDFPS -(R). FPS BAD.\
(0)	055105				EM235:	
(1)	055105	114	043104	051520	.ASCIZ	\LDFPS -(R) TRAPPED TO 4.\
(0)	055136				EM236:	
(1)	055136	042114	050106	020123	.ASCIZ	\LDFPS @(R)+. R0 BAD.\
(0)	055163				EM237:	
(1)	055163	114	043104	051520	.ASCIZ	\LDFPS @(R)+. FPS BAD.\
(0)	055211				EM240:	
(1)	055211	114	043104	051520	.ASCIZ	\LDFPS @(R)+ TRAPPED TO 4.\
(0)	055243				EM241:	
(1)	055243	114	043104	051520	.ASCIZ	\LDFPS @-(R). R0 BAD.\
(0)	055270				EM242:	

(1)	055270	042114	050106	020123		.ASCIZ	\LDFPS @-(R), FPS BAD,\
(0)	055316				EM243:		
(1)	055316	042114	050106	020123		.ASCIZ	\LDFPS @-(R) TRAPPED TO 4,\
(0)	055350				EM244:		
(1)	055350	042114	050106	020123		.ASCIZ	\LDFPS N(R), R0 BAD,\
(0)	055374				EM245:		
(1)	055374	042114	050106	020123		.ASCIZ	\LDFPS N(R), FPS BAD,\
(0)	055421				EM246:		
(1)	055421	120	020103	040502		.ASCIZ	\PC BAD AFTER LDFPS N(R),\
(0)	055452				EM247:		
(1)	055452	042114	050106	020123		.ASCIZ	\LDFPS N(R) TRAPPED TO 4,\
(0)	055503				EM250:		
(1)	055503	114	043104	051520		.ASCIZ	\LDFPS @N(R), R0 BAD,\
(0)	055530				EM251:		
(1)	055530	042114	050106	020123		.ASCIZ	\LDFPS @N(R), FPS BAD,\
(0)	055556				EM252:		
(1)	055556	041520	041040	042101		.ASCIZ	\PC BAD AFTER LDFPS @N(R),\
(0)	055610				EM253:		
(1)	055610	042114	050106	020123		.ASCIZ	\LDFPS @N(R) TRAPPED TO 4,\
(0)	055642				EM254:		
(1)	055642	041520	041040	042101		.ASCIZ	\PC BAD AFTER LDCLD (R7)+,A,\
(0)	055676				EM255:		
(1)	055676	042114	046103	020104		.ASCIZ	\LDCLD (R7)+,A TRAPPED TO 4,\
(0)	055732				EM256:		
(1)	055732	042114	046103	020104		.ASCIZ	\LDCLD (R)+,A, R0 BAD,\
(0)	055760				EM257:		
(1)	055760	042114	046103	020104		.ASCIZ	\LDCLD (R)+,A, FPS BAD,\
(0)	056007				EM260:		
(1)	056007	114	041504	043111		.ASCII	\LDCIF OR LDCLF (R),A FAILED,\
	056043	000				.BYTE	0
(0)	056044				EM261:		
(1)	056044	042114	044503	020106		.ASCII	\LDCIF OR LDCLF (R),A, FPS BAD,\
	056102	000				.BYTE	0
(0)	056103				EM262:		
(1)	056103	114	041504	043111		.ASCII	\LDCIF (R),A FAILED,\
(1)	056126	024200	052502	020124		.ASCIZ	<CR LF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301,\
(0)	056203				EM263:		
(1)	056203	114	041504	043114		.ASCII	\LDCLF (R),A, FPS BAD,\
	056230	000				.BYTE	0
(0)	056231				EM264:		
(1)	056231	114	041504	043111		.ASCII	\LDCIF (R),A FAILED,\
	056254	052600	042523	020104		.ASCIZ	<CR LF>'USED CONSTANT 237 INSTEAD OF 217 ST 107,'
(0)	056326				EM265:		
(1)	056326	042114	044503	020106		.ASCII	\LDCIF OR LDCLF (R),A FAILED,\
	056362	051600	052105	051440		.ASCIZ	<CR LF>'SET SIGN BIT FAILED ST 146.'

(0)	056417				EM266:		
(1)	056417	114	041504	043111		.ASCII	\LDCIF OR LDCLF (R),A FAILED.\
(1)	056453	200	041050	052125		.ASCIZ	<CRLF>\(BUT XNBT) ST 372 WENT TO 152 INSTEAD OF 112.\
(0)	056532				EM267:		
(1)	056532	042114	046103	020106		.ASCII	\LDCLF (R),A FAILED.\
	056555	200	051525	042105		.ASCIZ	<CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'
(0)	056627				EM270:		
(1)	056627	114	041504	043114		.ASCII	\LDCLF (R),A FAILED.\
	056652	051040	052517	042116		.ASCIZ	' ROUND ERROR.'
(0)	056670				EM271:		
(1)	056670	042114	046103	020106		.ASCII	\LDCLF (R),A FAILED.\
	056713	040	051124	047125		.ASCIZ	' TRUNCATION ERROR.'
(0)	056736				EM272:		
(1)	056736	042114	044503	020106		.ASCII	\LDCIF OR LDCLF (R),A FAILED.\
	056772	051200	032061	047040		.ASCIZ	<CRLF>'R14 NOT INCREMENTED ST 630.'
(0)	057027				EM273:		
(1)	057027	114	041504	042111		.ASCII	\LDCID OR LDCLD (R),A FAILED.\
	057063	000				.BYTE	0
(0)	057064				EM274:		
(1)	057064	042114	044503	020104		.ASCII	\LDCID OR LDCLD (R),A, FPS BAD.\
	057122	000				.BYTE	0
(0)	057123				EM275:		
(1)	057123	114	041504	042111		.ASCII	\LDCID (R),A FAILED.\
(1)	057146	024200	052502	020124		.ASCIZ	<CRLF>\(BUT FL) ST 277 WENT TO 300 INSTEAD OF 301.\
(0)	057223				EM276:		
(1)	057223	114	041504	042111		.ASCII	\LDCID (R),A FAILED.\
	057246	052600	042523	020104		.ASCIZ	<CRLF>'USED CONSTANT 237 INSTEAD OF 217 ST 107.'
(0)	057320				EM277:		
(1)	057320	042114	044503	020104		.ASCII	\LDCID (R),A FAILED.\
	057343	200	042523	020124		.ASCIZ	<CRLF>'SET SIGN FAILED ST 146.'
(0)	057374				EM300:		
(1)	057374	042114	046103	020104		.ASCII	\LDCLD (R),A FAILED.\
	057417	200	051525	042105		.ASCIZ	<CRLF>'USED CONSTANT 217 INSTEAD OF 237 ST 107.'
(0)	057471				EM301:		
(1)	057471	114	042504	050130		.ASCII	\LDEXP (R),A FAILED.\
	057514	000				.BYTE	0
(0)	057515				EM302:		
(1)	057515	114	042504	050130		.ASCII	\LDEXP (R),A, FPS BAD.\
	057542	000				.BYTE	0
	057543	114	042504	050130	EM303:	.ASCIZ	'LDEXP (R),A, FEC BAD.'
(0)	057571				EM304:		
(1)	057571	114	042504	050130		.ASCII	\LDEXP (R),A FAILED.\
	057614	042600	041530	051505		.ASCIZ	<CRLF>'EXCESS 200 CALCULATION ST 624 BAD.'

(0)	057660				EM305:	
(1)	057660	042114	054105	020120	.ASCII	\LDEXP (R),A, FPS BAD.\
	057705	050	052502	020124	.ASCII	'(BUT ENBT,EZBT,XNBT) ST 625 DID NOT GO TO 304.'
(0)	057763				EM306:	
(1)	057763	114	042504	050130	.ASCII	\LDEXP (R),A, FPS BAD.\
	060010	024200	052502	020124	.ASCII	<CRLF>'(BUT EZBT) ST 544 WENT TO 504 INSTEAD OF 704, OR'
(1)	060071	200	041050	052125	.ASCIIZ	<CRLF>\(BUT EZBT) ST 704 WENT TO 264 INSTEAD OF 064.\
(0)	060150				EM307:	
(1)	060150	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	060173	200	041050	052125	.ASCIIZ	<CRLF>\(BUT EZBT) ST 704 WENT TO 064 INSTEAD OF 264.\
(0)	060252				EM310:	
(1)	060252	042114	054105	020120	.ASCII	\LDEXP (R),A, FPS BAD.\
(1)	060277	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FIU) ST 264 WENT TO 115 INSTEAD OF 155.\
(0)	060355				EM311:	
(1)	060355	114	042504	050130	.ASCII	\LDEXP (R),A FAILED.\
(1)	060400	024200	052502	020124	.ASCIIZ	<CRLF>\(BUT FIU) ST 264 WENT TO 155 INSTEAD OF 115.\
(0)	060456				EM312:	
(1)	060456	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	060501	200	041050	052125	.ASCIIZ	<CRLF>\(BUT EZBT) ST 544 WENT TO 704 INSTEAD OF 504.\
(0)	060560				EM313:	
(1)	060560	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	060603	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FIU) ST 504 WENT TO 155 INSTEAD OF 115.\
(0)	060661				EM314:	
(1)	060661	114	042504	050130	.ASCII	\LDEXP (R),A FAILED.\
(1)	060704	024200	052502	020124	.ASCIIZ	<CRLF>\(BUT FIV) ST 104 WENT TO 116 INSTEAD OF 136.\
(0)	060762				EM315:	
(1)	060762	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	061005	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FIV) ST 104 WENT TO 136 INSTEAD OF 116.\
(0)	061063				EM316:	
(1)	061063	114	042504	050130	.ASCII	\LDEXP (R),A FAILED.\
(1)	061106	024200	052502	020124	.ASCIIZ	<CRLF>\(BUT FIV) ST 144 WENT TO 116 INSTEAD OF 136.\
(0)	061164				EM317:	
(1)	061164	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	061207	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FIV) ST 144 WENT TO 136 INSTEAD OF 116.\
(0)	061265				EM320:	
(1)	061265	114	042504	050130	.ASCII	\LDEXP (R),A FAILED.\
(1)	061310	024200	052502	020124	.ASCIIZ	<CRLF>\(BUT FIV) ST 344 WENT TO 116 INSTEAD OF 136.\
(0)	061366				EM321:	
(1)	061366	042114	054105	020120	.ASCII	\LDEXP (R),A FAILED.\
(1)	061411	200	041050	052125	.ASCIIZ	<CRLF>\(BUT FIV) ST 344 WENT TO 136 INSTEAD OF 116.\
(0)	061467				EM322:	
(1)	061467	123	041524	044504	.ASCII	\STCDI OR STCDL (R),A FAILED.\
	061523	000			.BYTE	0



(0)	061524				EM323:		
(1)	061524	052123	042103	020111	.ASCII	\STCDI OR STCDL (R),A, FPS BAD,\	
	061562	000			.BYTE	0	
	061563	123	041524	044504	EM324:	.ASCIZ	'STCDI OR STCDL (R),A, FEC BAD,'
(0)	061622				EM325:		
(1)	061622	052123	042103	020114	.ASCII	\STCDL (R),A, FPS BAD,\	
	061647	200	046103	040505	.ASCII	<CRLF>'CLEAR FLAG ST 774 FAILED, OR'	
(1)	061704	024200	052502	020124	.ASCIZ	<CRLF>\(BUT FLAG) ST 662 WENT TO 365 INSTEAD OF 361,\	
	061622				EM326=EM325		
(0)	061763				EM327:		
(1)	061763	123	041524	046104	.ASCII	\STCDL (R),A FAILED,\	
(1)	062006	024200	052502	020124	.ASCIZ	<CRLF>\(BUT ENBT) ST 632 WENT TO 473 INSTEAD OF 073,\	
(0)	062065				EM330:		
(1)	062065	123	041524	046104	.ASCII	\STCDL (R),A, FPS BAD,\	
(1)	062112	024200	052502	020124	.ASCIZ	<CRLF>\(BUT FIC) ST 004 WENT TO 305 INSTEAD OF 315,\	
(0)	062170				EM331:		
(1)	062170	052123	042103	020114	.ASCII	\STCDL (R),A, FPS BAD,\	
(1)	062215	200	041050	052125	.ASCIZ	<CRLF>\(BUT FIC) ST 004 WENT TO 315 INSTEAD OF 305,\	
	061524				EM333=EM323		
(0)	062273				EM334:		
(1)	062273	123	041524	044504	.ASCII	\STCDI (R),A, FPS BAD,\	
	062320	052600	042523	020104	.ASCIZ	<CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66,'	
(0)	062367				EM335:		
(1)	062367	123	041524	044504	.ASCII	\STCDI (R),A FAILED,\	
(1)	062412	024200	052502	020124	.ASCIZ	<CRLF>\(BUT ENBT) ST 632 WENT TO 073 INSTEAD OF 473,\	
(0)	062471				EM336:		
(1)	062471	123	041524	044504	.ASCII	\STCDI (R),A, FPS BAD,\	
	062516	051600	052105	043040	.ASCIZ	<CRLF>'SET FN ST 473 FAILED,'	
(0)	062545				EM337:		
(1)	062545	123	041524	046104	.ASCII	\STCDL (R),A FAILED,\	
(1)	062570	024200	052502	020124	.ASCIZ	<CRLF>\(BUT COUT) ST 275 WENT TO 074 INSTEAD OF 274,\	
(0)	062647				EM340:		
(1)	062647	123	041524	046104	.ASCII	\STCDL (R),A FAILED,\	
(1)	062672	024200	052502	020124	.ASCIZ	<CRLF>\(BUT COUT) ST 275 WENT TO 274 INSTEAD OF 074,\	
(0)	062751				EM341:		
(1)	062751	123	041524	046104	.ASCII	\STCDL (R),A, FPS BAD,\	
(1)	062776	024200	052502	020124	.ASCIZ	<CRLF>\(BUT EZBT) ST 377 WENT TO 633 INSTEAD OF 433,\	
(0)	063055				EM342:		
(1)	063055	123	041524	046104	.ASCII	\STCDL (R),A FAILED,\	
(1)	063100	024200	052502	020124	.ASCIZ	<CRLF>\(BUT COUT) ST 360 WENT TO 654 INSTEAD OF 454,\	

(0)	063157				EM343:		
(1)	063157	123	041524	046104		.ASCII	\STCDL (R),A FAILED,\
(1)	063202	024200	052502	020124		.ASCIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 531 INSTEAD OF 431,\
(0)	063261				EM344:		
(1)	063261	123	041524	046104		.ASCII	\STCDL (R),A FAILED,\
(1)	063304	024200	052502	020124		.ASCII	<CRLF>'(BUT COUT) ST 360 WENT TO 454 INSTEAD OF 654, OR'
(1)	063365	200	041050	052125		.ASCIZ	<CRLF>\(BUT NBIT) ST 654 WENT TO 431 INSTEAD OF 531,\
(0)	063444				EM332:		
(1)	063444	052123	042103	020111		.ASCII	\STCDI (R),A FAILED,\
	063467	200	051525	042105		.ASCIZ	<CRLF>'USED CONSTANT 37 INSTEAD OF 17 ST 66.'
(0)	063536				EM345:		
(1)	063536	052123	042103	020111		.ASCII	\STCDI (R),A FAILED,\
(1)	063561	200	041050	052125		.ASCIZ	<CRLF>\(BUT FL) ST 633 WENT TO 655 INSTEAD OF 654,\
(0)	063636				EM346:		
(1)	063636	052123	043103	020114		.ASCII	\STCFL (R),A FAILED,\
	063661	200	042532	047522		.ASCIZ	<CRLF>'ZERO LOW ORDER PART OF X11 FAILED ST 773.'
(0)	063734				EM347:		
(1)	063734	052123	054105	020120		.ASCII	\STEXP A,(R) FAILED,\
	063757	000				.BYTE	0
(0)	063760				EM350:		
(1)	063760	052123	054105	020120		.ASCII	\STEXP A,(R). FPS BAD,\
	064005	000				.BYTE	0
	064006	047515	042522	052040	EM351:	.ASCII	'MORE THAN ONE WORD '
	064031	127	044522	052124		.ASCIZ	'WRITTEN BY STEXP A,(R). '<CRLF>'ZERO FDFL ST 347 FAILED.'
(0)	064112				EM352:		
(1)	064112	052123	054105	020120		.ASCII	\STEXP A,(R). FPS BAD,\
(1)	064137	200	041050	052125		.ASCIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 071 INSTEAD OF 471,\
(0)	064216				EM353:		
(1)	064216	052123	054105	020120		.ASCII	\STEXP A,(R). FPS BAD,\
(1)	064243	200	041050	052125		.ASCIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 072 INSTEAD OF 272,\
(0)	064322				EM354:		
(1)	064322	052123	054105	020120		.ASCII	\STEXP A,(R). FPS BAD,\
(1)	064347	200	041050	052125		.ASCIZ	<CRLF>\(BUT EZBT) ST 071 WENT TO 272 INSTEAD OF 072,\
(0)	064426				EM355:		
(1)	064426	052123	054105	020120		.ASCII	\STEXP A,(R). FPS BAD,\
(1)	064453	200	041050	052125		.ASCIZ	<CRLF>\(BUT ENBT) ST 376 WENT TO 471 INSTEAD OF 071,\
	064532	052123	052123	024040	EM356:	.ASCII	'STST (R) GOT BAD FEC,'<CRLF>
	064560	043101	042524	020122		.ASCIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
	064630	052123	052123	024040	EM357:	.ASCII	'STST (R) GOT BAD FEA,'<CRLF>
	064656	043101	042524	020122		.ASCIZ	'AFTER EXECUTING AN ILLEGAL FPP OP CODE.'
	064726	047117	054514	047440	EM360:	.ASCII	'ONLY ONE WORD WRITTEN BY STST (R), '

	064771	123	052105	043040		.ASCIZ	'SET FDFL ST 636 FAILED.'
(0)	065021				EM401:		
(1)	065021	123	043124	051520		.ASCIZ	\STFPS (R). R0 BAD.\
(0)	065044				EM402:		
(1)	065044	052123	050106	020123		.ASCIZ	\STFPS (R) FAILED.\
	065066	047515	042522	052040	EM403:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R).'
(1)	065136	024200	052502	020124		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	065220				EM404:		
(1)	065220	052123	050106	020123		.ASCIZ	\STFPS (R) TRAPPED TO 4.\
(0)	065250				EM405:		
(1)	065250	052123	050106	020123		.ASCIZ	\STFPS (R)+. R0 BAD.\
(0)	065274				EM406:		
(1)	065274	052123	050106	020123		.ASCIZ	\STFPS (R)+ FAILED.\
	065317	115	051117	020105	EM407:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS (R)+.'
(1)	065370	024200	052502	020124		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	065452				EM410:		
(1)	065452	052123	050106	020123		.ASCIZ	\STFPS (R)+ TRAPPED TO 4.\
(0)	065503				EM411:		
(1)	065503	123	043124	051520		.ASCIZ	\STFPS -(R). R0 BAD.\
(0)	065527				EM412:		
(1)	065527	123	043124	051520		.ASCIZ	\STFPS -(R) FAILED.\
	065552	047515	042522	052040	EM413:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS -(R).'
(1)	065623	200	041050	052125		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	065705				EM414:		
(1)	065705	123	043124	051520		.ASCIZ	\STFPS -(R) TRAPPED TO 4.\
(0)	065736				EM415:		
(1)	065736	052123	050106	020123		.ASCIZ	\STFPS @(R)+. R0 BAD.\
(0)	065763				EM416:		
(1)	065763	123	043124	051520		.ASCIZ	\STFPS @(R)+ FAILED.\
	066007	123	043124	051520	EM417:	.ASCIZ	'STFPS @(R)+ DID NOT DEFFER THE WRITE.'
(0)	066055				EM420:		
(1)	066055	123	043124	051520		.ASCIZ	\STFPS @(R)+ TRAPPED TO 4.\
(0)	066107				EM421:		
(1)	066107	123	043124	051520		.ASCIZ	\STFPS @-(R). R0 BAD.\
(0)	066134				EM422:		
(1)	066134	052123	050106	020123		.ASCIZ	\STFPS @-(R) FAILED.\
	066160	052123	050106	020123	EM423:	.ASCIZ	'STFPS @-(R) DID NOT DEFFER THE WRITE.'
(0)	066226				EM424:		
(1)	066226	052123	050106	020123		.ASCIZ	\STFPS @-(R) TRAPPED TO 4.\
(0)	066260				EM425:		
(1)	066260	052123	050106	020123		.ASCIZ	\STFPS N(R). R0 BAD.\
(0)	066304				EM426:		
(1)	066304	052123	050106	020123		.ASCIZ	\STFPS N(R) FAILED.\
	066327	115	051117	020105	EM427:	.ASCII	'MORE THAN ONE WORD WRITTEN BY STFPS N(R).'
(1)	066400	024200	052502	020124		.ASCIZ	<CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	066462				EM430:		
(1)	066462	052123	050106	020123		.ASCIZ	\STFPS N(R) TRAPPED TO 4.\
	066513	120	020103	040502	EM431:	.ASCII	'PC BAD AFTER STFPS N(R). BAD CONSTANT USED.'

(0)	066566				EM432:	
(1)	066566	052123	050106	020123		.ASCIZ \STFPS @N(R). R0 BAD.\
(0)	066613				EM433:	
(1)	066613	123	043124	051520		.ASCIZ \STFPS @N(R) FAILED.\
(0)	066637	115	051117	020105	EM434:	.ASCII 'MORE THAN ONE WORD WRITTEN BY STFPS @N(R).'
(1)	066711	200	041050	052125		.ASCIZ <CRLF>\(BUT GR7,-FL) ST 357 WENT TO 416 INSTEAD OF 417.\
(0)	066773				EM435:	
(1)	066773	123	043124	051520		.ASCIZ \STFPS @N(R) TRAPPED TO 4.\
	067025	120	020103	040502	EM436:	.ASCIZ 'PC BAD AFTER STFPS @N(R). BAD CONSTANT USED.'
(0)	067102				EM437:	
(1)	067102	052123	042103	020114		.ASCIZ \STCDL A,(R)+. R0 BAD.\
(0)	067130				EM440:	
(1)	067130	052123	042103	020114		.ASCIZ \STCDL A,-(R). R0 BAD.\
	067156	052123	052123	024040	EM361:	.ASCIZ 'STST (R). FPS BAD.'
	000000				EM362=0	
	000000				EM363=0	
	000000				EM364=0	
	000000				EM365=0	
	000000				EM366=0	
	000000				EM367=0	
	000000				EM370=0	
	000000				EM371=0	
	000000				EM372=0	
	000000				EM373=0	
	000000				EM374=0	
	000000				EM375=0	
	000000				EM376=0	
	000000				EM377=0	
	000000				EM400=0	
	067201	125	042516	050130	EM441:	.ASCIZ 'UNEXPECTED FPP TRAP TO 244.'
	067235	125	042516	050130	EM442:	.ASCIZ 'UNEXPECTED CPU TRAP TO 4.'
	067267	125	042516	050130	EM443:	.ASCIZ 'UNEXPECTED CPU TRAP TO 10.'

;THESE ARE DATA TABLE HEADERS:

	067322	020040	042524	052123	DH1:	.ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
	067362	043011	051520	004456		.ASCIZ <TAB>'FPS.<TAB>'FEC.'
(0)	067375				DH2:	
(1)	067375	040	052040	051505		.ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
	067435	011	047507	020124		.ASCIZ <TAB>'GOT FPS.<TAB>'EXPECTED FPS.'
(0)	067465				DH3:	
(1)	067465	040	052040	051505		.ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
	067525	011	047507	020124		.ASCIZ <TAB>'GOT FEC.<TAB>'EXPECTED FEC.'
(0)	067555				DH4:	
(1)	067555	040	052040	051505		.ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
	067615	011	047507	020124		.ASCIZ <TAB>'GOT R0. <TAB>'EXPECTED R0.'
(0)	067644				DH5:	
(1)	067644	020040	042524	052123		.ASCII ' TEST.<TAB>'PC OF CALL.<TAB>'PC OF ERROR.'
	067704	000				.BYTE 0
	067644				DH6=DH5	
	067555				DH7=DH4	

	067644			DH10=DH5	
	067555			DH11=DH4	
	067644			DH12=DH5	
	067705	040	052040 051505	DH13:	.ASCIZ ' TEST.' <tab>'PC OF CALL,'<tab>'PC OF TRAP,'</tab></tab>
	067705			DH14=DH13	
	067644			DH15=DH5	
(0)	067745			DH16:	
(1)	067745	040	052040 051505		.ASCII ' TEST.' <tab>'PC OF CALL,'<tab>'PC OF ERROR,'</tab></tab>
	070005	011	047507 020124		.ASCIZ <TAB>'GOT PC,' <tab>'EXPECTED PC,'</tab>
	067705			DH17=DH13	
	067555			DH20=DH4	
	067644			DH21=DH5	
	067644			DH22=DH5	
	067705			DH23=DH13	
	067555			DH24=DH4	
	067644			DH25=DH5	
	067705			DH26=DH13	
	067555			DH27=DH4	
	067644			DH30=DH5	
	067705			DH31=DH13	
	067555			DH32=DH4	
	067644			DH33=DH5	
	067705			DH34=DH13	
	067555			DH35=DH4	
	067644			DH36=DH5	
070034	020040	042524	052123	DH37:	.ASCIZ ' TEST.' <tab>'PC OF CALL,'<tab>'PC OF ERROR,'<tab>'GOT FPS,'<tab>'EXPEC</tab></tab></tab></tab>
	070034			DH40=DH37	
070124	020040	042524	052123	DH41:	.ASCIZ ' TEST.' <tab>'PC OF CALL,'<tab>'PC OF ERROR,'<tab>'FPS,'<tab>'GOT FEC,</tab></tab></tab></tab>
	070034			DH42=DH37	
	070034			DH43=DH37	
	070034			DH44=DH37	
	070034			DH45=DH37	
	070034			DH46=DH37	
	070034			DH47=DH37	
	070034			DH50=DH37	
	070034			DH51=DH37	
	070034			DH52=DH37	
	070124			DH53=DH41	
	070034			DH54=DH37	
	070034			DH55=DH37	
	070034			DH56=DH37	
	070034			DH57=DH37	
	070034			DH60=DH37	
	070034			DH61=DH37	
	067375			DH62=DH2	
	067465			DH63=DH3	
	067644			DH64=DH5	
	067375			DH65=DH2	
	067555			DH66=DH4	
	067375			DH67=DH2	
	067465			DH70=DH3	
	067375			DH176=DH2	
	067465			DH177=DH3	
	067644			DH71=DH5	
	067375			DH72=DH2	
	067705			DH107=DH13	

067644	DH73=DH5
067555	DH74=DH4
067375	DH75=DH2
067705	DH76=DH107
067644	DH77=DH5
067555	DH100=DH4
067375	DH101=DH2
067705	DH102=DH107
067644	DH103=DH5
067555	DH104=DH4
067375	DH105=DH2
067705	DH106=DH107
067644	DH110=DH5
067555	DH111=DH4
067375	DH112=DH2
067705	DH113=DH107
067644	DH114=DH5
067555	DH115=DH4
067375	DH116=DH2
067705	DH117=DH107
067644	DH120=DH5
067555	DH121=DH4
067375	DH122=DH2
067705	DH123=DH107
067644	DH124=DH5
067555	DH125=DH4
067375	DH126=DH2
067705	DH127=DH107
067644	DH130=DH5
067375	DH131=DH2
067705	DH132=DH107
067644	DH133=DH5
067375	DH134=DH2
067644	DH135=DH5
067644	DH136=DH5
067375	DH137=DH2
067644	DH140=DH5
067555	DH141=DH4
067375	DH142=DH2
067644	DH143=DH5
067555	DH144=DH4
067375	DH145=DH2
067644	DH146=DH5
067555	DH147=DH4
067375	DH150=DH2
067644	DH151=DH5
067555	DH152=DH4
067375	DH153=DH2
067644	DH154=DH5
067555	DH155=DH4
067375	DH156=DH2
067644	DH157=DH5
067555	DH160=DH4
067375	DH161=DH2
067644	DH162=DH5
067375	DH163=DH2

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

067555	DH225=DH4
067375	DH226=DH2
070221 040 052040 051505	DH227: .ASCIZ ' TEST.'<TAB>'PC OF CALL.'<TAB>'PC OF TRAP.'
067555	DH230=DH4
067375	DH231=DH2
070221	DH232=DH227
067555	DH233=DH4
067375	DH234=DH2
070221	DH235=DH227
067555	DH236=DH4
067375	DH237=DH2
070221	DH240=DH227
067555	DH241=DH4
067375	DH242=DH2
070221	DH243=DH227
067555	DH244=DH4
067375	DH245=DH2
067745	DH246=DH16
070221	DH247=DH227
067555	DH250=DH4
067375	DH251=DH2
067745	DH252=DH16
070221	DH253=DH227
067745	DH254=DH16

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



070034	DH345=DH37
070034	DH346=DH37
070034	DH347=DH37
070034	DH350=DH37
067705	DH351=DH13
070034	DH352=DH37
070034	DH353=DH37
070034	DH354=DH37
070034	DH355=DH37
067555	DH356=DH11

070261	040	052040	051505	DH357:	.ASCII	' TEST, '<TAB>'PC OF CALL, '<TAB>'PC OF ERROR, '
070321	011	047507	020124		.ASCIZ	<TAB>'GOT FEA, '<TAB>'EXPECTED FEA, '

067705	DH360=DH13
067375	DH361=DH2

000000	DH362=0
000000	DH363=0
000000	DH364=0
000000	DH365=0
000000	DH366=0
000000	DH367=0
000000	DH370=0
000000	DH371=0
000000	DH372=0
000000	DH373=0
000000	DH374=0
000000	DH375=0
000000	DH376=0
000000	DH377=0
000000	DH400=0

067555	DH401=DH4
067375	DH402=DH2
067705	DH403=DH13
070221	DH404=DH227
067555	DH405=DH4
067375	DH406=DH2
067705	DH407=DH13
070221	DH410=DH227
067555	DH411=DH4
067375	DH412=DH2
067705	DH413=DH13
070221	DH414=DH227
067555	DH415=DH4
067375	DH416=DH2
067705	DH417=DH13
070221	DH420=DH227
067555	DH421=DH4
067375	DH422=DH2
067705	DH423=DH13
070221	DH424=DH227
067555	DH425=DH4
067375	DH426=DH2
067705	DH427=DH13
070221	DH430=DH227
067705	DH431=DH13

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

;THESE ARE FORMAT SPECIFICATIONS FOR THE DATA TABLES:

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

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

070467	DF141=DF2
070467	DF142=DF2
070515	DF143=DF12
070467	DF144=DF2
070467	DF145=DF2
070515	DF146=DF12
070467	DF147=DF2
070467	DF150=DF2
070515	DF151=DF12
070467	DF152=DF2
070467	DF153=DF2
070515	DF154=DF12
070467	DF155=DF2
070467	DF156=DF2
070515	DF157=DF12
070467	DF160=DF2
070467	DF161=DF2
070515	DF162=DF12
070467	DF163=DF2
070467	DF164=DF2
070467	DF215=DF2
070515	DF216=DF12
070467	DF217=DF2
070467	DF220=DF2
070467	DF221=DF2
070515	DF222=DF12
070467	DF223=DF2
070467	DF224=DF2
070527	DF165=DF37
070527	DF166=DF37
070527	DF167=DF37
070527	DF170=DF37
070527	DF171=DF37
070527	DF172=DF37
070550	DF173=DF41
070550	DF174=DF41
070550	DF175=DF41
070527	DF200=DF37
070527	DF201=DF37
070527	DF202=DF37
070527	DF203=DF37
070527	DF204=DF37
070527	DF205=DF37
070527	DF206=DF37
070527	DF207=DF37
070527	DF210=DF37
070527	DF211=DF37
070527	DF212=DF37
070527	DF213=DF37
070527	DF214=DF37

070606	004	000	005	DF225: .BYTE	4,0,5,0,5,0,5,0
	070606			DF226=DF225	
070616	004	000	005	DF227: .BYTE	4,0,5,0
	070606			DF230=DF225	
	070606			DF231=DF225	
	070616			DF232=DF227	

070606	DF233=DF225
070606	DF234=DF225
070616	DF235=DF227
070606	DF236=DF225
070606	DF237=DF225
070616	DF240=DF227
070606	DF241=DF225
070606	DF242=DF225
070616	DF243=DF227
070606	DF244=DF225
070606	DF245=DF225
070606	DF246=DF225
070616	DF247=DF227
070606	DF250=DF225
070606	DF251=DF225
070606	DF252=DF225
070616	DF253=DF227
070606	DF254=DF225
070616	DF255=DF227
070606	DF256=DF225
070606	DF257=DF225

070622	004	000	005	DF260: .BYTE	4,0,5,0,5,0,5,0,5,5,2,5,5,2,5,5,2
070622				DF261=DF260	
070622				DF262=DF260	
070622				DF263=DF260	
070622				DF264=DF260	
070622				DF265=DF260	
070622				DF266=DF260	
070622				DF267=DF260	
070622				DF270=DF260	
070622				DF271=DF260	
070622				DF272=DF260	

070643	004	000	005	DF273: .BYTE	4,0,5,0,5,0,5,0,5,5,2,5,5,3,5,5,3
070643				DF274=DF273	
070643				DF275=DF273	
070643				DF276=DF273	
070643				DF277=DF273	
070643				DF300=DF273	

070664	004	000	005	DF301: .BYTE	4,0,5,0,5,0,5,0,5,5,3,5,5,0,5,5,3,5,5,3
070664				DF302=DF301	

070710	004	000	005	DF303: .BYTE	4,0,5,0,5,0,0,0,5,5,3,5,5,0,5,5,3,5,5,3
070664				DF304=DF301	
070664				DF305=DF301	
070664				DF306=DF301	
070664				DF307=DF301	
070664				DF310=DF301	
070664				DF311=DF301	
070664				DF312=DF301	
070664				DF313=DF301	
070664				DF314=DF301	
070664				DF315=DF301	
070664				DF316=DF301	
070664				DF317=DF301	
070664				DF320=DF301	

070664 DF321=DF301

070734 004 000 005 DF322: .BYTE 4,0,5,0,5,0,5,0,5,5,3,5,5,2,5,5,2

070734 DF323=DF322

070755 004 000 005 DF324: .BYTE 4,0,5,0,5,0,0,0,5,5,3,5,5,2,5,5,2

070734 DF325=DF322

070734 DF326=DF322

070734 DF327=DF322

070734 DF330=DF322

070734 DF331=DF322

070734 DF332=DF322

070734 DF333=DF322

070734 DF334=DF322

070734 DF335=DF322

070734 DF336=DF322

070734 DF337=DF322

070734 DF340=DF322

070734 DF341=DF322

070734 DF342=DF322

070734 DF343=DF322

070734 DF344=DF322

070734 DF345=DF322

070734 DF346=DF322

070776 004 000 005 DF347: .BYTE 4,0,5,0,5,0,5,0,5,5,3,5,5,0,5,5,0

070776 DF350=DF347

070616 DF351=DF227

070776 DF352=DF347

070776 DF353=DF347

070776 DF354=DF347

070776 DF355=DF347

070606 DF356=DF225

070606 DF357=DF225

070616 DF360=DF227

070606 DF361=DF225

000000 DF362=0

000000 DF363=0

000000 DF364=0

000000 DF365=0

000000 DF366=0

000000 DF367=0

000000 DF370=0

000000 DF371=0

000000 DF372=0

000000 DF373=0

000000 DF374=0

000000 DF375=0

000000 DF376=0

000000 DF377=0

000000 DF400=0

070606 DF401=DF225

070606 DF402=DF225

070616 DF403=DF227

070616 DF404=DF227

070606 DF405=DF225  
 070606 DF406=DF225  
 070616 DF407=DF227  
 070616 DF410=DF227  
 070606 DF411=DF225  
 070606 DF412=DF225  
 070616 DF413=DF227  
 070616 DF414=DF227  
 070606 DF415=DF225  
 070606 DF416=DF225  
 070616 DF417=DF227  
 070616 DF420=DF227  
 070606 DF421=DF225  
 070606 DF422=DF225  
 070616 DF423=DF227  
 070616 DF424=DF227  
 070606 DF425=DF225  
 070606 DF426=DF225  
 070616 DF427=DF227  
 070616 DF430=DF227  
 070616 DF431=DF227  
 070606 DF432=DF225  
 070606 DF433=DF225  
 070616 DF434=DF227  
 070616 DF435=DF227  
 070616 DF436=DF227  
 070606 DF437=DF225  
 070606 DF440=DF225

071017 004 000 005 DF441: .BYTE 4,0,5,0,5,0  
 071017 DF442=DF441  
 071017 DF443=DF441

071026 .EVEN

;THESE ARE THE ERROR MESSAGE DATA TABLES:

071026 001232 001234 043005 DT1: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP3,\$TMP4,0  
 071046 001232 001234 043005 DT2: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP3,\$TAB,\$TMP5,0  
 071070 001232 001234 043005 DT3: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP4,\$TAB,\$TMP6,0  
 071112 001232 001234 043005 DT4: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP4,\$TAB,\$TMP3,0  
 071134 001232 001234 043005 DT5: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP4,\$TAB,\$TMP3,0  
 071152 001313 043025 001242 .WORD \$CRLF,\$MS2,\$TMP4,0  
 071162 001232 001234 043005 DT6: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,0  
 071112 DT7=DT4  
 071134 DT10=DT5  
 071112 DT11=DT4  
 071134 DT12=DT5  
 071162 DT13=DT6  
 071162 DT14=DT6  
 071134 DT15=DT5  
 071174 001232 001234 043005 DT16: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP4,\$TAB,\$TMP3,0  
 071162 DT17=DT6  
 071174 DT20=DT16  
 071134 DT21=DT5  
 071134 DT22=DT5

```

071162 DT23=DT6
071174 DT24=DT16
071134 DT25=DT5
071162 DT26=DT6
071174 DT27=DT16
071134 DT30=DT5
071162 DT31=DT6
071174 DT32=DT16
071134 DT33=DT5
071162 DT34=DT6
071174 DT35=DT16
071134 DT36=DT5
071216 001232 001234 043005 DT37: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TAB,$TMP10,$CRLF
071240 043065 001240 001313 .WORD MS4,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
071216 DT40=DT37
071262 001232 001234 043005 DT41: .WORD $TMP0,$TMP1,$TAB,$TMP2,$TAB,$TMP7,$TMP11,$TMP12
071302 001313 043065 001240 .WORD $CRLF,MS4,$TMP3,$CRLF,MS1,$TMP4,$CRLF,MS2,$TMP5,0
071216 DT42=DT37
071216 DT43=DT37
071216 DT44=DT37
071216 DT45=DT37
071216 DT46=DT37
071216 DT47=DT37
071216 DT50=DT37
071216 DT51=DT37
071216 DT52=DT37
071262 DT53=DT41
071216 DT54=DT37
071216 DT55=DT37
071216 DT56=DT37
071216 DT57=DT37
071216 DT60=DT37
071216 DT61=DT37
071174 DT62=DT16
071174 DT63=DT16
071134 DT64=DT5
071174 DT65=DT16
071112 DT66=DT4
071112 DT67=DT4
071112 DT70=DT4
071112 DT176=DT4
071112 DT177=DT4
071326 001232 001234 043005 DT71: .WORD $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS3,$TMP3,$CRLF,MS1
071350 001244 001313 043025 .WORD $TMP5,$CRLF,MS2,$TMP4,0
071112 DT72=DT4
071162 DT107=DT6
071362 001232 001234 043005 DT73: .WORD $TMP0,$TMP1,$TAB,$TMP2,$CRLF,MS4,$TMP4
071400 001313 043007 001244 .WORD $CRLF,MS1,$TMP5,$CRLF,MS2,$TMP3,0
071112 DT74=DT4
071046 DT75=DT2
071162 DT76=DT6
071362 DT77=DT73
071112 DT100=DT4
071046 DT101=DT2
071162 DT102=DT6
071362 DT103=DT73

```



071112	DT104=DT4
071046	DT105=DT2
071162	DT106=DT6
071362	DT110=DT73
071112	DT111=DT4
071046	DT112=DT2
071162	DT113=DT6
071362	DT114=DT73
071112	DT115=DT4
071046	DT116=DT2
071162	DT117=DT6
071362	DT120=DT73
071112	DT121=DT4
071046	DT122=DT2
071162	DT123=DT6
071362	DT124=DT73
071112	DT125=DT4
071046	DT126=DT2
071162	DT127=DT6
071362	DT130=DT73
071046	DT131=DT2
071162	DT132=DT6
071362	DT133=DT73
071046	DT134=DT2
071134	DT135=DT5
071134	DT136=DT5
071174	DT137=DT16
071134	DT140=DT5
071112	DT141=DT4
071112	DT142=DT4
071134	DT143=DT5
071112	DT144=DT4
071112	DT145=DT4
071134	DT146=DT5
071112	DT147=DT4
071112	DT150=DT4
071134	DT151=DT5
071112	DT152=DT4
071112	DT153=DT4
071134	DT154=DT5
071112	DT155=DT4
071112	DT156=DT4
071134	DT157=DT5
071112	DT160=DT4
071112	DT161=DT4
071134	DT162=DT5
071112	DT163=DT4
071112	DT164=DT4
071112	DT215=DT4
071134	DT216=DT5
071112	DT217=DT4
071112	DT220=DT4
071112	DT221=DT4
071134	DT222=DT5
071112	DT223=DT4
071112	DT224=DT4

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

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

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

071216 DT265=DT37  
 071216 DT266=DT37  
 071216 DT267=DT37  
 071216 DT270=DT37  
 071216 DT271=DT37  
 071216 DT272=DT37  
 071216 DT273=DT37  
 071216 DT274=DT37  
 071216 DT275=DT37  
 071216 DT276=DT37  
 071216 DT277=DT37  
 071216 DT300=DT37

071416 001232 001234 043005 DT301: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TAB,\$TMP10  
 071436 001313 043047 001240 .WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4  
 071452 001313 043007 001246 .WORD \$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0  
 071416 DT302=DT301  
 071470 001232 001234 043005 DT303: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TMP11,\$TMP12  
 071510 001313 043047 001240 .WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS11,\$TMP4  
 071524 001313 043007 001246 .WORD \$CRLF,MS1,\$TMP6,\$CRLF,MS2,\$TMP5,0  
 071416 DT304=DT301  
 071416 DT305=DT301  
 071416 DT306=DT301  
 071416 DT307=DT301  
 071416 DT310=DT301  
 071416 DT311=DT301  
 071416 DT312=DT301  
 071416 DT313=DT301  
 071416 DT314=DT301  
 071416 DT315=DT301  
 071416 DT316=DT301  
 071416 DT317=DT301  
 071416 DT320=DT301  
 071416 DT321=DT301

071542 001232 001234 043005 DT322: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TAB,\$TMP10  
 071562 001313 043047 001240 .WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0  
 071542 DT323=DT322  
 071606 001232 001234 043005 DT324: .WORD \$TMP0,\$TMP1,\$TAB,\$TMP2,\$TAB,\$TMP7,\$TMP11,\$TMP12  
 071626 001313 043047 001240 .WORD \$CRLF,MS10,\$TMP3,\$CRLF,MS1,\$TMP4,\$CRLF,MS2,\$TMP5,0  
 071542 DT325=DT322  
 071542 DT326=DT322  
 071542 DT327=DT322  
 071542 DT330=DT322  
 071542 DT331=DT322  
 071542 DT332=DT322  
 071542 DT333=DT322  
 071542 DT334=DT322  
 071542 DT335=DT322  
 071542 DT336=DT322  
 071542 DT337=DT322  
 071542 DT340=DT322  
 071542 DT341=DT322  
 071542 DT342=DT322  
 071542 DT343=DT322  
 071542 DT344=DT322

071542	DT345=DT322
071542	DT346=DT322
071542	DT347=DT322
071542	DT350=DT322
071162	DT351=DT6
071542	DT352=DT322
071542	DT353=DT322
071542	DT354=DT322
071542	DT355=DT322
071046	DT356=DT2
071070	DT357=DT3
071162	DT360=DT6
071416	DT361=DT302

000000	DT362=0
000000	DT363=0
000000	DT364=0
000000	DT365=0
000000	DT366=0
000000	DT367=0
000000	DT370=0
000000	DT371=0
000000	DT372=0
000000	DT373=0
000000	DT374=0
000000	DT375=0
000000	DT376=0
000000	DT377=0
000000	DT400=0

071112	DT401=DT4
071112	DT402=DT4
071162	DT403=DT6
071162	DT404=DT6
071112	DT405=DT4
071112	DT406=DT4
071162	DT407=DT6
071162	DT410=DT6
071112	DT411=DT4
071112	DT412=DT4
071162	DT413=DT6
071162	DT414=DT6
071112	DT415=DT4
071112	DT416=DT4
071162	DT417=DT6
071162	DT420=DT6
071112	DT421=DT4
071112	DT422=DT4
071162	DT423=DT6
071162	DT424=DT6
071112	DT425=DT4
071112	DT426=DT4
071162	DT427=DT6
071162	DT430=DT6
071162	DT431=DT6

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

000001

;12345

.END

AABBF0	013610	AMAMS2=	000000	BIT8	=	000400	DF122	=	070467	DF200	=	070527		
AABDON	013640	AMAMS3=	000000	BIT9	=	001000	DF123	=	070511	DF201	=	070527		
AABTP1	013620	AMAMS4=	000000	BPTVEC=	000014		DF124	=	070571	DF202	=	070527		
AABTP2	013630	AMSGAD=	000000	CCBDON	013740		DF125	=	070467	DF203	=	070527		
AAB1	013432	AMSGLG=	000000	CCB1	013644		DF126	=	070467	DF204	=	070527		
AAB2	013534	AMSGTY=	000000	CCB10	013706		DF127	=	070511	DF205	=	070527		
AAB3	013554	AMTYP1=	000000	CCB15	013724		DF13	=	070511	DF206	=	070527		
AAB4	013572	AMTYP2=	000000	CCB2	013662		DF130	=	070571	DF207	=	070527		
AACDON	023762	AMTYP3=	000000	CKSWR	=	104406	DF131	=	070467	DF21	=	070515		
AACTP1	023666	AMTYP4=	000000	CNT	=	000444	DF132	=	070511	DF210	=	070527		
AAC1	023606	APASS	=	000000	CPSPUR	042620	DF133	=	070571	DF211	=	070527		
AAC10	023672	APRIOR=	000000	CPTWO	042640		DF134	=	070467	DF212	=	070527		
AAC11	023710	APTCSU=	000040	CR	=	000015	DF135	=	070515	DF213	=	070527		
AAC2	023642	APTENV=	000001	CRLF	=	000200	DF136	=	070515	DF214	=	070527		
AAC20	023726	APTSIZ=	000200	DDBBF0	014236		DF137	=	070467	DF215	=	070467		
ABASE	=	000000	APTSPO=	000100	DDBDON	014256	DF14	=	070511	DF216	=	070515		
ACDW1	=	000000	ASWREG=	000000	DDBTP1	014216	DF140	=	070515	DF217	=	070467		
ACDW2	=	000000	ATESTN=	000000	DDBTP2	014226	DF141	=	070467	DF22	=	070515		
ACPUOP=	000000	AUNIT	=	000000	DDBTP3	014246	DF142	=	070467	DF220	=	070467		
AC0	=	000000	AUSWR	=	000000	DDB1	014044	DF143	=	070515	DF221	=	070467	
AC1	=	000000	AVECT1=	000000	DDB2	014110	DF144	=	070467	DF222	=	070515		
AC2	=	000000	AVECT2=	000000	DDB5	014152	DF145	=	070467	DF223	=	070467		
AC3	=	000000	BBCDON	024140	DDB6	014200	DF146	=	070515	DF224	=	070467		
AC4	=	000000	BBCTP1	024046	DDCDON	024334	DF147	=	070467	DF225	=	070606		
AC5	=	000000	BBC1	023766	DDCTP1	024234	DF15	=	070515	DF226	=	070606		
AC6	=	000000	BBC10	024052	DDC1	024144	DF150	=	070467	DF227	=	070616		
AC7	=	000000	BBC11	024070	DDC10	024246	DF151	=	070515	DF23	=	070511		
ADDW0	=	000000	BBC2	024022	DDC11	024264	DF152	=	070467	DF230	=	070606		
ADDW1	=	000000	BBC20	024106	DDC2	024202	DF153	=	070467	DF231	=	070606		
ADDW10=	000000	BIT0	=	000001	DDC20	024302	DF154	=	070515	DF232	=	070616		
ADDW11=	000000	BIT00	=	000001	DDISP	=	177570	DF155	=	070467	DF233	=	070606	
ADDW12=	000000	BIT01	=	000002	DF1	=	070460	DF156	=	070467	DF234	=	070606	
ADDW13=	000000	BIT02	=	000004	DF10	=	070477	DF157	=	070515	DF235	=	070616	
ADDW14=	000000	BIT03	=	000010	DF100	=	070467	DF16	=	070467	DF236	=	070606	
ADDW15=	000000	BIT04	=	000020	DF101	=	070467	DF160	=	070467	DF237	=	070606	
ADDW2	=	000000	BIT05	=	000040	DF102	=	070511	DF161	=	070467	DF24	=	070467
ADDW3	=	000000	BIT06	=	000100	DF103	=	070571	DF162	=	070515	DF240	=	070616
ADDW4	=	000000	BIT07	=	000200	DF104	=	070467	DF163	=	070467	DF241	=	070606
ADDW5	=	000000	BIT08	=	000400	DF105	=	070467	DF164	=	070467	DF242	=	070606
ADDW6	=	000000	BIT09	=	001000	DF106	=	070511	DF165	=	070527	DF243	=	070616
ADDW7	=	000000	BIT1	=	000002	DF107	=	070511	DF166	=	070527	DF244	=	070606
ADDW8	=	000000	BIT10	=	002000	DF11	=	070467	DF167	=	070527	DF245	=	070606
ADDW9	=	000000	BIT11	=	004000	DF110	=	070571	DF17	=	070511	DF246	=	070606
ADEVCT=	000000	BIT12	=	010000	DF111	=	070467	DF170	=	070527	DF247	=	070616	
ADEVM	=	000000	BIT13	=	020000	DF112	=	070467	DF171	=	070527	DF25	=	070515
AENV	=	000000	BIT14	=	040000	DF113	=	070511	DF172	=	070527	DF250	=	070606
AENVM	=	000000	BIT15	=	100000	DF114	=	070571	DF173	=	070550	DF251	=	070606
AFATAL=	000000	BIT2	=	000004	DF115	=	070467	DF174	=	070550	DF252	=	070606	
AMADR1=	000000	BIT3	=	000010	DF116	=	070467	DF175	=	070550	DF253	=	070616	
AMADR2=	000000	BIT4	=	000020	DF117	=	070511	DF176	=	070467	DF254	=	070606	
AMADR3=	000000	BIT5	=	000040	DF12	=	070515	DF177	=	070467	DF255	=	070616	
AMADR4=	000000	BIT6	=	000100	DF120	=	070571	DF2	=	070467	DF256	=	070606	
AMAMS1=	000000	BIT7	=	000200	DF121	=	070467	DF20	=	070467	DF257	=	070606	

DF26 = 070511	DF336 = 070734	DF414 = 070616	DF74 = 070467	DH152 = 067555
DF260 = 070622	DF337 = 070734	DF415 = 070606	DF75 = 070467	DH153 = 067375
DF261 = 070622	DF34 = 070511	DF416 = 070606	DF76 = 070511	DH154 = 067644
DF262 = 070622	DF340 = 070734	DF417 = 070616	DF77 = 070571	DH155 = 067555
DF263 = 070622	DF341 = 070734	DF42 = 070527	DH1 = 067322	DH156 = 067375
DF264 = 070622	DF342 = 070734	DF420 = 070616	DH10 = 067644	DH157 = 067644
DF265 = 070622	DF343 = 070734	DF421 = 070606	DH100 = 067555	DH16 = 067745
DF266 = 070622	DF344 = 070734	DF422 = 070606	DH101 = 067375	DH160 = 067555
DF267 = 070622	DF345 = 070734	DF423 = 070616	DH102 = 067705	DH161 = 067375
DF27 = 070467	DF346 = 070734	DF424 = 070616	DH103 = 067644	DH162 = 067644
DF270 = 070622	DF347 = 070776	DF425 = 070606	DH104 = 067555	DH163 = 067375
DF271 = 070622	DF35 = 070467	DF426 = 070606	DH105 = 067375	DH164 = 067745
DF272 = 070622	DF350 = 070776	DF427 = 070616	DH106 = 067705	DH165 = 070034
DF273 = 070643	DF351 = 070616	DF43 = 070527	DH107 = 067705	DH166 = 070034
DF274 = 070643	DF352 = 070776	DF430 = 070616	DH11 = 067555	DH167 = 070034
DF275 = 070643	DF353 = 070776	DF431 = 070616	DH110 = 067644	DH17 = 067705
DF276 = 070643	DF354 = 070776	DF432 = 070606	DH111 = 067555	DH170 = 070034
DF277 = 070643	DF355 = 070776	DF433 = 070606	DH112 = 067375	DH171 = 070034
DF3 = 070467	DF356 = 070606	DF434 = 070616	DH113 = 067705	DH172 = 070034
DF30 = 070515	DF357 = 070606	DF435 = 070616	DH114 = 067644	DH173 = 070124
DF300 = 070643	DF36 = 070515	DF436 = 070616	DH115 = 067555	DH174 = 070124
DF301 = 070664	DF360 = 070616	DF437 = 070606	DH116 = 067375	DH175 = 070124
DF302 = 070664	DF361 = 070606	DF44 = 070527	DH117 = 067705	DH176 = 067375
DF303 = 070710	DF362 = 000000	DF440 = 070606	DH12 = 067644	DH177 = 067465
DF304 = 070664	DF363 = 000000	DF441 = 071017	DH120 = 067644	DH2 = 067375
DF305 = 070664	DF364 = 000000	DF442 = 071017	DH121 = 067555	DH20 = 067555
DF306 = 070664	DF365 = 000000	DF443 = 071017	DH122 = 067375	DH200 = 070034
DF307 = 070664	DF366 = 000000	DF45 = 070527	DH123 = 067705	DH201 = 070034
DF31 = 070511	DF367 = 000000	DF46 = 070527	DH124 = 067644	DH202 = 070034
DF310 = 070664	DF37 = 070527	DF47 = 070527	DH125 = 067555	DH203 = 070034
DF311 = 070664	DF370 = 000000	DF5 = 070477	DH126 = 067375	DH204 = 070034
DF312 = 070664	DF371 = 000000	DF50 = 070527	DH127 = 067705	DH205 = 070034
DF313 = 070664	DF372 = 000000	DF51 = 070527	DH13 = 067705	DH206 = 070034
DF314 = 070664	DF373 = 000000	DF52 = 070527	DH130 = 067644	DH207 = 070034
DF315 = 070664	DF374 = 000000	DF53 = 070527	DH131 = 067375	DH21 = 067644
DF316 = 070664	DF375 = 000000	DF54 = 070527	DH132 = 067705	DH210 = 070034
DF317 = 070664	DF376 = 000000	DF55 = 070527	DH133 = 067644	DH211 = 070034
DF32 = 070467	DF377 = 000000	DF56 = 070527	DH134 = 067375	DH212 = 070034
DF320 = 070664	DF4 = 070467	DF57 = 070527	DH135 = 067644	DH213 = 070034
DF321 = 070664	DF40 = 070527	DF6 = 070511	DH136 = 067644	DH214 = 070034
DF322 = 070734	DF400 = 000000	DF60 = 070527	DH137 = 067375	DH215 = 067745
DF323 = 070734	DF401 = 070606	DF61 = 070527	DH14 = 067705	DH216 = 067644
DF324 = 070755	DF402 = 070606	DF62 = 070467	DH140 = 067644	DH217 = 067555
DF325 = 070734	DF403 = 070616	DF63 = 070467	DH141 = 067555	DH22 = 067644
DF326 = 070734	DF404 = 070616	DF64 = 070477	DH142 = 067375	DH220 = 067375
DF327 = 070734	DF405 = 070606	DF65 = 070467	DH143 = 067644	DH221 = 067745
DF33 = 070515	DF406 = 070606	DF66 = 070467	DH144 = 067555	DH222 = 067644
DF330 = 070734	DF407 = 070616	DF67 = 070467	DH145 = 067375	DH223 = 067555
DF331 = 070734	DF41 = 070550	DF7 = 070467	DH146 = 067644	DH224 = 067375
DF332 = 070734	DF410 = 070616	DF70 = 070467	DH147 = 067555	DH225 = 067555
DF333 = 070734	DF411 = 070606	DF71 = 070571	DH15 = 067644	DH226 = 067375
DF334 = 070734	DF412 = 070606	DF72 = 070467	DH150 = 067375	DH227 = 070221
DF335 = 070734	DF413 = 070616	DF73 = 070571	DH151 = 067644	DH23 = 067705

SYMBOL TABLE

DH230 = 067555	DH307 = 070034	DH366 = 000000	DH45 = 070034	DT120 = 071362
DH231 = 067375	DH31 = 067705	DH367 = 000000	DH46 = 070034	DT121 = 071112
DH232 = 070221	DH310 = 070034	DH37 = 070034	DH47 = 070034	DT122 = 071046
DH233 = 067555	DH311 = 070034	DH370 = 000000	DH5 = 067644	DT123 = 071162
DH234 = 067375	DH312 = 070034	DH371 = 000000	DH50 = 070034	DT124 = 071362
DH235 = 070221	DH313 = 070034	DH372 = 000000	DH51 = 070034	DT125 = 071112
DH236 = 067555	DH314 = 070034	DH373 = 000000	DH52 = 070034	DT126 = 071046
DH237 = 067375	DH315 = 070034	DH374 = 000000	DH53 = 070124	DT127 = 071162
DH24 = 067555	DH316 = 070034	DH375 = 000000	DH54 = 070034	DT13 = 071162
DH240 = 070221	DH317 = 070034	DH376 = 000000	DH55 = 070034	DT130 = 071362
DH241 = 067555	DH32 = 067555	DH377 = 000000	DH56 = 070034	DT131 = 071046
DH242 = 067375	DH320 = 070034	DH4 = 067555	DH57 = 070034	DT132 = 071162
DH243 = 070221	DH321 = 070034	DH40 = 070034	DH6 = 067644	DT133 = 071362
DH244 = 067555	DH322 = 070034	DH400 = 000000	DH60 = 070034	DT134 = 071046
DH245 = 067375	DH323 = 070034	DH401 = 067555	DH61 = 070034	DT135 = 071134
DH246 = 067745	DH324 = 070124	DH402 = 067375	DH62 = 067375	DT136 = 071134
DH247 = 070221	DH325 = 070034	DH403 = 067705	DH63 = 067465	DT137 = 071174
DH25 = 067644	DH326 = 070034	DH404 = 070221	DH64 = 067644	DT14 = 071162
DH250 = 067555	DH327 = 070034	DH405 = 067555	DH65 = 067375	DT140 = 071134
DH251 = 067375	DH33 = 067644	DH406 = 067375	DH66 = 067555	DT141 = 071112
DH252 = 067745	DH330 = 070034	DH407 = 067705	DH67 = 067375	DT142 = 071112
DH253 = 070221	DH331 = 070034	DH41 = 070124	DH7 = 067555	DT143 = 071134
DH254 = 067745	DH332 = 070034	DH410 = 070221	DH70 = 067465	DT144 = 071112
DH255 = 070221	DH333 = 070034	DH411 = 067555	DH71 = 067644	DT145 = 071112
DH256 = 067555	DH334 = 070034	DH412 = 067375	DH72 = 067375	DT146 = 071134
DH257 = 067375	DH335 = 070034	DH413 = 067705	DH73 = 067644	DT147 = 071112
DH26 = 067705	DH336 = 070034	DH414 = 070221	DH74 = 067555	DT15 = 071134
DH260 = 070034	DH337 = 070034	DH415 = 067555	DH75 = 067375	DT150 = 071112
DH261 = 070034	DH34 = 067705	DH416 = 067375	DH76 = 067705	DT151 = 071134
DH262 = 070034	DH340 = 070034	DH417 = 067705	DH77 = 067644	DT152 = 071112
DH263 = 070034	DH341 = 070034	DH42 = 070034	DISPLA 001142	DT153 = 071112
DH264 = 070034	DH342 = 070034	DH420 = 070221	DISPRE 000174	DT154 = 071134
DH265 = 070034	DH343 = 070034	DH421 = 067555	DSWR = 177570	DT155 = 071112
DH266 = 070034	DH344 = 070034	DH422 = 067375	DT1 = 071026	DT156 = 071112
DH267 = 070034	DH345 = 070034	DH423 = 067705	DT10 = 071134	DT157 = 071134
DH27 = 067555	DH346 = 070034	DH424 = 070221	DT100 = 071112	DT16 = 071174
DH270 = 070034	DH347 = 070034	DH425 = 067555	DT101 = 071046	DT160 = 071112
DH271 = 070034	DH35 = 067555	DH426 = 067375	DT102 = 071162	DT161 = 071112
DH272 = 070034	DH350 = 070034	DH427 = 067705	DT103 = 071362	DT162 = 071134
DH273 = 070034	DH351 = 067705	DH43 = 070034	DT104 = 071112	DT163 = 071112
DH274 = 070034	DH352 = 070034	DH430 = 070221	DT105 = 071046	DT164 = 071112
DH275 = 070034	DH353 = 070034	DH431 = 067705	DT106 = 071162	DT165 = 071216
DH276 = 070034	DH354 = 070034	DH432 = 067555	DT107 = 071162	DT166 = 071216
DH277 = 070034	DH355 = 070034	DH433 = 067375	DT11 = 071112	DT167 = 071216
DH3 = 067465	DH356 = 067555	DH434 = 067705	DT110 = 071362	DT17 = 071162
DH30 = 067644	DH357 = 070261	DH435 = 070221	DT111 = 071112	DT170 = 071216
DH300 = 070034	DH36 = 067644	DH436 = 067705	DT112 = 071046	DT171 = 071216
DH301 = 070034	DH360 = 067705	DH437 = 067555	DT113 = 071162	DT172 = 071216
DH302 = 070034	DH361 = 067375	DH44 = 070034	DT114 = 071362	DT173 = 071262
DH303 = 070124	DH362 = 000000	DH440 = 067555	DT115 = 071112	DT174 = 071262
DH304 = 070034	DH363 = 000000	DH441 = 070351	DT116 = 071046	DT175 = 071262
DH305 = 070034	DH364 = 000000	DH442 = 070417	DT117 = 071162	DT176 = 071112
DH306 = 070034	DH365 = 000000	DH443 = 070417	DT12 = 071134	DT177 = 071112



SYMBOL TABLE

DT2 = 071046	DT256 = 071112	DT334 = 071542	DT412 = 071112	DT72 = 071112
DT20 = 071174	DT257 = 071112	DT335 = 071542	DT413 = 071162	DT73 = 071362
DT200 = 071216	DT26 = 071162	DT336 = 071542	DT414 = 071162	DT74 = 071112
DT201 = 071216	DT260 = 071216	DT337 = 071542	DT415 = 071112	DT75 = 071046
DT202 = 071216	DT261 = 071216	DT34 = 071162	DT416 = 071112	DT76 = 071162
DT203 = 071216	DT262 = 071216	DT340 = 071542	DT417 = 071162	DT77 = 071362
DT204 = 071216	DT263 = 071216	DT341 = 071542	DT42 = 071216	EEBBF0 = 014412
DT205 = 071216	DT264 = 071216	DT342 = 071542	DT420 = 071162	EEBBF1 = 014422
DT206 = 071216	DT265 = 071216	DT343 = 071542	DT421 = 071112	EEBDON = 014546
DT207 = 071216	DT266 = 071216	DT344 = 071542	DT422 = 071112	EEBTP1 = 014372
DT21 = 071134	DT267 = 071216	DT345 = 071542	DT423 = 071162	EEBTP2 = 014402
DT210 = 071216	DT27 = 071174	DT346 = 071542	DT424 = 071162	EEB1 = 014262
DT211 = 071216	DT270 = 071216	DT347 = 071542	DT425 = 071112	EEB10 = 014432
DT212 = 071216	DT271 = 071216	DT35 = 071174	DT426 = 071112	EEB15 = 014466
DT213 = 071216	DT272 = 071216	DT350 = 071542	DT427 = 071162	EEB2 = 014332
DT214 = 071216	DT273 = 071216	DT351 = 071162	DT43 = 071216	EEB20 = 014514
DT215 = 071112	DT274 = 071216	DT352 = 071542	DT430 = 071162	EEB25 = 014532
DT216 = 071134	DT275 = 071216	DT353 = 071542	DT431 = 071162	EECDON = 024544
DT217 = 071112	DT276 = 071216	DT354 = 071542	DT432 = 071112	EECTP1 = 024434
DT22 = 071134	DT277 = 071216	DT355 = 071542	DT433 = 071112	EECTP2 = 024444
DT220 = 071112	DT3 = 071070	DT356 = 071046	DT434 = 071162	EEC1 = 024340
DT221 = 071112	DT30 = 071134	DT357 = 071070	DT435 = 071162	EEC10 = 024456
DT222 = 071134	DT300 = 071216	DT36 = 071134	DT436 = 071162	EEC11 = 024474
DT223 = 071112	DT301 = 071416	DT360 = 071162	DT437 = 071112	EEC2 = 024402
DT224 = 071112	DT302 = 071416	DT361 = 071416	DT44 = 071216	EEC20 = 024512
DT225 = 071112	DT303 = 071470	DT362 = 000000	DT440 = 071112	EMTVEC = 000030
DT226 = 071112	DT304 = 071416	DT363 = 000000	DT441 = 071652	EM1 = 043132
DT227 = 071162	DT305 = 071416	DT364 = 000000	DT442 = 071670	EM10 = 043506
DT23 = 071162	DT306 = 071416	DT365 = 000000	DT443 = 071670	EM100 = 047425
DT230 = 071112	DT307 = 071416	DT366 = 000000	DT45 = 071216	EM101 = 047450
DT231 = 071112	DT31 = 071162	DT367 = 000000	DT46 = 071216	EM102 = 047474
DT232 = 071162	DT310 = 071416	DT37 = 071216	DT47 = 071216	EM103 = 047535
DT233 = 071112	DT311 = 071416	DT370 = 000000	DT5 = 071134	EM104 = 047557
DT234 = 071112	DT312 = 071416	DT371 = 000000	DT50 = 071216	EM105 = 047602
DT235 = 071162	DT313 = 071416	DT372 = 000000	DT51 = 071216	EM106 = 047626
DT236 = 071112	DT314 = 071416	DT373 = 000000	DT52 = 071216	EM107 = 047214
DT237 = 071112	DT315 = 071416	DT374 = 000000	DT53 = 071262	EM11 = 043531
DT24 = 071174	DT316 = 071416	DT375 = 000000	DT54 = 071216	EM110 = 047670
DT240 = 071162	DT317 = 071416	DT376 = 000000	DT55 = 071216	EM111 = 047713
DT241 = 071112	DT32 = 071174	DT377 = 000000	DT56 = 071216	EM112 = 047737
DT242 = 071112	DT320 = 071416	DT4 = 071112	DT57 = 071216	EM113 = 047764
DT243 = 071162	DT321 = 071416	DT40 = 071216	DT6 = 071162	EM114 = 050026
DT244 = 071112	DT322 = 071542	DT400 = 000000	DT60 = 071216	EM115 = 050051
DT245 = 071112	DT323 = 071542	DT401 = 071112	DT61 = 071216	EM116 = 050075
DT246 = 071112	DT324 = 071606	DT402 = 071112	DT62 = 071174	EM117 = 050122
DT247 = 071162	DT325 = 071542	DT403 = 071162	DT63 = 071174	EM12 = 043572
DT25 = 071134	DT326 = 071542	DT404 = 071162	DT64 = 071134	EM120 = 050163
DT250 = 071112	DT327 = 071542	DT405 = 071112	DT65 = 071174	EM121 = 050205
DT251 = 071112	DT33 = 071134	DT406 = 071112	DT66 = 071112	EM122 = 050230
DT252 = 071112	DT330 = 071542	DT407 = 071162	DT67 = 071112	EM123 = 050254
DT253 = 071162	DT331 = 071542	DT41 = 071262	DT7 = 071112	EM124 = 050316
DT254 = 071112	DT332 = 071542	DT410 = 071162	DT70 = 071112	EM125 = 050341
DT255 = 071162	DT333 = 071542	DT411 = 071112	DT71 = 071326	EM126 = 050365

EM127	050412	EM205	053554	EM264	056231	EM342	063055	EM420	066055
EM13	043615	EM206	053653	EM265	056326	EM343	063157	EM421	066107
EM130	050454	EM207	053754	EM266	056417	EM344	063261	EM422	066134
EM131	050477	EM21	044020	EM267	056532	EM345	063536	EM423	066160
EM132	050524	EM210	054053	EM27	044227	EM346	063636	EM424	066226
EM133	050567	EM211	054152	EM270	056627	EM347	063734	EM425	066260
EM134	050613	EM212	054260	EM271	056670	EM35	044476	EM426	066304
EM135	050641	EM213	054361	EM272	056736	EM350	063760	EM427	066327
EM136	050714	EM214	054506	EM273	057027	EM351	064006	EM43	044743
EM137	050733	EM215	052051	EM274	057064	EM352	064112	EM430	066462
EM14 =	043615	EM216	052202	EM275	057123	EM353	064216	EM431	066513
EM140	050754	EM217	052224	EM276	057223	EM354	064322	EM432	066566
EM141	050775	EM22 =	044020	EM277	057320	EM355	064426	EM433	066613
EM142	051044	EM220	052274	EM3	043224	EM356	064532	EM434	066637
EM143	051067	EM221	052320	EM30	044271	EM357	064630	EM435	066773
EM144	051111	EM222	052452	EM300	057374	EM36	044540	EM436	067025
EM145	051161	EM223	052475	EM301	057471	EM360	064726	EM437	067102
EM146	051205	EM224	052546	EM302	057515	EM361	067156	EM44	045047
EM147	051227	EM225	054633	EM303	057543	EM362 =	000000	EM440	067130
EM15	043651	EM226	054656	EM304	057571	EM363 =	000000	EM441	067201
EM150	051277	EM227	054702	EM305	057660	EM364 =	000000	EM442	067235
EM151	051323	EM23	044043	EM306	057763	EM365 =	000000	EM443	067267
EM152	051346	EM230	054732	EM307	060150	EM366 =	000000	EM45	045147
EM153	051417	EM231	054756	EM31	044315	EM367 =	000000	EM46	045225
EM154	051444	EM232	055003	EM310	060252	EM37	044564	EM47	045331
EM155	051467	EM233	055034	EM311	060355	EM370 =	000000	EM5	043317
EM156	051540	EM234	055060	EM312	060456	EM371 =	000000	EM50	045431
EM157	051565	EM235	055105	EM313	060560	EM372 =	000000	EM51	045545
EM16	043672	EM236	055136	EM314	060661	EM373 =	000000	EM52	045571
EM160	051607	EM237	055163	EM315	060762	EM374 =	000000	EM53	045615
EM161	051701	EM24	044102	EM316	061063	EM375 =	000000	EM54	045641
EM162	051725	EM240	055211	EM317	061164	EM376 =	000000	EM55	045720
EM163	051750	EM241	055243	EM32	044353	EM377 =	000000	EM56	046046
EM164	051775	EM242	055270	EM320	061265	EM4	043257	EM57	046150
EM165	052573	EM243	055316	EM321	061366	EM40	044610	EM6	043341
EM166	052614	EM244	055350	EM322	061467	EM400 =	000000	EM60	046260
EM167	052635	EM245	055374	EM323	061524	EM401	065021	EM61	046370
EM17	043721	EM246	055421	EM324	061563	EM402	065044	EM62	046472
EM170	052656	EM247	055452	EM325	061622	EM403	065066	EM63	046576
EM171	052701	EM25	044144	EM326 =	061622	EM404	065220	EM64	046624
EM172	052724	EM250	055503	EM327	061763	EM405	065250	EM65	046700
EM173	052747	EM251	055530	EM33	044414	EM406	065274	EM66	046723
EM174	052772	EM252	055556	EM330	062065	EM407	065317	EM67	046762
EM175	053015	EM253	055610	EM331	062170	EM41	044636	EM7	043445
EM176	047106	EM254	055642	EM332	063444	EM410	065452	EM70	047063
EM177	047131	EM255	055676	EM333 =	061524	EM411	065503	EM71	047154
EM2	043171	EM256	055732	EM334	062273	EM412	065527	EM72	047173
EM20	043757	EM257	055760	EM335	062367	EM413	065552	EM73	047254
EM200	053040	EM26	044170	EM336	062471	EM414	065705	EM74	047275
EM201	053115	EM260	056007	EM337	062545	EM415	065736	EM75	047317
EM202	053216	EM261	056044	EM34	044437	EM416	065763	EM76	047342
EM203	053317	EM262	056103	EM340	062647	EM417	066007	EM77	047403
EM204	053477	EM263	056203	EM341	062751	EM42	044664	ERM10	040150

SYMBOL TABLE

ERRVEC= 000004	HHB1 015332	KKB20 016672	MMBBF1 017332	NNC10 032236
ERTYPE 042132	HHB10 015512	KKB25 016710	MMBDON 017440	NNC15 032256
ERT1 042314	HHB15 015546	KKCDON 027242	MMBTP1 017272	NNC2 032170
ERT2 042532	HHB2 015402	KKC1 025674	MMBTP2 017312	NNC20 032300
ERT3 042536	HHB20 015574	KKC10 026312	MMB1 017172	NNC25 032322
ERT4 042546	HHB25 015612	KKC11 026350	MMB10 017342	OOBDON 020052
ERT5 042560	HHC DON 025416	KKC12 026406	MMB15 017376	OOBTP1 017760
FFBBF0 014702	HHCTP1 025270	KKC13 026444	MMB2 017236	OOBTP2 017770
FFBBF1 014712	HHCTP2 025300	KKC14 026502	MMB25 017424	OOB1 017646
FFBDON 015036	HHC1 025170	KKC15 026540	MMCDON 032116	OOB10 020000
FFBTP1 014662	HHC10 025310	KKC16 026576	MMC1 030254	OOB15 020020
FFBTP2 014672	HHC11 025326	KKC17 026634	MMC10 031112	OOB2 017714
FFB1 014552	HHC2 025236	KKC18 026672	MMC11 031170	OOB20 020036
FFB10 014722	HHC20 025364	KKC19 026730	MMC12 031246	OOC DON 032602
FFB15 014756	HHC25 025344	KKC2 025732	MMC13 031324	OOC T B0 032454
FFB2 014622	HT = 000011	KKC20 026766	MMC14 031402	OOC T B1 032460
FFB20 015004	IIBBF0 015772	KKC3 025770	MMC15 031460	OOC1 032354
FFB25 015022	IIBBF1 016002	KKC4 026026	MMC16 031536	OOC10 032470
FFCDON 024746	IIBDON 016126	KKC5 026064	MMC2 030332	OOC15 032510
FFCTP1 024640	IIBTP1 015742	KKC6 026122	MMC3 030410	OOC2 032422
FFCTP2 024650	IIBTP2 015762	KKC7 026160	MMC4 030466	OOC20 032532
FFC1 024550	IIB1 015632	KKC8 026216	MMC5 030544	OOC25 032554
FFC10 024660	IIB10 016012	KKC9 026254	MMC6 030622	OODON 006734
FFC11 024676	IIB15 016046	LDCDSU 030044	MMC7 030700	OOT 006644
FFC2 024614	IIB2 015702	LDCFSU 027026	MMC8 030756	OOO1 006600
FFC20 024714	IIB20 016074	LDCT 027232	MMC9 031034	OOO2 006624
FPPUR 042564	IIB25 016112	LDXSUB 031616	MNUMBE= 000443	OOO3 006660
FPVECT= 000244	IIC DON 025532	LDXT 032106	MS1 043007	OOO4 006726
GGBBF0 015172	IIC1 025422	LF = 000012	MS10 = 043047	PIRQ = 177772
GGBBF1 015202	IIC2 025450	LLBBF0 017050	MS11 043107	PIRQVE= 000240
GGBDON 015326	IIC20 025522	LLBBF1 017060	MS2 043025	POWERM 042736
GGBTP1 015152	IIC3 025470	LLBDON 017166	MS3 043047	PPBDON 020262
GGBTP2 015162	IOTVEC= 000020	LLBTP1 017030	MS4 043065	PPBTP1 020170
GGB1 015042	JJBBF0 016266	LLBTP2 017040	NATBF1 023570	PPBTP2 020200
GGB10 015212	JJBBF1 016276	LLB1 016730	NATER1 023520	PPB1 020056
GGB15 015246	JJBDON 016422	LLB10 017070	NATER2 023536	PPB10 020210
GGB2 015112	JJBTP1 016244	LLB15 017124	NATER3 023552	PPB15 020230
GGB20 015274	JJBTP2 016256	LLB2 016774	NATINS 023246	PPB2 020124
GGB25 015312	JJB1 016132	LLB25 017152	NATRET 023532	PPB20 020246
GGCDON 025164	JJB10 016306	LLCDON 030250	NATSUB 023166	PPCDON 033034
GGCTP1 025044	JJB15 016342	LLC1 027246	NNBBF0 017562	PPCTB0 032706
GGC1 024752	JJB2 016202	LLC10 027774	NNBDON 017642	PPCTB1 032712
GGC10 025056	JJB20 016370	LLC2 027314	NNBTP1 017542	PPC1 032606
GGC11 025074	JJB25 016406	LLC3 027362	NNBTP2 017552	PPC10 032722
GGC2 025012	KKBBF0 016570	LLC4 027430	NNB1 017444	PPC15 032742
GGC20 025132	KKBBF1 016600	LLC5 027476	NNB10 017572	PPC2 032654
GGC25 025112	KKBDON 016724	LLC6 027544	NNB11 017622	PPC20 032764
GTSWR = 104405	KKBTP1 016540	LLC7 027612	NNB15 017626	PPC25 033006
HHBBF0 015472	KKBTP2 016560	LLC8 027660	NNB2 017470	PPPBF0 007076
HHBBF1 015502	KKB1 016426	LLC9 027726	NNCDON 032350	PPPBF1 007112
HHBDON 015626	KKB10 016610	LOOP 006576	NNCTB0 032222	PPPDON 007174
HHBTP1 015442	KKB15 016644	LPERR = 104413	NNCTB1 032226	PPPTP1 007126
HHBTP2 015462	KKB2 016476	MMBBF0 017322	NNC1 032122	PPP1 006740

PPP10	007136	RRBTP1	020630	SSC25	033760	TAB	= 000011	TST54	025420
PPP15	007156	RRBTP2	020640	SSC30	034010	TBITVE	= 000014	TST55	025534
PPP2	006756	RRBTP3	020650	SSSA1	010252	TCCBF0	025650	TST56	025672
PPP3	007020	RRB1	020510	SSSBF0	010242	TCCBF1	025660	TST57	027244
PPP4	007040	RRB10	020652	SSSDON	010410	TCCDON	025670	TST6	010412
PROGNU	= 000003	RRB15	020672	SSSTP1	010262	TCC1	025536	TST60	030252
PR0	= 000000	RRB2	020564	SSSTP2	010272	TCC2	025560	TST61	032120
PR1	= 000040	RRB20	020710	SSS1	010104	TCC3	025624	TST62	032352
PR2	= 000100	RRC DON	033542	SSS10	010302	TKVEC	= 000060	TST63	032604
PR3	= 000140	RRCTB0	033410	SSS11	010324	TPVEC	= 000064	TST64	033036
PR4	= 000200	RRCTB1	033414	SSS15	010334	TRAPVE	= 000034	TST65	033300
PR5	= 000240	RRCTB2	033424	SSS2	010162	TRTVEC	= 000014	TST66	033544
PR6	= 000300	RRC1	033302	SSS20	010352	TST1	006576	TST67	034016
PR7	= 000340	RRC10	033430	SSS25	010372	TST10	011132	TST7	010662
PS	= 177776	RRC15	033450	STACK	= 001100	TST11	011366	TST70	034302
PSW	= 177776	RRC2	033356	START	006106	TST12	011640	TST71	034406
PWRVEC	= 000024	RRC20	033472	STCDF5	013034	TST13	012474	TST72	034512
QQBDON	020504	RRC25	033514	STCDT	013316	TST14	013330	TST73	036154
QQBTP1	020402	RRRDON	010100	STCFDS	012200	TST15	013430	TST74	036224
QQBTP2	020422	RRREXP	007730	STCFT	012462	TST16	013642	TST75	036736
QQB1	020266	RRRTP1	007710	STCIBF	036142	TST17	013742	TST76	037142
QQB10	020432	RRRTP2	007720	STCSUB	035662	TST2	006736	TTBDON	021356
QQB15	020452	RRR1	007562	STKLMT	= 177774	TST20	014042	TTBTP1	021264
QQB2	020336	RRR10	007740	STXBF	036722	TST21	014260	TTBTP2	021274
QQB20	020470	RRR11	007762	STXSUB	036514	TST22	014550	TTB1	021152
QQCDON	033276	RRR12	010002	SWR	001140	TST23	015040	TTB10	021304
QQCTB0	033144	RRR15	010034	SWREG	000176	TST24	015330	TTB15	021324
QQCTB1	033150	RRR2	007640	SW0	= 000001	TST25	015630	TTB2	021220
QQCTB2	033160	RRR25	010014	SW00	= 000001	TST26	016130	TTB20	021342
QQC1	033040	RRR3	007642	SW01	= 000002	TST27	016424	TTCDON	034300
QQC10	033164	RRR4	007666	SW02	= 000004	TST3	007176	TTCTB0	034140
QQC15	033204	RSETUP	= 104412	SW03	= 000010	TST30	016726	TTCTB1	034144
QQC2	033112	R6	= %000006	SW04	= 000020	TST31	017170	TTCTB2	034154
QQC20	033226	R7	= %000007	SW05	= 000040	TST32	017442	TTC1	034020
QQC25	033250	SAVREG	= 104410	SW06	= 000100	TST33	017644	TTC10	034160
QQQBF0	007340	SPACE	043002	SW07	= 000200	TST34	020054	TTC15	034200
QQQBF1	007354	SSBDON	021146	SW08	= 000400	TST35	020264	TTC2	034076
QQQDON	007556	SSBTP1	021052	SW09	= 001000	TST36	020506	TTC20	034222
QQQTP1	007370	SSBTP2	021062	SW1	= 000002	TST37	020726	TTC25	034244
QQQ1	007200	SSBTP3	021072	SW10	= 002000	TST4	007560	TTC30	034274
QQQ10	007400	SSB1	020730	SW11	= 004000	TST40	021150	TTTA1	010546
QQQ2	007216	SSB10	021074	SW12	= 010000	TST41	021360	TTTA2	010550
QQQ20	007420	SSB15	021114	SW13	= 020000	TST42	021606	TTTA3	010552
QQQ22	007434	SSB2	021006	SW14	= 040000	TST43	022050	TTTBF0	010534
QQQ23	007466	SSB20	021132	SW15	= 100000	TST44	022322	TTTDON	010660
QQQ24	007514	SSCDON	034014	SW2	= 000004	TST45	023604	TTTTP1	010560
QQQ25	007540	SSCTB0	033660	SW3	= 000010	TST46	023764	TTT1	010414
QQQ3	007254	SSCTB1	033664	SW4	= 000020	TST47	024142	TTT10	010570
QQQ4	007302	SSC1	033546	SW5	= 000040	TST5	010102	TTT11	010612
RDCHR	= 104407	SSC10	033674	SW6	= 000100	TST50	024336	TTT15	010624
RESREG	= 104411	SSC15	033714	SW7	= 000200	TST51	024546	TTT2	010500
RESVEC	= 000010	SSC2	033616	SW8	= 000400	TST52	024750	TTT20	010642
RRBDON	020724	SSC20	033736	SW9	= 001000	TST53	025166	TTT3	010524

SYMBOL TABLE

TYPE = 104401	WWB2 022402	YYBDON 022320	\$CM1 = 000024	\$ICNT 001104
TYPOC = 104402	WWB3 022460	YYBTP1 022204	\$CM2 = 000050	\$ILLUP 042124
TYPON = 104404	WWB4 022536	YYBTP2 022214	\$CM3 = 000024	\$INTAG 001135
TYPOS = 104403	WWB5 022614	YYBTP3 022224	\$CM4 = 000024	\$ITEMB 001114
UUBDON 021604	WWB6 022672	YYB1 022052	\$CNTLG 041633	\$LF 001314
UUBTP1 021436	WWB7 022750	YYB10 022246	\$CNTLU 041626	\$LFLG 041221
UUBTP2 021510	WWB8 023026	YYB15 022266	\$CPUOP 001344	\$LOOP 037422
UUB1 021362	WWB9 023104	YYB2 022132	\$CRLF 001313	\$LPADR 001106
UUB10 021520	WWCDON 036152	YYB20 022304	\$DDW0 001402	\$LPERR 001110
UUB15 021556	WWC1 034514	YYB25 022226	\$DDW1 001404	\$MADR1 001350
UUB2 021434	WWC10 035154	YYCDON 036734	\$DDW10 001426	\$MADR2 001354
UUB20 021540	WWC11 035220	YYC1 036226	\$DDW11 001430	\$MADR3 001360
UUCBF0 034376	WWC12 035264	YYC2 036264	\$DDW12 001432	\$MADR4 001364
UUCDON 034404	WWC13 035330	YYC3 036322	\$DDW13 001434	\$MAIL 001316
UUCTP1 034364	WWC14 035374	YYC4 036360	\$DDW14 001436	\$MAMS1 001346
UUC1 034304	WWC15 035440	YYC5 036416	\$DDW15 001440	\$MAMS2 001352
UUC2 034334	WWC16 035504	YYC6 036454	\$DDW2 001406	\$MAMS3 001356
UUC3 034344	WWC17 035550	YYDON 013326	\$DDW3 001410	\$MAMS4 001362
UUUA1 011016	WWC18 035614	YY1 012476	\$DDW4 001412	\$MBADR 006074
UUUA2 011020	WWC2 034560	YY2 012552	\$DDW5 001414	\$MFLG 041220
UUUA3 011022	WWC4 034624	YY3 012626	\$DDW6 001416	\$MNEW 041651
UUUBF0 011004	WWC5 034670	YY4 012702	\$DDW7 001420	\$MSGAD 001332
UUUDON 011130	WWC6 034734	YY5 012756	\$DDW8 001422	\$MSGLG 001334
UUUTP1 011030	WWC7 035000	ZZCBF 037126	\$DDW9 001424	\$MSGTY 001316
UUU1 010664	WWC8 035044	ZZCDON 037140	\$DEVCT 001326	\$MSWR 041640
UUU10 011040	WWC9 035110	ZZC1 036740	\$DEVM 001374	\$MTYP1 001347
UUU11 011062	WWWBF0 011516	ZZC10 037100	\$DOAGN 037364	\$MTYP2 001353
UUU15 011074	WWWBF1 011536	ZZC12 037114	\$ENDAD 037354	\$MTYP3 001357
UUU2 010750	WWWDON 011636	ZZC15 037120	\$ENDCT 037176	\$MTYP4 001363
UUU20 011112	WWWTP1 011526	ZZC2 036750	\$ENULL 037430	\$MXCNT 037712
UUU3 010774	WWW1 011370	ZZC3 036776	\$ENV 001336	\$NULL 001154
VVBDON 014040	WWW10 011546	ZZC5 037074	\$ENVM 001337	\$NWTST= 000001
VVB1 013744	WWW11 011570	ZZZDON 013426	\$EOP 037142	\$OCNT 040752
VVB10 014006	WWW15 011602	ZZZ1 013332	\$EOPCT 037170	\$OMODE 040754
VVB15 014024	WWW2 011460	ZZZ10 013374	\$ERFLG 001103	\$OVER 037676
VVB2 013762	WWW20 011620	ZZZ15 013412	\$ERMAX 001115	\$PASS 001324
VVCBF0 034502	XXBDON 022046	ZZZ2 013350	\$ERROR 037714	\$PASTM 006100
VVCDON 034510	XXBTP1 021734	\$APTHD 006072	\$ERRPC 001116	\$PWRAD 042106
VVCTP1 034470	XXBTP2 021744	\$ATYC 041002	\$ERRTB 001442	\$PWRDN 041746
VVC1 034410	XXB1 021610	\$ATY1 040756	\$ERTTL 001112	\$PWRMG 042102
VVC2 034440	XXB10 021774	\$ATY3 040764	\$ESCAP 001304	\$PWRUP 042020
VVC3 034450	XXB15 022014	\$ATY4 040774	\$ETABL 001336	\$QUES 001312
VVBF0 011254	XXB2 021662	\$AUTOB 001134	\$ETEND 001442	\$RDCHR 041506
VVVDON 011364	XXB20 022032	\$BASE 001372	\$FATAL 001320	\$RDSZ = 000001
VVVTP1 011264	XXB25 021754	\$BDADR 001122	\$FFLG 041222	\$REGAD 001160
VVV1 011134	XXCDON 036222	\$BDDAT 001126	\$FILLC 001156	\$REG0 001162
VVV10 011274	XXC1 036156	\$BELL 001306	\$FILLS 001155	\$REG1 001164
VVV11 011316	XXXDON 012472	\$CDW1 001376	\$GDADR 001120	\$REG10 001202
VVV15 011330	XXX1 011642	\$CDW2 001400	\$GDDAT 001124	\$REG11 001204
VVV2 011216	XXX2 011716	\$CHARC 040524	\$GET42 037326	\$REG12 001206
VVV20 011346	XXX3 011772	\$CKSWR 041224	\$GTSWR 041274	\$REG13 001210
WWBDON 023602	XXX4 012046	\$CLR.T 037344	\$HD = 000003	\$REG14 001212
WWB1 022324	XXX5 012122	\$CMTAG 001100	\$HIBTS 006072	\$REG15 001214

\$REG16 001216	\$SCOPE 037434	\$TMP1 001234	\$TMP6 001246	\$TYPOS 040530
\$REG17 001220	\$SETUP= 000137	\$TMP10 001252	\$TMP7 001250	\$UNIT 001330
\$REG2 001166	\$STUP = 177777	\$TMP11 001254	\$TN = 000076	\$UNITM 006102
\$REG20 001222	\$SVLAD 037642	\$TMP12 001256	\$TPB 001152	\$USWR 001342
\$REG21 001224	\$SVPC = 006072	\$TMP13 001260	\$TPFLG 001157	\$VECT1 001366
\$REG22 001226	\$SWR = 177400	\$TMP14 001262	\$TPS 001150	\$VECT2 001370
\$REG23 001230	\$SWREG 001340	\$TMP15 001264	\$TRAP 041662	\$XTSTR 037446
\$REG3 001170	\$SWRMK= 000000	\$TMP16 001266	\$TRAP2 041704	\$GET4= 000001
\$REG4 001172	\$SWRMS= 000200	\$TMP17 001270	\$TRP = 000014	\$FILL 040753
\$REG5 001174	\$TAB 043005	\$TMP2 001236	\$TRPAD 041716	. = 071702
\$REG6 001176	\$TBIT 037426	\$TMP20 001272	\$TSTM 006076	.LPER 042660
\$REG7 001200	\$TERM = 000030	\$TMP21 001274	\$TSTNM 001102	.RSET 042666
\$RESRE 040210	\$TESTN 001322	\$TMP22 001276	\$TYPE 040246	.SX = 006072
\$RTNAD 037424	\$TIMES 001302	\$TMP23 001300	\$TYPEC 040460	
\$RTRN 037420	\$TKB 001146	\$TMP3 001240	\$TYPEX 040526	
\$SAVRE 040152	\$TKS 001144	\$TMP4 001242	\$TYPOC 040554	
\$SAVR6 042130	\$TMP0 001232	\$TMP5 001244	\$TYPON 040570	

. ABS. 071702 000

ERRORS DETECTED: 0

DEFAULT GLOBALS GENERATED: 0

DSKZ:DFFPCA.BIN, DSKZ:DFFPCA, SEQ/SOL\_DSKZ:DFFPCA.P11

RUN-TIME: 106 94 8 SECONDS

RUN-TIME RATIO: 1179/208=5.6

CORE USED: 31K (61 PAGES)