

KEF11-AA

KEF11-AA DIAG #2
CJKDDBO

AH-F245B-MC
FICHE 1 OF 1

MAY 1980
COPYRIGHT © 1979
MADE IN USA



A large grid of technical data, likely a diagnostic chart or data table, covering the majority of the page. The grid consists of numerous small rectangular cells, each containing text or numerical data. The text is too small to be legible in this image. The grid is organized into approximately 15 columns and 25 rows.



IDENTIFICATION

PRODUCT CODE: AC-F244B-MC
PRODUCT NAME: CJKDDBO KEF11-AA DIAG #2
DATE CREATED: NOV-79
MAINTAINER: DIAGNOSTIC ENGINEERING

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

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

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

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

PROGRAM HISTORY

<u>DATE</u>	<u>REVISION</u>	<u>REASON FOR REVISION</u>
JUNE-79	A	FIRST RELEASE
NOVEMBER-79	B	CORRECTIONS WERE MADE TO MULTI-TESTER SUPPORT CODE.

CONTENTS

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

THE TWO PROGRAMS:

CJKDCB, CJKDDB

ARE DESIGN TO DETECT AND REPORT LOGIC FAULTS IN THE F-11 MMU AND FLOATING POINT CHIP SET. THE DESIGN IS AN ATTEMPT TO REACH ALL MICRO-CODE LOCATIONS. TESTS ARE PARTITIONED INTO TWO STAND-ALONE PROGRAMS DESCRIBED BELOW.

NOTE THAT ERROR REPORTS IN THESE PROGRAMS ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS (CPU, MMU, FP1) HAVE BEEN RUN AND IN MOST CASE THAT THERE IS ONLY A SINGLE POINT FAULT EXISTS. IF THE PROGRAMS OR TESTS ARE NOT RUN IN ORDER THEN ERROR MESSAGES MAY NOT BE ACCURATE.

A. CJKDCA

CJKDCA TESTS: (FLOATING POINT TEST 1)

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)
ADDF, ADDD AND SUBD (ALL CONDITIONS NOT TESTED IN DFFPA)

CMPD AND CMPF
DIVD AND DIVF
MULD AND MULF
MODD AND MODF

B. CJKDDB

CJKDDB TESTS: (FLOATING POINT TEST 2)

STF AND STD (ALL MODES)
STCFD AND STCDF
CLRD AND CLRF
NEGF AND NEG
ABSF AND ABSD
TSTF AND TSTD
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
NEGF, ABSF AND TSTF (ALL SOURCE MODES)
LDFPS (ALL SOURCE MODES)
LDCIF AND LDCLF
LDCID AND LDCLD

LDEXP
STFPS (ALL DESTINATION MODES)
STCFL AND STCFI
STCDL AND STCDI
STEXP
STST

2. REQUIREMENTS

2.1 EQUIPMENT

A PROCESSOR USING THE DCF11-AA, KTF11-AA AND KEF11-A CHIP SET.

2.2 STORAGE

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

2.3 PRELIMINARY PROGRAMS

THESE TWO DIAGNOSTICS WILL ASSUME THAT THE BASIC CENTRAL PROCESSOR IS FAULTLESS, THEREFORE WHEN IN DOUBT RUN THE DCF11-AA PROCESSOR DIAGNOSTICS BEFORE THESE FLOATING POINT DIAGNOSTICS.

3. LOADING PROCEDURE

THE PROGRAMS WILL BE SUPPLIED ON THE 11/23 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

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 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>

6. ERRORS

6.1 SUMMARIES

WHEN AN ERROR IS ENCOUNTERED, AN ERROR MESSAGE ACCOMPANIED BY THE ERROR PC ARE TYPED. THERE ARE FOUR STANDARD ERROR MESSAGES USED, DESCRIBING THE PROBABLE CAUSE OF FAILURE, SUCH AS: PROBABLY BAD MMU CHIP; BAD FP1 CHIP; BAD HYBRID FP CHIP; FLOATING POINT ERROR.

6.2 ERROR RECOVERY

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

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

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIMES

LESS THAN 2 SECONDS FOR EACH PROGRAM ON ANY PASS.

8.2 STACK POINTER

THE STACK POINTER IS INITIALIZED TO 1100 IN EACH OF THE TWO 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.

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

EACH OF THE TWO 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 ACT, APT AND XXDP COMPATIBILITY

THESE PROGRAMS ARE FULLY COMPATIBLE WITH:

APT

ACT

XXDP MONITOR AND CHAIN PROGRAMS.

9. PROGRAM DESCRIPTION

TEST 1 STF WITH ILLEGAL ACCUMULATOR TEST

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

TEST 2 FDST MODE 1, FLOATING MODE, TEST

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

TEST 3 FDST MODE 2 TEST

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

TEST 4 FDST MODE 2, WITH GR7, TEST

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

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

THIS IS A TEST OF THE STCFD INSTRUCTION.

TEST 13 STCDF TEST

THIS IS A TEST OF THE STCDF INSTRUCTION.

TEST 14 STCFD WITH ILLEGAL ACCUMULATOR TEST

THIS TEST STCFD WITH ILLEGAL AC 6.

TEST 15 CLRD TEST

THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.

TEST 16 CLRD WITH ILLEGAL ACCUMULATOR TEST

THIS IS A TEST OF CLRD WITH ILLEGAL AC7.

TEST 17 NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST

THIS IS A TEST OF THE SPECIAL DEST FLOWS USING THE
NEGD INST WITH MODE ZERO AND ILLEGAL AC7.

TEST 20 NEGF, ABSF AND TSTF SOURCE MODE 0 TEST

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

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

TEST 22 NEGF, ABSF AND TSTF SOURCE MODE 2 TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE ABSD INSTRUCTION IS USED TO TEST MODE 2

TEST 23 NEGF, ABSF AND TSTF SOURCE MODE 4 TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE ABSD INSTRUCTION IS USED TO TEST MODE 4

TEST 24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE ABSD INSTRUCTION IS USED TO TEST MODE 3

TEST 25 NEGF, ABSF AND TSTF SOURCE MODE 5 TEST

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

TEST 26 NEGF, ABSF AND TSTF SOURCE MODE 6 TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE ABSD INSTRUCTION IS USED TO TEST MODE 6

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

TEST 30 NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST

THIS IS A TEST THE NEGF, ABSF AND TSTF SOURCE FLOWS.
THE NEGD INSTRUCTION IS USED TO TEST MODE 6

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

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

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.

TEST 33 SPECIAL DEST, MODE 1, TEST

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

TEST 34 SPECIAL DEST, MODE 2, TEST

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

TEST 35 SPECIAL DEST, MODE 4, TEST

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

TEST 36 SPECIAL DEST, MODE 3, TEST

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

TEST 37 SPECIAL DEST, MODE 5, TEST

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

TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST

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

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.

TEST 42 SPECIAL DEST, MODE 6, TEST

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

TEST 43 SPECIAL DEST, MODE 7, TEST

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

TEST 44 NEGD, ABSD AND TSTD TEST

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

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

THIS IS A TEST OF SOURCE MODE 1 USING THE LDFPS INSTR

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

THIS IS A TEST OF SOURCE MODE 2 USING THE LDFPS INSTR

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

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.

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

THIS IS A TEST OF DESTINATION MODE 1 USING THE STFPS INSTRUCTION

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

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

TEST 63 DESTINATION MODES, MODE 4 (FL=0), TEST

THIS IS A TEST OF DESTINATION MODE 4 USING THE STFPS INSTRUCTION

TEST 64 DESTINATION MODES, MODE 3 (FL=0), TEST

THIS IS A TEST OF DESTINATION MODE 3 USING THE STFPS INSTRUCTION

TEST 65 DESTINATION MODES, MODE 5 (FL=0), TEST

THIS IS A TEST OF DESTINATION MODE 5 USING THE STFPS INSTRUCTION

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

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

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

THIS IS A TEST OF DESTINATION MODE 7 USING THE STFPS INSTRUCTION

TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST

THIS IS A TEST OF DESTINATION MODE 2 USING STCOL WITH REGISTER 0

TEST 71 DESTINATION MODES, MODE 4 (FL=1), TEST

THIS IS A TEST OF DESTINATION MODE 4 USING STCDL WITH REGISTER 0

TEST 72 STCDI AND STCDL TEST

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

TEST 73 STCFL AND STCFI TEST

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

TEST 74 STEXP TEST

THIS IS A TEST OF THE STEXP INSTRUCTION

TEST 75 STST TEST

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

TEST 76 SPECIAL CASE TEST

THIS TEST IS DERIVED FROM A SET OF FORTRAN 4 BENCH MARKS. IT WAS FOUND THAT THIS CODE FAILED EVEN AFTER THE FLOATING POINT UNIT HAD PASSED ALL OF THE DIAGNOSTICS. THE HARDWARE WAS MODIFIED TO CORRECT THE ERROR BUT SINCE A TEST DEFICIENCY WAS INDICATED THIS TEST WAS ADDED. THE TEST LOADS A MIXED NUMBER IN THE FPAC AND SUBTRACTS A WHOLE NUMBER FROM IT.

TEST 77 INTERRUPTABILITY TEST

THIS TEST VERIFIES THE ABILITY OF THE KEF11-A TO BE INTERRUPTED DURING THE MICRO CODE MULTIPLY LOOP.

10. LISTING
&

67	BASIC DEFINITIONS
68	FPP REGISTER DEFINITIONS
79	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
80	COMMON TAGS
(2)	APT MAILBOX-ETABLE
(1)	ERROR POINTER TABLE
86	ACT11 HOOKS
87	APT PARAMETER BLOCK
90	INITIALIZE THE COMMON TAGS
91	TYPE PROGRAM NAME
(3)	GET VALUE FOR SOFTWARE SWITCH REGISTER
104	T1 STF WITH ILLEGAL ACCUMULATOR TEST
164	T2 FDST MODE 1, FLOATING MODE, TEST
234	T3 FDST MODE 2 TEST
327	T4 FDST MODE 2, WITH GR7, TEST
428	T5 FDST MODE 4 TEST
515	T6 FDST MODE 3 TEST
591	T7 FDST MODE 5 TEST
664	T10 FDST MODE 6, INDEX MODE, TEST
730	T11 FDST MODE 7, INDEX DEFERRED MODE, TEST
801	T12 STCFD TEST
1036	T13 STCDF TEST
1264	T14 STCFD WITH ILLEGAL ACCUMULATOR TEST
1298	T15 CLRD TEST
1363	T16 CLRD WITH ILLEGAL ACCUMULATOR TEST
1399	T17 NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
1437	T20 NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
1503	T21 NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
1582	T22 NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
1661	T23 NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
1740	T24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
1814	T25 NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
1892	T26 NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
1968	T27 NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
2046	T30 NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
2113	T31 NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST
2187	T32 SPECIAL DEST, MODE 0, TEST
2246	T33 SPECIAL DEST, MODE 1, TEST
2303	T34 SPECIAL DEST, MODE 2, TEST
2360	T35 SPECIAL DEST, MODE 4, TEST
2420	T36 SPECIAL DEST, MODE 3, TEST
2480	T37 SPECIAL DEST, MODE 5, TEST
2538	T40 SPECIAL DEST, FLOATING MODE 2, TEST
2594	T41 SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST
2652	T42 SPECIAL DEST, MODE 6, TEST
2720	T43 SPECIAL DEST, MODE 7, TEST
2793	T44 NEGD, ABSD AND TSTD TEST
3178	T45 SOURCE MODES, MODE 1 (FL=0), TEST
3233	T46 SOURCE MODES, MODE 2 (FL=0), TEST
3291	T47 SOURCE MODES, MODE 4 (FL=0), TEST
3330	T50 SOURCE MODES, MODE 3 (FL=0), TEST
3378	T51 SOURCE MODES, MODE 5 (FL=0), TEST
3427	T52 SOURCE MODES, MODE 6 (FL=0), TEST
3481	T53 SOURCE MODES, MODE 7 (FL=0), TEST
3545	T54 SOURCE MODES, MODE 2 GR7 (FL=1), TEST

3589	T55	SOURCE MODES, MODE 2 (FL=1), TEST
3635	T56	LDCIF AND LDCLF TEST
4023	T57	LDCID AND LDCLD TEST
4263	T60	LDEXP TEST
4650	T61	DESTINATION MODES, MODE 1 (FL=0), TEST
4712	T62	DESTINATION MODES, MODE 2 (FL=0), TEST
4775	T63	DESTINATION MODES, MODE 4 (FL=0), TEST
4837	T64	DESTINATION MODES, MODE 3 (FL=0), TEST
4901	T65	DESTINATION MODES, MODE 5 (FL=0), TEST
4965	T66	DESTINATION MODES, MODE 6 (FL=0), TEST
5038	T67	DESTINATION MODES, MODE 7 (FL=0), TEST
5114	T70	DESTINATION MODES, MODE 2 (FL=1), TEST
5146	T71	DESTINATION MODES, MODE 4 (FL=1), TEST
5182	T72	STCDI AND STCDL TEST
5562	T73	STCFL AND STCFI TEST
5589	T74	STEXP TEST
5775	T75	STST TEST
5843	T76	SPECIAL CASE TEST
5873	T77	INTERRUPTABILITY TEST
5929	END OF	PASS ROUTINE
5987		SCOPE HANDLER ROUTINE
5989		ERROR HANDLER ROUTINE
5991		SAVE AND RESTORE R0-R5 ROUTINES
5993		TYPE ROUTINE
5995		BINARY TO OCTAL (ASCII) AND TYPE
5997		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5999		APT COMMUNICATIONS ROUTINE
6002		TTY INPUT ROUTINE
6004		TRAP DECODER
(3)		TRAP TABLE
6009		POWER DOWN AND UP ROUTINES
6012		ERROR TYPE OUT ROUTINE
6052		FPP SPURIOUS TRAP TO 244 HANDLER
6069		CPU SPURIOUS TRAP TO 4 HANDLER
6080		CPU SPURIOUS TRAP TO 10 HANDLER
6094		SET LOOP ON ERROR ADDRESS ROUTINE
6100		FLAG RESET AND CONSOLE TEST ROUTINE

2
3
4
5
6
7
8
9
10
17
18
19
30
31
32
36
37
38
42
46
47
48
49
50
51
56
57
58
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
59
60
61
62
63
64
65
66
67
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

000443
000003

MNUMBER=443
PROGNUM=3

.LIST ME
.NLIST MD,MC,CND

.ENABL ABS

.TITLE CJKDDB KEF11-A DIAG PART 2
;*COPYRIGHT (C) NOVEMBER 1979
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
:*
;*PROGRAM BY DIAGNOSTIC ENGINEERING
:*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
:*
\$TN=1
\$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

000001
160000

FPVECT=244
\$SWR=177400
\$SWRMSK=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

001100

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

000011


```
(1) 000012 LF= 12 ::CODE FOR LINE FEED
(1) 000015 CR= 15 ::CODE FOR CARRIAGE RETURN
(1) 000200 CRLF= 200 ::CODE FOR CARRIAGE RETURN-LINE FEED
(1) 177776 PS= 177776 ::PROCESSOR STATUS WORD
(1) .EQUIV PS,PSW
(1) 177774 STKLMT= 177774 ::STACK LIMIT REGISTER
(1) 177772 PIRQ= 177772 ::PROGRAM INTERRUPT REQUEST REGISTER
(1) 177570 DSWR= 177570 ::HARDWARE SWITCH REGISTER
(1) 177570 DDISP= 177570 ::HARDWARE DISPLAY REGISTER
(1)
(1) ;*GENERAL PURPOSE REGISTER DEFINITIONS
(1) 000000 R0= %0 ::GENERAL REGISTER
(1) 000001 R1= %1 ::GENERAL REGISTER
(1) 000002 R2= %2 ::GENERAL REGISTER
(1) 000003 R3= %3 ::GENERAL REGISTER
(1) 000004 R4= %4 ::GENERAL REGISTER
(1) 000005 R5= %5 ::GENERAL REGISTER
(1) 000006 R6= %6 ::GENERAL REGISTER
(1) 000007 R7= %7 ::GENERAL REGISTER
(1) 000006 SP= %6 ::STACK POINTER
(1) 000007 PC= %7 ::PROGRAM COUNTER
(1)
(1) ;*PRIORITY LEVEL DEFINITIONS
(1) 000000 PR0= 0 ::PRIORITY LEVEL 0
(1) 000040 PR1= 40 ::PRIORITY LEVEL 1
(1) 000100 PR2= 100 ::PRIORITY LEVEL 2
(1) 000140 PR3= 140 ::PRIORITY LEVEL 3
(1) 000200 PR4= 200 ::PRIORITY LEVEL 4
(1) 000240 PR5= 240 ::PRIORITY LEVEL 5
(1) 000300 PR6= 300 ::PRIORITY LEVEL 6
(1) 000340 PR7= 340 ::PRIORITY LEVEL 7
(1)
(1) ;*'SWITCH REGISTER' SWITCH DEFINITIONS
(1) 100000 SW15= 100000
(1) 040000 SW14= 40000
(1) 020000 SW13= 20000
(1) 010000 SW12= 10000
(1) 004000 SW11= 4000
(1) 002000 SW10= 2000
(1) 001000 SW09= 1000
(1) 000400 SW08= 400
(1) 000200 SW07= 200
(1) 000100 SW06= 100
(1) 000040 SW05= 40
(1) 000020 SW04= 20
(1) 000010 SW03= 10
(1) 000004 SW02= 4
(1) 000002 SW01= 2
(1) 000001 SW00= 1
(1) .EQUIV SW09,SW9
(1) .EQUIV SW08,SW8
(1) .EQUIV SW07,SW7
(1) .EQUIV SW06,SW6
(1) .EQUIV SW05,SW5
(1) .EQUIV SW04,SW4
(1) .EQUIV SW03,SW3
```

```
(1) .EQUIV SW02,SW2
(1) .EQUIV SW01,SW1
(1) .EQUIV SW00,SW0
(1)
(1) ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1) 100000 BIT15= 100000
(1) 040000 BIT14= 40000
(1) 020000 BIT13= 20000
(1) 010000 BIT12= 10000
(1) 004000 BIT11= 4000
(1) 002000 BIT10= 2000
(1) 001000 BIT09= 1000
(1) 000400 BIT08= 400
(1) 000200 BIT07= 200
(1) 000100 BIT06= 100
(1) 000040 BIT05= 40
(1) 000020 BIT04= 20
(1) 000010 BIT03= 10
(1) 000004 BIT02= 4
(1) 000002 BIT01= 2
(1) 000001 BIT00= 1
(1) .EQUIV BIT09,BIT9
(1) .EQUIV BIT08,BIT8
(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0
(1)
(1) ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1) 000004 ERRVEC= 4 ;:TIME OUT AND OTHER ERRORS
(1) 000010 RESVEC= 10 ;:RESERVED AND ILLEGAL INSTRUCTIONS
(1) 000014 TBITVEC=14 ;: 'T' BIT
(1) 000014 TRTVEC= 14 ;:TRACE TRAP
(1) 000014 BPTVEC= 14 ;:BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 ;:INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 ;:POWER FAIL
(1) 000030 EMTVEC= 30 ;:EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC=34 ;: 'TRAP' TRAP
(1) 000060 TKVEC= 60 ;:TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 ;:TTY PRINTER VECTOR
(1) 000240 PIRQVEC=240 ;:PROGRAM INTERRUPT REQUEST VECTOR
68 .SBTTL FPP REGISTER DEFINITIONS
69 000000 AC0 =%0
70 000001 AC1 =%1
71 000002 AC2 =%2
72 000003 AC3 =%3
73 000004 AC4 =%4
74 000005 AC5 =%5
75 000006 AC6 =%6
76 000007 AC7 =%7
78
79 .SBTTL TRAP CATCHER
```


(1)
(1) 000000
(1)
(1)
(1) 000174
(1) 000174 000000
(1) 000176 000000
(1)
(1) 000200 000137 001370

. = 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

80
(1)
(2)
(1)
(1)
(1)
(1)
(1) 001100 001100
(1) 001100 000000
(1) 001102 000
(1) 001103 000
(1) 001104 000000
(1) 001106 000000
(1) 001110 000000
(1) 001112 000000
(1) 001114 000
(1) 001115 001
(1) 001116 000000
(1) 001120 000000
(1) 001122 000000
(1) 001124 000000
(1) 001126 000000
(1) 001130 000000
(1) 001132 000000
(1) 001134 000
(1) 001135 000
(1) 001136 000000
(1) 001140 177570
(1) 001142 177570
(1) 001144 177560
(1) 001146 177562
(1) 001150 177564
(1) 001152 177566
(1) 001154 000
(1) 001155 002
(1) 001156 012
(1) 001157 000
(1) 001160 000000
(1)
(3) 001162 000000
(3) 001164 000000
(3) 001166 000000
(3) 001170 000000
(3) 001172 000000
(3) 001174 000000
(3) 001176 000000
(3) 001200 000000
(3) 001202 000000
(3) 001204 000000
(3) 001206 000000
(3) 001210 000000
(3) 001212 000000
(3) 001214 000000
(3) 001216 000000
(3) 001220 000000
(3) 001222 000000

.SBTTL COMMON TAGS

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

SCMTAG: .=1100

.WORD 0
\$STNM: .BYTE 0
\$ERFLG: .BYTE 0
\$ICNT: .WORD 0
\$LPADR: .WORD 0
\$LPERR: .WORD 0
\$ERTTL: .WORD 0
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
\$DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$STPFLG: .BYTE 0
\$REGAD: .WORD 0
\$REG0: .WORD 0
\$REG1: .WORD 0
\$REG2: .WORD 0
\$REG3: .WORD 0
\$REG4: .WORD 0
\$REG5: .WORD 0
\$REG6: .WORD 0
\$REG7: .WORD 0
\$REG10: .WORD 0
\$REG11: .WORD 0
\$REG12: .WORD 0
\$REG13: .WORD 0
\$REG14: .WORD 0
\$REG15: .WORD 0
\$REG16: .WORD 0
\$REG17: .WORD 0
\$REG20: .WORD 0

::START OF COMMON TAGS

::CONTAINS THE TEST NUMBER
::CONTAINS ERROR FLAG
::CONTAINS SUBTEST ITERATION COUNT
::CONTAINS SCOPE LOOP ADDRESS
::CONTAINS SCOPE RETURN FOR ERRORS
::CONTAINS TOTAL ERRORS DETECTED
::CONTAINS ITEM CONTROL BYTE
::CONTAINS MAX. ERRORS PER TEST
::CONTAINS PC OF LAST ERROR INSTRUCTION
::CONTAINS ADDRESS OF 'GOOD' DATA
::CONTAINS ADDRESS OF 'BAD' DATA
::CONTAINS 'GOOD' DATA
::CONTAINS 'BAD' DATA
::RESERVED--NOT TO BE USED
::AUTOMATIC MODE INDICATOR
::INTERRUPT MODE INDICATOR
::ADDRESS OF SWITCH REGISTER
::ADDRESS OF DISPLAY REGISTER
::TTY KBD STATUS
::TTY KBD BUFFER
::TTY PRINTER STATUS REG. ADDRESS
::TTY PRINTER BUFFER REG. ADDRESS
::CONTAINS NULL CHARACTER FOR FILLS
::CONTAINS # OF FILLER CHARACTERS REQUIRED
::INSERT FILL CHARS. AFTER A 'LINE FEED'
::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
::CONTAINS THE ADDRESS FROM WHICH (\$REG0) WAS OBTAINED
::CONTAINS ((\$REGAD)+0)
::CONTAINS ((\$REGAD)+2)
::CONTAINS ((\$REGAD)+4)
::CONTAINS ((\$REGAD)+6)
::CONTAINS ((\$REGAD)+10)
::CONTAINS ((\$REGAD)+12)
::CONTAINS ((\$REGAD)+14)
::CONTAINS ((\$REGAD)+16)
::CONTAINS ((\$REGAD)+20)
::CONTAINS ((\$REGAD)+22)
::CONTAINS ((\$REGAD)+24)
::CONTAINS ((\$REGAD)+26)
::CONTAINS ((\$REGAD)+30)
::CONTAINS ((\$REGAD)+32)
::CONTAINS ((\$REGAD)+34)
::CONTAINS ((\$REGAD)+36)
::CONTAINS ((\$REGAD)+40)


```
(3) 001224 000000 $REG21: .WORD 0 ::CONTAINS (($REGAD)+42)
(3) 001226 000000 $REG22: .WORD 0 ::CONTAINS (($REGAD)+44)
(3) 001230 000000 $REG23: .WORD 0 ::CONTAINS (($REGAD)+46)
(3) 001232 000000 $TMP0: .WORD 0 ::USER DEFINED
(3) 001234 000000 $TMP1: .WORD 0 ::USER DEFINED
(3) 001236 000000 $TMP2: .WORD 0 ::USER DEFINED
(3) 001240 000000 $TMP3: .WORD 0 ::USER DEFINED
(3) 001242 000000 $TMP4: .WORD 0 ::USER DEFINED
(3) 001244 000000 $TMP5: .WORD 0 ::USER DEFINED
(3) 001246 000000 $TMP6: .WORD 0 ::USER DEFINED
(3) 001250 000000 $TMP7: .WORD 0 ::USER DEFINED
(3) 001252 000000 $TMP10: .WORD 0 ::USER DEFINED
(3) 001254 000000 $TMP11: .WORD 0 ::USER DEFINED
(3) 001256 000000 $TMP12: .WORD 0 ::USER DEFINED
(3) 001260 000000 $TMP13: .WORD 0 ::USER DEFINED
(3) 001262 000000 $TMP14: .WORD 0 ::USER DEFINED
(3) 001264 000000 $TMP15: .WORD 0 ::USER DEFINED
(3) 001266 000000 $TMP16: .WORD 0 ::USER DEFINED
(3) 001270 000000 $TMP17: .WORD 0 ::USER DEFINED
(3) 001272 000000 $TMP20: .WORD 0 ::USER DEFINED
(3) 001274 000000 $TMP21: .WORD 0 ::USER DEFINED
(3) 001276 000000 $TMP22: .WORD 0 ::USER DEFINED
(3) 001300 000000 $TMP23: .WORD 0 ::USER DEFINED
(1) 001302 000000 $TIMES: 0 ::MAX. NUMBER OF ITERATIONS
(1) 001304 000000 $ESCAPE:0 ::ESCAPE ON ERROR ADDRESS
(1) 001306 177607 000377 $BELL: .ASCIZ <207><377><377> ::CODE FOR BELL
(1) 001312 077 $QUES: .ASCII /?/ ::QUESTION MARK
(1) 001313 015 $CRLF: .ASCII <15> ::CARRIAGE RETURN
(1) 001314 000012 $LF: .ASCIZ <12> ::LINE FEED
(2) ::*****
(2) .SBTTL APT MAILBOX-ETABLE
(2)
(3) ::*****
(2) .EVEN
(2) 001316 $MAIL: ::APT MAILBOX
(2) 001316 000000 $MSGTY: .WORD AMSGTY ::MESSAGE TYPE CODE
(2) 001320 000000 $FATAL: .WORD AFATAL ::FATAL ERROR NUMBER
(2) 001322 000000 $TESTN: .WORD ATESTN ::TEST NUMBER
(2) 001324 000000 $PASS: .WORD APASS ::PASS COUNT
(2) 001326 000000 $DEVCT: .WORD ADEVCT ::DEVICE COUNT
(2) 001330 000000 $UNIT: .WORD AUNIT ::I/O UNIT NUMBER
(2) 001332 000000 $MSGAD: .WORD AMSGAD ::MESSAGE ADDRESS
(2) 001334 000000 $MSGLG: .WORD AMSGLG ::MESSAGE LENGTH
(2) 001336 $ETABLE: ::APT ENVIRONMENT TABLE
(2) 001336 000 $ENV: .BYTE AENV ::ENVIRONMENT BYTE
(2) 001337 000 $ENVM: .BYTE AENVM ::ENVIRONMENT MODE BITS
(2) 001340 000000 $SWREG: .WORD ASWREG ::APT SWITCH REGISTER
(2) 001342 000000 $USWR: .WORD AUSWR ::USER SWITCHES
(2) 001344 000000 $CPUOP: .WORD ACPUOP ::CPU TYPE, OPTIONS
(2) *
(2) * BITS 15-11=CPU TYPE
(2) * 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2) * 11/70=06,PDQ=07,Q=10
(2) *
(2) * BIT 10=REAL TIME CLOCK
(2) * BIT 9=FLOATING POINT PROCESSOR
(2) * BIT 8=MEMORY MANAGEMENT
(2) 001346 $ETEND:
```

CJKDDB KEF11-A DIAG PART 2
CJKDDB.P11 12-MAR-80 07:22

MACY11 30A(1052) 12-MAR-80 07:25 K 2
APT MAILBOX-ETABLE PAGE 1-6

SEQ 0023

(2)

.MEXIT


```
(1) .SBTTL ERROR POINTER TABLE
(1)
(1) ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1) ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1) ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1) ;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(1) ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1) ;* EM ;;POINTS TO THE ERROR MESSAGE
(1) ;* DH ;;POINTS TO THE DATA HEADER
(1) ;* DT ;;POINTS TO THE DATA
(1) ;* DF ;;POINTS TO THE DATA FORMAT
(1)
(1) 001346 $ERRTB:
81 ;ITEM NUMBER
82 001346 036417 .WORD EM1
83 001350 036446 .WORD EM2
84 001352 036503 .WORD EM3
85
86 .SBTTL ACT11 HOOKS
(1)
(2) ;*****
(1) ;HOOKS REQUIRED BY ACT11
(1) 001354 $SVPC= ;SAVE PC
(1) 000046 .=46
(1) 033000 $ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
(1) 000052 .=52
(1) 000000 .WORD 0 ;;2)SET LOC.52 TO ZERO
(1) 001354 .=$SVPC ;; RESTORE PC
87 .SBTTL APT PARAMETER BLOCK
(1)
(2) ;*****
(1) ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2) ;*****
(1) 001354 .SX= ;;SAVE CURRENT LOCATION
(1) 000024 .=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
(1) 000200 200 ;;FOR APT START UP
(1) 000044 .=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
(1) 000044 $APTHDR ;;POINT TO APT HEADER BLOCK
(1) 001354 .=$X ;;RESET LOCATION COUNTER
(2) ;*****
(1) ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1) ;INTERFACE SPEC.
(1)
(1) 001354 $APTHD:
(1) 001354 000000 $HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 001356 001316 $MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 001360 000002 $STMT: .WORD 2 ;;RUN TIM OF LONGEST TEST
(1) 001362 000004 $PASTM: .WORD 4 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 001364 000000 $UNITM: .WORD 0 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 001366 000014 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
88
89
90 001370 START:
(1) .SBTTL INITIALIZE THE COMMON TAGS
```

```

(1)          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
(1) 001370 012706 001100      MOV    # $CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
(1) 001374 005026             CLR    (R6)+            ;;CLEAR MEMORY LOCATION
(1) 001376 022706 001140      CMP    #SWR,R6 ;;DONE?
(1) 001402 001374             BNE    -6              ;;LOOP BACK IF NO
(1) 001404 012706 001100      MOV    #STACK,SP      ;;SETUP THE STACK POINTER
(1)          ;;INITIALIZE A FEW VECTORS
(1) 001410 012737 033132 000020  MOV    # $SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1) 001416 012737 000340 000022  MOV    #340,@#IOTVEC+2 ;;LEVEL 7
(1) 001424 012737 033412 000030  MOV    # $ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
(1) 001432 012737 000340 000032  MOV    #340,@#EMTVEC+2 ;;LEVEL 7
(1) 001440 012737 035546 000034  MOV    # $TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
(1) 001446 012737 000340 000036  MOV    #340,@#TRAPVEC+2;LEVEL 7
(1) 001454 012737 035634 000024  MOV    # $PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
(1) 001462 012737 000340 000026  MOV    #340,@#PWRVEC+2 ;;LEVEL 7
(1) 001470 016767 031242 031232  MOV    $ENDCT,$EOPCT  ;;SETUP END-OF-PROGRAM COUNTER
(1) 001476 005067 177600             CLR    $TIMES         ;;INITIALIZE NUMBER OF ITERATIONS
(1) 001502 005067 177576             CLR    $ESCAPE       ;;CLEAR THE ESCAPE ON ERROR ADDRESS
(1) 001506 112767 000001 177401     MOV    #1,$ERMAX     ;;ALLOW ONE ERROR PER TEST
(2)          ;;INITIALIZE THE 'T-BIT' TRAP VECTOR. THEN LOAD LOCATION '$RTRN', IN
(2)          ;;THE 'END-OF-PASS' ($EOP) ROUTINE, WITH A 'RTI' OR 'RTT'.
(2) 001514 012737 033102 000014     MOV    # $RTRN,@#TBITVEC ;;SET 'T' BIT VECTOR TO $RTRN
(2) 001522 012737 000340 000016     MOV    #340,@#TBITVEC+2 ;;LEVEL 7
(2) 001530 012767 000002 031344     MOV    #RTI,$RTRN    ;;SET $RTRN TO A RTI
(2) 001536 012737 001564 000010     MOV    #65$,@#RESVEC ;;TRY TO DO A RTT
(2) 001544 005046             CLR    -(SP)         ;;DUMMY PS
(2) 001546 012746 001554             MOV    #64$,-(SP)    ;;AND PC
(2) 001552 000006             RTT                ;;TRY THE RTT
(2) 001554 012767 000006 031320 64$: MOV    #RTT,$RTRN    ;;RTT IS LEGAL--SET $RTRN TO A RTT
(2) 001562 000402             BR     66$          ;;
(2) 001564 062706 000010 65$: ADD    #10,SP      ;;RTT ILLEGAL--CLEAN OFF THE STACK
(2) 001570 012737 000012 000010 66$: MOV    #RESVEC+2,@#RESVEC ;;RESTORE TRAP CATCHER
(2) 001576 005067 031306             CLR    $TBIT        ;;CLEAR 'T' BIT SWITCH
(1) 001602 012767 001602 177276     MOV    #.,$LPADR     ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1) 001610 012767 001610 177272     MOV    #.,$LPERR     ;;SETUP THE ERROR LOOP ADDRESS
(2)          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
(2)          ;;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
(2) 001616 013746 000004             MOV    @#ERRVEC,-(SP) ;;SAVE ERROR VECTOR
(2) 001622 012737 001656 000004     MOV    #67$,@#ERRVEC ;;SET UP ERROR VECTOR
(2) 001630 012767 177570 177302     MOV    #DSWR,SWR     ;;SETUP FOR A HARDWARE SWICH REGISTER
(2) 001636 012767 177570 177276     MOV    #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
(2) 001644 022777 177777 177266     CMP    #-1,@SWR     ;;TRY TO REFERENCE HARDWARE SWR
(2) 001652 001012             BNE    69$          ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
(2)          ;;AND THE HARDWARE SWR IS NOT = -1
(2) 001654 000403             BR     68$          ;;BRANCH IF NO TIMEOUT
(2) 001656 012716 001664 67$: MOV    #68$,(SP)    ;;SET UP FOR TRAP RETURN
(2) 001662 000002             RTI                ;;
(2) 001664 012767 000176 177246 68$: MOV    #SWREG,SWR    ;;POINT TO SOFTWARE SWR
(2) 001672 012767 000174 177242     MOV    #DISPREG,DISPLAY
(2) 001700 012637 000004 69$: MOV    (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR
(1)          ;;
(2) 001704 005067 177414             CLR    $PASS        ;;CLEAR PASS COUNT
(2) 001710 132767 000200 177421     BIT    #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
(2) 001716 001403             BEQ    70$          ;;YES,USE NON-APT SWITCH
(2) 001720 012767 001340 177212     MOV    # $SWREG,SWR  ;;NO,USE APT SWITCH REGISTER
(2) 001726             70$:

```


91
(2)
(2)
(2)
(2)
(2)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(2)
(2)
92
93
94
95
96
97
103
104
(3)
(4)
(4)
(4)
(3)
(2)
105
106
(1)
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

001726 005227 177777
001732 001051
001734 022737 033000 000042
001742 001445
001744 104401 002012
001750 005737 000042
001754 001012
001756 126727 177354 000001
001764 001406
001766 026727 177146 000176
001774 001005
001776 104406
002000 000403
002002 112767 000001 177124
002010
002010 000422
002056
002056
002056
000004
002060 104414
002062 005000
002064 170100
002066 012737 002124 000244
002074 012737 002102 001236
002102 174007
002104 170200
002106 010037 001240
002112 170300
002114 010037 001242
002120 104001
002122 000434

```
.SBTTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
INC #-1 ;;FIRST TIME?
BNE 71$ ;;BRANCH IF NO
CMP #SENDAD,@#42 ;;ACT-11?
BEQ 71$ ;;BRANCH IF YES
TYPE ,72$ ;;TYPE ASCIZ STRING
.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
TST @#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
BNE 73$ ;;BRANCH IF YES
CMPB $ENV,#1 ;;ARE WE RUNNING UNDER APT?
BEQ 73$ ;;BRANCH IF YES
CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
BNE 74$ ;;BRANCH IF NO
GTSWR ;;GET SOFT-SWR SETTINGS
BR 74$
73$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
74$: BR 71$ ;;GET OVER THE ASCIZ
;;72$: .ASCIZ <CRLF>*CJKDDB KEF11-A DIAGNOSTIC PART 2*<CRLF>
71$:

LOOP:

*****
*TEST 1 STF WITH ILLEGAL ACCUMULATOR TEST
*
*THIS IS A TEST OF THE STF INSTRUCTION USING ILLEGAL ACCUMULATOR 7, MODE 0.
*
*****
TST1: SCOPE

0001: LPERR R0 ;SET UP THE LOOP ON ERROR ADDRESS.
CLR R0 ;SET THE FPS.
LDFPS R0

MOV #000T,@#FPVECT ;SET UP FOR FP TRAPS.
MOV #1$,@#$TMP2

1$: STF AC0,AC7 ;THIS TEST INSTRUCTION SHOULD
;CAUSE A TRAP.

;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
0002: STFPS R0 ;GET FPS.
MOV R0,@#$TMP3 ;GET FEC.
STST R0
MOV R0,@#$TMP4

3$: ERROR 1 ;STF WITH ILLEGAL ACCUMULATOR, MODE
;0, DIDN'T TRAP. ST 765 TO ST 537.

BR 000DONE
```

```
125
126 ;TRAP TO OOOT, HERE, WHEN THE EXPECTED ERROR OCCURS.
127 002124 011600 OOOT: MOV (SP),R0 ;MAKE SURE THE ERROR OCCURRED
128 002126 022700 002104 CMP #0002,R0 ;AT THE CORRECT ADDRESS.
129 002132 001402 BEQ 0003 ;BRANCH IF TRAP ADDRESS CORRECT.
130 002134 000137 036134 JMP @#FPSPUR ;IF INCORRECT GO REPORT SPURIOUS
131 ;FP TRAP.
132
133 002140 170204 0003: STFPS R4 ;GET FPS.
134 002142 170305 STST R5 ;GET FEC.
135 002144 010437 001240 MOV R4,@#STMP3 ;SAVE DATA INCASE OF ERROR.
136 002150 010537 001242 MOV R5,@#STMP4
137 002154 012702 100000 MOV #100000,R2 ;EXPECTED FPS
138 002160 012703 000002 MOV #2,R3 ;EXPECTED FEC
139 002164 010237 001244 MOV R2,@#STMP5
140 002170 010337 001246 MOV R3,@#STMP6
141 002174 022626 CMP (SP)+,(SP)+ ;RESET THE STACK.
142
143 002176 020204 CMP R2,R4 ;WAS FPS CORRECT?
144 002200 001402 BEQ 0004 ;BRANCH IF YES.
145 ;OTHERWISE REPORT FPS INCORRECTLY
146 002202 104001 1$: ERROR 1 ;SET AFTER USE OF ILLEGAL ACC.
147 002204 000403 BR 000DONE
148
149 002206 020305 0004: CMP R3,R5 ;WAS THE FEC CORRECT?
150 002210 001401 BEQ 000DONE ;BRANCH IF CORRECT.
151 ;OTHERWISE REPORT INCORRECT FEC
152 002212 104001 1$: ERROR 1 ;AFTER USE OF ILLEGAL ACC.
153
154 002214 000DONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) 002214 104413 ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
155
156
157
163
164 ;*****
(3) ;*TEST 2 FDST MODE 1, FLOATING MODE, TEST
(4) ;*
(4) ;*THIS IS A TEST OF THE STF INSTRUCTION USING FDST MODE 1.
(4) ;*
(3) ;*****
(2) 002216 000004 TST2: SCOPE
165
166 002220 PPP1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 002220 104414
167
168 002222 012700 177777 MOV #-1,R0 ;SET UP A BACKROUND PATTERN IN THE
169 002226 012701 002356 MOV #PPPBFO,R1 ;INPUT BUFFER.
170 002232 012702 000014 MOV #14,R2
171 002236 010021 PPP2: MOV R0,(R1)+
172 002240 077202 SOB R2,PPP2
173
```



```

174 002242 012700 000200      MOV      #200,R0          ;SET FD MODE.
175 002246 170100             LDFPS   R0
176 002250 012700 002406      MOV      #PPPTP1,R0      ;PUT TEST DATA INTO ACO.
177 002254 172410             LDD     (R0),AC0
178
179 002256 012700 002372      MOV      #PPPBF1,R0      ;FDST ADDRESS.
180 002262 005002             CLR     R2                ;CLEAR THE FPS.
181 002264 170102             LDFPS   R2
182 002266 012737 002300 001236  MOV      #PPP3,@#$TMP2
183 002274 010037 001240      MOV      R0,@#$TMP3
184
185 002300 174010             PPP3:   STF     AC0,(R0)   ;TEST INSTRUCTION.
186
187 002302 022700 002372      CMP      #PPPBF1,R0      ;WAS R0 MODIFIED DURING EXECUTION?
188 002306 001404             BEQ     PPP4              ;BRANCH IF R0 NOT MODIFIED, CORRECT.
189
190 002310 010037 001242      MOV      R0,@#$TMP4      ;OTHERWISE REPORT ERROR, R0 MODIFIED.
191 002314 104001             1$:     ERROR  1
192 002316 000456             BR      PPPDONE          ;GO TO NEXT TEST.
193
194 002320 012700 002372      PPP4:   MOV      #PPPBF1,R0 ;CHECK THE DATA IN THE OUTPUT BUFFER.
195 002324 012701 002406      MOV      #PPPTP1,R1
196 002330 022021             CMP     (R0)+,(R1)+
197 002332 001031             BNE     PPP10            ;BRANCH IF INCORRECT.
198 002334 022011             CMP     (R0)+,(R1)
199 002336 001027             BNE     PPP10            ;BRANCH IF INCORRECT.
200 002340 022720 177777      CMP     #-1,(R0)+       ;WAS FLOATING MODE USED?
201 002344 001034             BNE     PPP15            ;BRANCH IF NOT.
202 002346 022710 177777      CMP     #-1,(R0)
203 002352 001031             BNE     PPP15
204 002354 000437             BR      PPPDONE ;GO TO NEXT TEST.
205
206 002356 177777 177777 177777  PPPBF0: .WORD  -1,-1,-1,-1,-1,-1
207 002364 177777 177777 177777
208 002372 177777 177777 177777  PPPBF1: .WORD  -1,-1,-1,-1,-1,-1
209 002400 177777 177777 177777
210 002406 123456 023456      PPPTP1: .WORD  123456,23456
211 002412 034567 045671      .WORD  34567,45671
212
213 ;REPORT DATA IN OUT PUT BUFFER INCORRECT.
214 002416 012737 002406 001242  PPP10: MOV      #PPPTP1,@#$TMP4
215 002424 012737 002372 001240  MOV      #PPPBF1,@#$TMP3
216 002432 104001             1$:     ERROR  1          ;BAD DATA.
217 002434 000407             BR      PPPDONE
218
219 ;REPORT FLOATING MODE NOT USED, BUT FD FAILED.
220 002436 012737 002406 001242  PPP15: MOV      #PPPTP1,@#$TMP4
221 002444 012737 002372 001240  MOV      #PPPBF1,@#$TMP3
222 002452 104001             1$:     ERROR  1          ;ST 707 TO 245 INTO 244 (BUT FD).
223
224 002454             PPPDONE:
(1) 002454 104413             RSETUP
(1)
(1)

```

;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?).

```
(1)
(1)
225
226
227
233
234
(3)
(4)
(4)
(4)
(3)
(2) 002456 000004
235
236
237 002460
(1) 002460 104414
238
239 002462 012700 177777
240 002466 012701 002620
241 002472 012702 000014
242 002476 010021
243 002500 077202
244
245 002502 012700 000200
246 002506 170100
247 002510 012700 002650
248 002514 172410
249
250 002516 012700 002634
251 002522 005002
252 002524 170102
253 002526 012737 002534 001236
254
255 002534 174020
256
257 002536 022700 002640
258
259 002542 001407
260 002544 010037 001242
261 002550 012737 002640 001240
262 002556 104001
263 002560 000526
264 002562 012700 002634
265 002566 012701 002650
266 002572 022021
267 002574 001031
268 002576 022021
269 002600 001027
270 002602 022027 177777
271 002606 001024
272 002610 022027 177777
273 002614 001021
274 002616 000430
275
276 002620 177777 177777 177777 177777 177777 177777
QQQBF0: .WORD -1,-1,-1,-1,-1,-1

;*****
;*TEST 3 FDST MODE 2 TEST
;*
;*THIS IS A TEST OF BOTH STF AND STD WITH FDST MODE 2.
;*
;*****
TST3: SCOPE

;FIRST TEST STF.
QQQ1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.

MOV #-1,R0 ;SET UP THE OUTPUT BUFFER.
MOV #QQQBF0,R1
MOV #14,R2
QQQ2: MOV R0,(R1)+
SOB R2,QQQ2

MOV #200,R0 ;SET FD MODE.
LDFPS R0
MOV #QQQTP1,R0 ;SETUP ACO.
LDD (R0),AC0

MOV #QQQBF1,R0 ;FDST ADDRESS.
CLR R2
LDFPS R2 ;SET FPS.
MOV #QQQ3,@#STMP2

QQQ3: STF ACO,(R0)+ ;TEST INSTRUCTION.

CMP #QQQBF1+4,R0 ;WAS R0 INCREMENTED BY 4 PROPERLY?

BEQ QQQ4 ;BRANCH IF R0 CORRECT.
MOV R0,@#STMP4 ;REPORT R0 INCORRECT AFTER FDST MODE 2.
MOV #QQQBF1+4,@#STMP3
1$: ERROR 1 ;BAD CONSTANT USED OR DIDN'T GO 527 TO 642
BR QQQDONE
QQQ4: MOV #QQQBF1,R0 ;WAS THE OUTPUT DATA CORRECT?
MOV #QQQTP1,R1
CMP (R0)+,(R1)+
BNE QQQ10 ;BRANCH IF INCORRECT.
CMP (R0)+,(R1)+
BNE QQQ10 ;BRANCH IF INCORRECT.
CMP (R0)+,#-1 ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
BNE QQQ10 ;BRANCH IF INCORRECT.
CMP (R0)+,#-1
BNE QQQ10 ;BRANCH IF INCORRECT.
BR QQQ20
```



```
277 002626 177777 177777 177777
002634 177777 177777 177777
002642 177777 177777 177777
278 002650 076543
279 002652 065432
280 002654 054321
281 002656 043210
282 ;REPORT OUTPUT DATA INCORRECT:
283 002660 012737 002650 001240 QQQ10: MOV #QQQTP1,@#$TMP3
284 002666 012737 002634 001242 QQQ10: MOV #QQQBF1,@#$TMP4
285 002674 104001 1$: ERROR 1 ;BAD DATA
286 002676 000457 BR QQQDONE
287
288 ;NOW TEST STD MODE 2.
289
290 002700 QQQ20:
(1) 002700 104414 LPPER ;SET UP THE LOOP ON ERROR ADDRESS.
291 002702 012700 002620 MOV #QQQBF0,R0 ;SET UP DEFAULT INPUT DATA BUFFER.
292 002706 010001 MOV R0,R1
293 002710 012702 000014 MOV #14,R2
294 002714 010021 QQQ22: MOV R0,(R1)+
295 002716 077202 SOB R2,QQQ22
296 002720 012700 000200 MOV #200,R0 ;ENTER FLOATING DOUBLE MODE.
297 002724 170100 LDFPS R0
298 002726 012700 002650 MOV #QQQTP1,R0 ;LOAD ACO.
299 002732 172410 LDD (R0),AC0
300 002734 012700 002634 MOV #QQQBF1,R0 ;SET DESTINATION ADDRESS.
301 002740 012737 002746 001236 QQQ23: MOV #QQQ23,@#$TMP2
302 002746 174020 QQQ23: STD ACO,(R0)+ ;TEST INSTRUCTION.
303 002750 022700 002644 CMP #QQQBF1+10,R0 ;WAS R0 INCREMENTED BY 10 CORRECTLY?
304 002754 001407 BEQ QQQ24 ;BRANCH IF CORRECT.
305 002756 010037 001242 MOV R0,@#$TMP4 ;REPORT R0 INCORRECTLY INCREMENTED.
306 002762 012737 002644 001240 QQQ23: MOV #QQQBF1+10,@#$TMP3
307 002770 104001 1$: ERROR 1 ;DO NOT INCREM BY 10 BAD CONSTANT
308 002772 000421 BR QQQDONE
309 002774 012700 002634 QQQ24: MOV #QQQBF1,R0 ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
310 003000 012701 002650 MOV #QQQTP1,R1
311 003004 012702 000004 MOV #4,R2
312 003010 022021 1$: CMP (R0)+,(R1)+
313 003012 001002 BNE QQQ25 ;BRANCH IF INCORRECT.
314 003014 077203 SOB R2,1$
315 003016 000407 BR QQQDONE
316 ;REPORT DATA INCORRECT.
317 003020 012737 002650 001240 QQQ25: MOV #QQQTP1,@#$TMP3
318 003026 012737 002634 001242 QQQ25: MOV #QQQBF1,@#$TMP4
319 003034 104001 1$: ERROR 1 ;BAD DATA
320 003036 QQQDONE:
(1) 003036 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
321
327 ;*****
(3) ;*TEST 4 FDST MODE 2, WITH GR7, TEST
(4) ;*
```

(4)
(4)
(3)
(2) 003040 000004
328
329 003042
(1) 003042 104414
330 003044 012700 003122
331 003050 012701 003170
332 003054 012702 000004
333 003060 012021
334 003062 077202
335 003064 012700 000200
336 003070 170100
337 003072 012700 003200
338 003076 172410
339 003100 012737 003220 000004
340 003106 012737 003120 001236
341 003114 005001
342 003116 005004
343
344
345
346
347
348
349 003120 174027
350 003122 005201
351 003124 005201
352 003126 005201
353 003130 005201
354 003132 012700 003210
355 003136 012702 003122
356 003142 012703 000004
357 003146 022022
358 003150 001051
359 003152 077303
360 003154 005704
361 003156 001056
362 003160 022701 000003
363 003164 001053
364 003166 000474
365
366 003170 005201
367 003172 005201
368 003174 005201
369 003176 005201
370
371 003200 005204
372 003202 005204
373 003204 005204
374 003206 005204
375
376 003210 005204
377
378 003212 005201

```
;*THIS IS A TEST OF STF WITH GR7 MODE 2 OR IMMEDIATE MODE.
:
:*****
TST4:  SCOPE

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

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

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

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

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



```
379 003214 005201          INC      R1
380 003216 005201          INC      R1
381                          ;IF A FAILURE IN THE FDST FLOWS RESULTS IN AN ODD ADDRESS TRAP THROUGH
382                          ;4 TO HERE:
383 003220 011602          RRR10:  MOV      (SP),R2          ;SEE IF THE TRAP WAS BECAUSE OF AN ODD ADDRESS.
384 003222 032702 000001    BIT      #1,R2
385 003226 001005          BNE     RRR11          ;BRANCH IF YES.
386 003230 020227 003124    CMP     R2,#RRR3+2     ;SEE IF THE TRAP OCCURRED AT THE TEST INSTRUCTION.
387 003234 001412          BEQ     RRR12          ;BRANCH IF YES.
388 003236 000137 036166    JMP     @#CPSUR        ;OTHERWISE REPORT A SPURIOUS TRAP THROUGH VECTOR 4.
389                          ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
390 003242 010237 001236    RRR11:  MOV     R2,@#STMP2
391 003246 012737 003124 001240  MOV     #RRR3+2,@#STMP3
392 003254 022626          CMP     (SP)+,(SP)+
393 003256 104001          1$:    ERROR 1          ;BAD CONSTANT #2 + PC ODD ADDR.
394 003260 000437          BR     RRRDONE
395 003262 010237 001236    RRR12:  MOV     R2,@#STMP2
396 003266 022626          CMP     (SP)+,(SP)+
397 003270 104001          1$:    ERROR 1          ;ODD ADDRESS TRAP
398 003272 000432          BR     RRRDONE          ;WRONG MODE USED.
399
400                          ;REPORT DATA INCORRECT:
401 003274 012737 003122 001240  RRR25:  MOV     #RRR3,@#STMP3
402 003302 012737 003210 001242  MOV     #RRREXP,@#STMP4
403 003310 104001          1$:    ERROR 1          ;BAD DATA BUT GR7 FAIL
404 003312 000422          BR     RRRDONE
405
406                          ;REPORT PC INCORRECT MODIFIED DURING THE EXECUTION OF FDST IMMEDIATE
407                          ;MODE. THE PC SHOULD HAVE BEEN INCREMENTED BY 2 BUT IT WASN'T.
408                          ;USE R1 AND R4 TO COMPUTE THE ACTUAL ACTION THAT WAS TAKEN ON THE PC.
409 003314 012737 003124 001240  RRR15:  MOV     #RRR3+2,@#STMP3
410 003322 005704          TST     R4          ;IS R4 CLEAR.
411 003324 001404          BEQ     1$
412 003326 012737 003122 001242  MOV     #RRR3,@#STMP4
413 003334 000410          BR     2$
414 003336 012702 003124 1$:  MOV     #RRR3+2,R2
415 003342 062701 177775    ADD     #-3,R1
416 003346 006301          ASL     R1
417 003350 160102          SUB     R1,R2
418 003352 010237 001242    MOV     R2,@#STMP4
419 003356          2$:
420 003356 104001          3$:    ERROR 1          ;BAD CONSTANT PC+
421 003360          RRRDONE:
(1) 003360 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                          ;SEE IF THE USER HAS EXPRESSED
(1)                          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                          ;THE USER TYPED CONTROL G?).
422
428                          ;*****
(3)                          ;*TEST 5          FDST MODE 4 TEST
(4)                          ;*
(4)                          ;*THIS IS A TEST OF STD WITH FDST MODE 4.
(4)                          ;*
(3)                          ;*****
(2) 003362 000004    TST5:  SCOPE
```

```

429
430 003364          SSS1:  LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 003364 104414  MOV          #-1,R0          ;SET UP THE OUTPUT BUFFER.
431 003366 012700 177777  MOV          #SSSBF0,R1
432 003372 012701 003522  MOV          #10,R2
433 003376 012702 000010  1$:  MOV          R0,(R1)+
434 003402 010021  SOB          R2,1$
435 003404 077202  MOV          #200,R0          ;ENTER FLOATING DOUBLE MODE.
436 003406 012700 000200  LDFPS       R0
437 003412 170100  MOV          #SSSTP1,R0        ;SET UP ACO.
438 003414 012700 003542  LDD         (R0),ACO
439 003420 172410  MOV          #SSS10,@#ERRVECT ;SET UP FOR A TRAP TO 4.
440 003422 012737 003562 000C04  MOV          #SSS2,@#$TMP2
441 003430 012737 003442 001.36  MOV          #SSSA1,R0        ;SET UP THE DESTINATION ADDRESS.
442 003436 012700 003532
443
444 003442 174040  SSS2:  STD         ACO,-(R0)    ;TEST INSTRUCTION.
445 003444 005201  INC         R1
446 003446 020027 003522  CMP         R0,#SSSBF0        ;SEE IF R0 WAS DECREMENTED PROPERLY.
447 003452 001060  BNE        SSS15             ;BRANCH IF R0 IS INCORRECT.
448 003454 012700 003522  MOV          #SSSBF0,R0        ;WAS THE OUTPUT DATA CORRECT?
449 003460 012701 003542  MOV          #SSSTP1,R1
450 003464 012702 000004  MOV          #4,R2
451 003470 022021  1$:  CMP         (R0)+,(R1)+
452 003472 001057  BNE        SSS20             ;BRANCH IF INCORRECT.
453 003474 077203  SOB          R2,1$
454 003476 012700 177777  MOV          #-1,R0          ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
455 003502 012701 003532  MOV          #SSSA1,R1
456 003506 012702 000004  MOV          #4,R2
457 003512 020021  2$:  CMP         R0,(R1)+
458 003514 001056  BNE        SSS25             ;BRANCH IF INCORRECT.
459 003516 077203  SOB          R2,2$
460 003520 000463  BR         SSSDONE
461
462          ;THIS IS THE OUTPUT DATA BUFFER.
463 003522 177777  SSSBF0: -1
464 003524 177777  -1
465 003526 177777  -1
466 003530 177777  -1
467 003532 177777  SSSA1: -1
468 003534 177777  -1
469 003536 177777  -1
470 003540 177777  -1
471
472          ;THIS IS THE TEST DATA LOADED INTO ACO:
473 003542 147250  SSSTP1: 147250
474 003544 036147  36147
475 003546 025036  25036
476 003550 147250  147250
477 003552 177777  SSSTP2: -1
478 003554 177777  -1
479 003556 177777  -1
480 003560 177777  -1
481
482          ;IF AN ODD ADDRESS TRAP OCCURS COME HERE:
483 003562 011600  SSS10: MOV         (SP),R0    ;SEE IF THE TRAP ACCURRED ON THE TEST INSTRUCTION.
    
```



```
484 003564 020027 003444      CMP      R0,#SSS2+2
485 003570 001405              BEQ      SSS11          ;BRANCH IF YES.
486 003572 020027 003446      CMP      R0,#SSS2+4
487 003576 001402              BEQ      SSS11          ;BRANCH IF YES.
488 003600 000137 036166      JMP      @#CPSPUR      ;OTHERWISE GO REPORT A SPURIOUS TRAP THROUGH 4.
489                               ;REPORT FAILURE IN FDST FLOWS RESULTED IN AN ODD ADDRESS.
490 003604 010037 001236      SSS11:  MOV      R0,@#STMP2
491 003610 104001              2$:      ERROR    1          ;FDST FORK X ODD AD RES.
492 003612 000426              BR       SSSDONE
493
494                               ;REPORT R0 INCORRECTLY DECREMENTED.
495 003614 010037 001242      SSS15:  MOV      R0,@#STMP4
496 003620 012737 003522 001240  MOV      #SSSBFO,@#STMP3
497 003626 104001              1$:      ERROR    1          ;R0 NOT DECRE PROP
498 003630 000417              BR       SSSDONE
499
500                               ;REPORT OUTPUT DATA INCORRECT:
501 003632 012737 003522 001240  SSS20:  MOV      #SSSBFO,@#STMP3
502 003640 012737 003542 001242  MOV      #SSSTP1,@#STMP4
503 003646 104001              1$:      ERROR    1          ;BAD DATA
504 003650 000407              BR       SSSDONE
505 003652 012737 003532 001242  SSS25:  MOV      #SSSA1,@#STMP4
506 003660 012737 003552 001240  MOV      #SSSTP2,@#STMP3
507 003666 104001              1$:      ERROR    1          ;DATA BAD OUTSIDE TARGET AREA
508 003670              SSSDONE:
(1) 003670 104413              RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                               ;SEE IF THE USER HAS EXPRESSED
(1)                               ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                               ;THE USER TYPED CONTROL G?).
509
515                               ;*****
(3)                               ;*TEST 6          FDST MODE 3 TEST
(4)                               ;*
(4)                               ;*THIS IS A TEST OF FDST MODE 3 USING STD.
(4)                               ;*
(3)                               ;*****
(2) 003672 000004      TST6:   SCOPE
516
517 003674              TTT1:
(1) 003674 104414              LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
518 003676 012701 004014      MOV      #TTTBFO,R1    ;SET UP THE OUTPUT DATA BUFFER.
519 003702 012700 177777      MOV      #-1,R0
520 003706 012702 000013      MOV      #13,R2
521 003712 010021              1$:      MOV      R0,(R1)+
522 003714 077202              SOB          R2,1$
523 003716 012737 004014 004030  MOV      #TTTBFO,@#TTTA2
524 003724 012700 000200      MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE.
525 003730 170100              LDFPS        R0
526 003732 012700 004040      MOV      #TTTTP1,R0   ;SET UP ACO.
527 003736 172410              LDD          (R0),ACO
528 003740 012737 004050 000004  MOV      #TTT10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
529 003746 016737 000006 001236  MOV      TTT2,@#STMP2
530 003754 012700 004030      MOV      #TTTA2,R0    ;SET UP THE DESTINATION ADDRESS.
531
532 003760 174030              TTT2:  STD      ACO,@(R0)+ ;TEST INSTRUCTION.
```

```

533
534 003762 020027 004032      CMP      R0,#TTTA2+2      ;SEE IF R0 WAS INCREMENTED CORRECTLY.
535 003766 001046              BNE      TTT15            ;BRANCH IF INCORRECT.
536 003770 012701 004014      MOV      #TTTBFO,R1      ;CHECK THE OUTPUT DATA BUFFER.
537 003774 012702 004040      MOV      #TTTTP1,R2
538 004000 012703 000004      MOV      #4,R3
539 004004 022122      TTT3:  CMP      (R1)+,(R2)+
540 004006 001045              BNE      TTT20            ;BRANCH IF NOT CORRECT.
541 004010 077303              SOB      R3,TTT3
542 004012 000452              BR       TTTDONE
543
544      ;THIS IS THHE OUTPUT DATA BUFFER:
545 004014 177777      TTTBFO: -1
546 004016 177777              -1
547 004020 177777              -1
548 004022 177777              -1
549 004024 177777              -1
550 004026 177777      TTTA1:  -1
551 004030 004014      TTTA2:  TTTBFO
552 004032 177777      TTTA3:  -1
553 004034 177777              -1
554 004036 177777              -1
555 004040 101213      TTTTP1: 101213
556 004042 141516              141516
557 004044 071727              71727
558 004046 037475              37475
559
560      ;TRAP THROUGH VECTOR 4 TO HERE.
561 004050 011602      TTT10: MOV      (SP),R2      ;SEE IF THE TRAP ADDRESS IS THAT OF THE TEST INSTRUCTION
562 004052 020227 003762      CMP      R2,#TTT2+2
563 004056 001405              BEQ      TTT11            ;BRANCH IF YES.
564 004060 020227 003764      CMP      R2,#TTT2+4
565 004064 001402              BEQ      TTT11            ;BRANCH IF YES.
566 004066 000137 036166      JMP      @#CPSPUR        ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
567
568      ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP.
569 004072 010237 001236      TTT11: MOV      R2,@#STMP2
570 004076 022626      CMP      (SP)+,(SP)+
571 004100 104001      1$:  ERROR  1              ;BET FDST X ODD ADR
572 004102 000416      BR       TTTDONE
573
574      ;REPORT R0 INCORRECT:
575 004104 010037 001242      TTT15: MOV      R0,@#STMP4
576 004110 012737 004032 001240  MOV      #TTTA2+2,@#STMP3
577 004116 104001      1$:  ERROR  1              ;R0 NOT INCREMENT PROPERLY
578 004120 000407      BR       TTTDONE
579
580      ;REPORT INCORRECT OUTPUT DATA:
581 004122 012737 004014 001240  TTT20: MOV      #TTTBFO,@#STMP3
582 004130 012737 004040 001242  MOV      #TTTTP1,@#STMP4
583 004136 104001      1$:  ERROR  1              ;BAD DATA
584 004140      TTTDONE:
(1) 004140 104413      RSETUP              ;GO INITIALIZE THE FPS AND STACK; AND
(1)              ;SEE IF THE USER HAS EXPRESSED
(1)              ;THE DESIRE TO CHANGE THE SOFTWARE
(1)              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS

```


;THE USER TYPED CONTROL G?).

```
(1) 585
591 (3) *****
(4) *TEST 7 FDST MODE 5 TEST
(4) *
(4) *THIS IS A TEST OF FDST MODE 5 USING STD.
(3) *
(2) 004142 000004 TST7: SCOPE
592
593 004144 UUU1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 004144 104414 MOV #UUUBF0,R1 ;SET UP THE OUTPUT DATA BUFFER.
594 004146 012701 004264 MOV #-1,R0
595 004152 012700 177777 MOV #13,R2
596 004156 012702 000013 1$: MOV R0,(R1)+
597 004162 010021 SOB R2,1$
598 004164 077202 MOV #UUUBF0,@#UUUA1
599 004166 012737 004264 004276 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE.
600 004174 012700 000200 LDFPS R0
601 004200 170100 MOV #UUUTP1,R0 ;SET UP AC0.
602 004202 012700 004310 LDD (R0),AC0
603 004206 172410 MOV #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
604 004210 012737 004320 000004 MOV UUU2,@#STMP2
605 004216 016737 000006 001236 MOV #UUUA2,R0 ;SET UP THE DESTINATION ADDRESS.
606 004224 012700 004300 UUU2: STD AC0,@-(R0) ;TEST INSTRUCTION.
607 004230 174050 CMP R0,#UUUA2-2 ;WAS R0 DECREMENTED PROPERLY?
608 004232 020027 004276 BNE UUU15 ;BRANCH IF R0 IS INCORRECT.
609 004236 001046 MOV #UUUBF0,R1 ;WAS THE DATA OUTPUT CORRECTLY?
610 004240 012701 004264 MOV #UUUTP1,R2
611 004244 012702 004310 MOV #4,R3
612 004250 012703 000004 UUU3: CMP (R1)+,(R2)+
613 004254 022122 BNE UUU20 ;BRANCH IF DATA IS INCORRECT.
614 004256 001045 SOB R3,UUU3
615 004260 077303 BR UUDONE
616 004262 000452
617
618 ;THIS IS THE OUTPUT DATA BUFFER
619 004264 177777 UUUBF0: -1
620 004266 177777 -1
621 004270 177777 -1
622 004272 177777 -1
623 004274 177777 -1
624 004276 004264 UUUA1: UUUBF0
625 004300 177777 UUUA2: -1
626 004302 177777 UUUA3: -1
627 004304 177777 -1
628 004306 177777 -1
629 004310 020212 UUUTP1: 20212
630 004312 023242 23242
631 004314 026273 26273
632 004316 031323 031323
633
634 ;IF A TRAP TO 4 OCCURS COME HERE.
635 004320 011602 UUU10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
636 004322 020227 004232 CMP R2,#UUU2+2
637 004326 001405 BEQ UUU11 ;BRANCH IF YES.
```

```
638 004330 020227 004234      CMP      R2,#UUU2+4
639 004334 001402              BEQ      UUU11          ;BRANCH IF YES.
640 004336 000137 036166      JMP      @#CPSPUR      ;OTHERWISE REPORT A SPURIOUS TRAP TO 4.
641      ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
642 004342 010237 001236      UUU11:  MOV      R2,@#$TMP2
643 004346 022626              CMP      (SP)+,(SP)+
644 004350 104001              1$:     ERROR    1          ;BET FDST X ODD ADR
645 004352 000416              BR       UUUDONE
646
647      ;REPORT R0 INCORRECT.
648 004354 010037 001242      UUU15:  MOV      R0,@#$TMP4
649 004360 012737 004302 001240  MOV      #UUUA2+2,@#$TMP3
650 004366 104001              1$:     ERROR    1          ;R0 NOT INCREMENT PROPERLY
651 004370 000407              BR       UUUDONE
652
653      ;REPORT BAD DATA.
654 004372 012737 004264 001242  UUU20:  MOV      #UUUBF0,@#$TMP4
655 004400 012737 004310 001240  MOV      #UUUTP1,@#$TMP3
656 004406 104001              1$:     ERROR    1          ;BAD DATA
657 004410              UUUDONE:
(1) 004410 104413              RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                          ;SEE IF THE USER HAS EXPRESSED
(1)                          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                          ;THE USER TYPED CONTROL G?).
658
664      ;*****
(3)      ;*TEST 10      FDST MODE 6, INDEX MODE, TEST
(4)      ;*
(4)      ;*THIS IS A TEST OF FDST MODE 6, INDEX MODE, USING STD.
(4)      ;*
(3)      ;*****
(2) 004412 000004      TST10:  SCOPE
665
666 004414              VVV1:
(1) 004414 104414              LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
667 004416 012700 000200      MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE.
668 004422 170100              LDFPS         R0
669 004424 012701 004534      MOV      #VVVBFO,R1   ;SET UP THE OUT PUT DATA BUFFER.
670 004430 012700 177777      MOV      #-1,R0
671 004434 012702 000004      MOV      #4,R2
672 004440 010021              1$:     MOV      R0,(R1)+
673 004442 077202              SOB          R2,1$
674 004444 012737 004554 000004  MOV      #VVV10,@#ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR.
675 004452 012700 004544      MOV      #VVVTP1,R0   ;SET UP ACO.
676 004456 172410              LDD          (R0),ACO
677 004460 012737 004476 001236  MOV      #VVV2,@#$TMP2
678 004466 012700 176633      MOV      #VVVBFO-5701,R0 ;SET UP THE DESTINATION ADDRESS.
679 004472 012701 000001      MOV      #1,R1
680 004476 174060 005701      VVV2:  STD          ACO,5701(R0) ;TEST INSTRUCTION.
681
682 004502 020027 176633      CMP      R0,#VVVBFO-5701 ;SEE IF R0 WAS MODIFIED.
683 004506 001040              BNE          VVV15     ;BRANCH IF INCORRECT.
684 004510 012702 004534      MOV      #VVVBFO,R2   ;WAS THE OUTPUT DATA CORRECT.
685 004514 012703 004544      MOV      #VVVTP1,R3
686 004520 012704 000004      MOV      #4,R4
```



```
687 004524 022223      1$:    CMP      (R2)+,(R3)+
688 004526 001037      BNE      VVV20      ;BRANCH IF INCORRECT DATA.
689 004530 077403      SOB      R4,1$
690 004532 000444      BR       VVVDONE
691 004534 177777      VVVBF0: -1
692 004536 177777      -1
693 004540 177777      -1
694 004542 177777      -1
695 004544 030313      VVVTP1: 30313
696 004546 023334      23334
697 004550 035363      35363
698 004552 074041      74041
699
700      ;COME HERE AFTER A TRAP THROUGH VECTOR 4.
701 004554 011602      VVV10: MOV      (SP),R2      ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
702 004556 020227 004500      CMP      R2,#VVV2+2
703 004562 001405      BEQ      VVV11      ;BRANCH IF YES.
704 004564 020227 004502      CMP      R2,#VVV2+4
705 004570 001402      BEQ      VVV11      ;BRANCH IF YES.
706 004572 000137 036134      JMP      @#FPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
707      ;REPORT FAILURE OF FDST RESULTED IN AN ODD ADDRESS TRAP TO 4.
708 004576 010237 001236      VVV11: MOV      R2,@#$TMP2
709 004602 022626      CMP      (SP)+,(SP)+
710 004604 104001      1$:    ERROR   1      ;FDST FORK X ODD ADD
711 004606 000416      BR       VVVDONE
712
713      ;REPORT R0 MODIFIED.
714 004610 010037 001242      VVV15: MOV      R0,@#$TMP4
715 004614 012737 176633 001240      MOV      #VVVBF0-5701,@#$TMP3
716 004622 104001      1$:    ERROR   1      ;R0 MODIFIED!
717 004624 000407      BR       VVVDONE
718
719      ;REPORT INCORRECT DATA.
720 004626 012737 004534 001240      VVV20: MOV      #VVVBF0,@#$TMP3
721 004634 012737 004544 001242      MOV      #VVVTP1,@#$TMP4
722 004642 104001      1$:    ERROR   1      ;BAD DATA
723 004644      VVVDONE:
(1) 004644 104413      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
(1)      ;SEE IF THE USER HAS EXPRESSED
(1)      ;THE DESIRE TO CHANGE THE SOFTWARE
(1)      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)      ;THE USER TYPED CONTROL G?).
724
730      ;*****
(3)      ;*TEST 11      FDST MODE 7, INDEX DEFERRED MODE, TEST
(4)      ;*
(4)      ;*THIS IS A TEST OF FDST MODE 7, INDEX DEFERRED MODE, USING STD.
(4)      ;*
(3)      ;*****
(2) 004646 000004      TST11: SCOPE
731
732 004650      WWW1:
(1) 004650 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
733 004652 012700 000200      MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE.
734 004656 170100      LDFPS     R0
735 004660 012701 004776      MOV      #WWWBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
```

```
736 004664 012700 177777      MOV      #-1,R0
737 004670 012702 000004      MOV      #4,R2
738 004674 010021      1$:     MOV      R0,(R1)+
739 004676 077202      SOB      R2,1$
740 004700 012737 005026 000004      MOV      #WWW10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
741 004706 012700 005006      MOV      #WWWTP1,R0 ;SET UP ACO.
742 004712 172410      LDD      (R0),AC0
743 004714 012737 004740 001236      MOV      #WWW2,@#STMP2
744 004722 012700 177115      MOV      #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS.
745 004726 012701 000001      MCV      #1,R1
746 004732 012737 004776 005016      MOV      #WWWBF0,@#WWWBF1
747 004740 174070 005701      WWW2:   STD      AC0,@5701(R0) ;TEST INSTRUCTION.
748
749 004744 020027 177115      CMP      R0,#WWWBF1-5701 ;IS R0 CORRECT?
750 004750 001044      BNE      WWW15 ;BRANCH IF INCORRECT.
751 004752 012702 004776      MOV      #WWWBF0,R2 ;WAS THE DATA OUTPUT CORRECTLY?
752 004756 012703 005006      MOV      #WWWTP1,R3
753 004762 012704 000004      MOV      #4,R4
754 004766 022223      1$:     CMP      (R2)+,(R3)+
755 004770 001043      BNE      WWW20 ;BRANCH IF DATA IS INCORRECT.
756 004772 077403      SOB      R4,1$
757 004774 000450      BR       WWWDONE
758 004776 177777      WWWBF0: -1
759 005000 177777      -1
760 005002 177777      -1
761 005004 177777      -1
762 005006 041424      WWWTP1: 41424
763 005010 034445      34445
764 005012 046475      46475
765 005014 051525      WWWBF1: 051525
766 005016 177777      -1
767 005020 177777      -1
768 005022 177777      -1
769 005024 177777      -1
770
771      ;TRAP THROUGH 4 TO HERE.
772 005026 011602      WWW10: MOV      (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
773 005030 020227 004742      CMP      R2,#WWW2+2
774 005034 001405      BEQ      WWW11 ;BRANCH IF YES.
775 005036 020227 004744      CMP      R2,#WWW2+4
776 005042 001402      BEQ      WWW11 ;BRANCH IF YES.
777 005044 000137 036134      JMP      @#FPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
778      ;REPORT FAILURE OF FDST FORK RESULTED IN AN ODD ADDRESS TRAP TO 4.
779 005050 010237 001236      WWW11: MOV      R2,@#STMP2
780 005054 022626      CMP      (SP)+,(SP)+
781 005056 104001      1$:     ERROR   1 ;FDST FORK X ODD ADD
782 005060 000416      BR       WWWDONE
783
784      ;REPORT R0 MODIFIED.
785 005062 010037 001242      WWW15: MOV      R0,@#STMP4
786 005066 012737 177075 001240      MOV      #WWWBF0-5701,@#STMP3
787 005074 104001      1$:     ERROR   1 ;R0 MODIFIED!
788 005076 000407      BR       WWWDONE
789
790      ;REPORT DATA INCORRECT
791 005100 012737 004776 001240      WWW20: MOV      #WWWBF0,@#STMP3
```



```

792 005106 012737 005006 001242      MOV      #WWWTP1,@#STMP4
793 005114 104001      1$:      ERROR      1      ;BAD DATA
794 005116 104413      WWWDONE:
(1) 005116 104413      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
(1)                                     ;SEE IF THE USER HAS EXPRESSED
(1)                                     ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                                     ;THE USER TYPED CONTROL G?).
795
801                                     ;*****
(3)                                     ;*TEST 12      STCFD TEST
(4)                                     ;*
(4)                                     ;*THIS IS A TEST OF THE STCFD INSTRUCTION.
(4)                                     ;*
(3)                                     ;*****
(2) 005120 000004      TST12:  SCOPE
802
803                                     ;AC=0
804 005122 104414      XXX1:
(1) 005122 104414      LPERR      PC,STCFDS      ;SET UP THE LOOP ON ERROR ADDRESS.
805 005124 004767 000330      JSR
806 005130 000000      1$:      0      ;AC
807 005132 000000      0
808 005134 000000      0
809 005136 000000      0
810 005140 000000      2$:      0      ;RES
811 005142 000000      0
812 005144 000000      0
813 005146 000000      0
814 005150 000000      3$:      0      ;ERROR RES.
815 005152 000000      0
816 005154 177777      -1
817 005156 177777      -1
818 005160 047000      4$:      47000      ;FPS BEFORE EXECUTION.
819 005162 047004      47004      ;FPS AFTER EXECUTION.
820 005164 177777      -1      ;FEC
821 005166 147004      147004      ;ERROR FPS.
822 005170 104001      5$:      ERROR      1      ;FDFL<---FDFLXST 767
823 005172 000401      BR      6$
824 005174 104001      ERROR      1      ;BUT EZBT X ST560 TO 061 INTO 261
825 005176
826
827 005176 104414      XXX2:
(1) 005176 104414      LPERR      PC,STCFDS      ;SET UP THE LOOP ON ERROR ADDRESS.
828 005200 004767 000254      JSR
829 005204 017203      1$:      17203      ;AC
830 005206 142536      142536
831 005210 047506      47506
832 005212 172031      172031
833 005214 017203      2$:      17203      ;RES
834 005216 142536      142536
835 005220 000000      0
836 005222 000000      0
837 005224 017203      3$:      17203      ;ERROR RES.
838 005226 142536      142536
839 005230 047506      47506
    
```

```
840 005232 172031 172031
841 005234 040000 4$: 40000 ;FPS BEFORE EXECUTION.
842 005236 040000 40000 ;FPS AFTER EXECUTION.
843 005240 177777 -1 ;FEC
844 005242 177777 -1 ;ERROR FPS.
845 005244 104001 5$: ERROR 1 ;X11(1,0)<---0 X ST766
846 005246 000401 BR 6$
847 005250 104001 ERROR 1
848 005252
849
850 005252 XXX3:
(1) 005252 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
851 005254 004767 000200 JSR PC,STCFDS
852 005260 050717 1$: 50717 ;AC
853 005262 027374 27374
854 005264 075767 75767
855 005266 077071 77071
856 005270 050717 2$: 50717 ;RES
857 005272 027374 27374
858 005274 000000 0
859 005276 000000 0
860 005300 000000 3$: 0 ;ERROR RES.
861 005302 000000 0
862 005304 000000 0
863 005306 000000 0
864 005310 047000 4$: 47000 ;FPS BEFORE EXECUTION.
865 005312 047000 47000 ;FPS AFTER EXECUTION.
866 005314 177777 -1 ;FEC
867 005316 174002 174002 ;ERROR FPS.
868 005320 104001 5$: ERROR 1 ;BUT OPIC X ST251
869 005322 000401 BR 6$
870 005324 104001 ERROR 1 ;BUT EZBT X ST421
871 005326
872
873 005326 XXX4:
(1) 005326 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
874 005330 004767 000124 JSR PC,STCFDS
875 005334 020212 1$: 20212 ;AC
876 005336 032425 32425
877 005340 026272 26272
878 005342 002123 02123
879 005344 020212 2$: 20212 ;RES
880 005346 032425 32425
881 005350 000000 0
882 005352 000000 0
883 005354 020212 3$: 20212 ;ERROR RES.
884 005356 032425 32425
885 005360 100000 100000
886 005362 000000 0
887 005364 040000 4$: 40000 ;FPS BEFORE EXECUTION.
888 005366 040000 40000 ;FPS AFTER EXECUTION.
889 005370 177777 -1 ;FEC
890 005372 177777 -1 ;ERROR FPS.
891 005374 104001 5$: ERROR 1 ;BUT FD IN ROUND X ST113
892 005376 000401 BR 6$
893 005400 104001 ERROR 1
```


894 005402
 895
 896 005402
 (1) 005402 104414
 897 005404 004767 000050
 898 005410 121314
 899 005412 151617
 900 005414 101112
 901 005416 131415
 902 005420 121314
 903 005422 151617
 904 005424 000000
 905 005426 000000
 906 005430 021314
 907 005432 151617
 908 005434 000000
 909 005436 000000
 910 005440 040000
 911 005442 040010
 912 005444 177777
 913 005446 177777
 914 005450 104001
 915 005452 000401
 916 005454 104001
 917 005456 000535

```

6$:
:
XXX5:
LPERR
JSR PC,STCFDS ;SET UP THE LOOP ON ERROR ADDRESS.
1$: 121314 ;AC
    151617
    101112
    131415
2$: 121314 ;RES
    151617
    0
    0
3$: 21314 ;ERROR RES.
    151617
    0
    0
4$: 40000 ;FPS BEFORE EXECUTION.
    40010 ;FPS AFTER EXECUTION.
    -1 ;FEC
    -1 ;ERROR FPS.
5$: ERROR 1 ;BUT ENBT X ST567 OR BAD SIGN ST460
    BR 6$
    ERROR 1
6$: BR XXXDONE
  
```

918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948

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

```

JSR PC,@STCFDS
ACARG: .WORD X,X,X,X ;AC OPERAND
RES: .WORD X,X,X,X ;EXPECTED RESULT
ERRES: .WORD X,X,X,X ;ERROR RESULT
FPSB: .WORD X ;FPS BEFORE EXECUTION
FPSA: .WORD X ;FPS AFTER EXECUTION
FEC: .WORD X ;EXPECTED FEC
ERFPS: .WORD X ;ERROR FPS.
ERR1: ERROR 1 ;DATA ERROR.
      BR CONT
ERR2: ERROR 1 ;FPS ERROR.
CONT: ;RETURN ADDRESS
  
```

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

```

949                                     ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
950                                     ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
951                                     ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
952                                     ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
953
954 005460 012601 STCFDS: MOV (SP)+,R1 ;PICK UP THE POINTER TO THE OPERANDS.
955 005462 012700 000200 MOV #200,R0 ;ENTER DOUBLE FLOATING MODE.
956 005466 170100 LDFPS R0
957 005470 010100 MOV R1,R0 ;LOAD ACO.
958 005472 172410 LDD (R0),ACO
959 005474 012700 177777 MOV #-1,R0 ;FILL THE OUTPUT BUFFER WITH -1'S.
960 005500 012702 005742 MOV #STCFT,R2
961 005504 012703 000004 MOV #4,R3
962 005510 010022 1$: MOV R0,(R2)+
963 005512 077302 SOB R3,1$
964 005514 016100 000030 MOV 30(R1),R0 ;LOAD THE FPS.
965
966 005520 170100 LDFPS R0
967 005522 012737 005534 001236 MOV #2$,@#$TMP2
968 005530 012700 005742 MOV #STCFT,R0 ;SET UP THE DESTINATION ADDRESS.
969 005534 176010 2$: STCFD ACO,(R0) ;TEST INSTRUCTION.
970
971 005536 170204 STFPS R4 ;GET THE FPS.
972 005540 170305 STST R5 ;GET THE FEC.
973 005542 010102 MOV R1,R2 ;SAVE THE DATA IN CASE OF ERROR.
974 005544 010237 001240 MOV R2,@#$TMP3
975 005550 062702 000010 ADD #10,R2
976 005554 010237 001244 MOV R2,@#$TMP5
977 005560 012737 005742 001242 MOV #STCFT,@#$TMP4
978 005566 010437 001250 MOV R4,@#$TMP7
979 005572 016137 000032 001252 MOV 32(R1),@#$TMP10
980
981 005600 010102 MOV R1,R2 ;CHECK THE RESULT.
982 005602 062702 000010 ADD #10,R2
983 005606 012703 005742 MOV #STCFT,R3
984 005612 012700 000004 MOV #4,R0
985 005616 022223 3$: CMP (R2)+,(R3)+
986 005620 001014 BNE 15$ ;BRANCH IF INCORRECT.
987 005622 077003 SOB R0,3$
988
989 005624 016102 000032 MOV 32(R1),R2
990 005630 020204 CMP R2,R4 ;IS THE FPS CORRECT?
991 005632 001025 BNE 20$ ;BRANCH IF FPS INCORRECT.
992 005634 005702 TST R2 ;IF EXPECTED FPS IS NEGATIVE, THEN
993 005636 100003 BPL 4$ ;GO AHEAD AND CHECK THE FEC.
994 005640 026105 000036 CMP 36(R1),R5
995 005644 001027 BNE 25$ ;BRANCH IF FEC IS INCORRECT.
996 005646 000161 000046 4$: JMP 46(R1) ;RETURN.
997
998 ;RESULT INCORRECT:
999 005652 010102 15$: MOV R1,R2 ;SEE IF ERROR WAS ANTICIPATED.
1000 005654 062702 000020 ADD #20,R2
1001
1002 005660 012703 005742 MOV #STCFT,R3
1003 005664 012700 000004 MOV #4,R0
1004 005670 022223 16$: CMP (R2)+,(R3)+
    
```



```

1005 005672 001003          BNE      17$          ;BRANCH IF NOT ANTICIPATED.
1006 005674 077003          SOB      R0,16$
1007 005676 000161 000040  JMP      40(R1)       ;IF ERROR WAS ANTICIPATEI RETURN.
1008                                     ;OTHERWISE REPORT RESULT INCORRECT HERE.
1009 005702          17$:
1010 005702 104001          18$:  ERROR  1          ;DATA ERROR
1011 005704 000760          BR       4$
1012
1013                                     ;FPS INCORRECT:
1014 005706 020461 000034  20$:  CMP      R4,34(R1) ;WAS THE ERROR ANTICIPATED.
1015 005712 001002          BNE      21$          ;BRANCH IF NOT ANTICIPATED.
1016 005714 000161 000044  JMP      44(R1)       ;IF IT WAS ANTICIPATED RETURN.
1017
1018                                     ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
1019 005720          21$:
1020 005720 104001          22$:  ERROR  1          ;FPS X
1021 005722 000751          BR       4$
1022
1023                                     ;REPORT FEC INCORRECT:
1024 005724 016137 000036 001256 25$:  MOV      36(R1),@#$TMP12
1025 005732 010537 001254  MOV      R5,@#$TMP11
1026 005736 104001          26$:  ERROR  1          ;FEC X
1027 005740 000742          BR       4$
1028 005742 177777 177777 177777 STCFT: -1,-1,-1,-1
1029 005750 177777
1029 005752          XXXDONE:
(1) 005752 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                                     ;SEE IF THE USER HAS EXPRESSED
(1)                                     ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                                     ;THE USER TYPED CONTROL G?).
1030
1036                                     ;*****
(3)                                     ;*TEST 13          STCDF TEST
(4)                                     ;*
(4)                                     ;*THIS IS A TEST OF THE STCDF INSTRUCTION.
(4)                                     ;*
(3)                                     ;*****
(2) 005754 000004  TST13: SCOPE
1037
1038                                     ;AC=0
1039 005756          YYY1:
(1) 005756 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1040 005760 004767 000330  JSR      PC,STCDFS
1041 005764 000000          1$:  0          ;AC
1042 005766 000000          0
1043 005770 000000          0
1044 005772 000000          0
1045 005774 000000          2$:  0          ;RES
1046 005776 000000          0
1047 006000 177777          -1
1048 006002 177777          -1
1049 006004 000000          3$:  0          ;ERROR RES.
1050 006006 000000          0
1051 006010 000000          0
1052 006012 000000          0
    
```

1053	006014	047200		4\$:	47200			:FPS BEFORE EXECUTION.
1054	006016	047204			47204			:FPS AFTER EXECUTION.
1055	006020	177777			-1			:FEC
1056	006022	177777			-1			:ERROR FPS.
1057	006024	104001		5\$:	ERROR	1		:FDFL<---FDFL X ST767
1058	006026	000401			BR	6\$		
1059	006030	104001			ERROR	1		:FPS INCORRECT.
1060	006032			6\$:				
1061				:				
1062	006032			YYY2:				
(1)	006032	104414			LPERR			:SET UP THE LOOP ON ERROR ADDRESS.
1063	006034	004767	000254		JSR		PC,STCDF5	
1064	006040	067574		1\$:	67574			:ACO
1065	006042	073727			73727			
1066	006044	170777			170777			
1067	006046	067574			67574			
1068	006050	067574		2\$:	67574			:RES
1069	006052	073730			73730			
1070	006054	177777			-1			
1071	006056	177777			-1			
1072	006060	067574		3\$:	67574			:ERROR RES.
1073	006062	073727			73727			
1074	006064	177777			-1			
1075	006066	177777			-1			
1076	006070	040200		4\$:	40200			:FPS BEFORE EXECUTION.
1077	006072	040200			40200			:FPS AFTER EXECUTION.
1078	006074	177777			-1			:FEC
1079	006076	177777			-1			:ERROR FPS.
1080	006100	104001		5\$:	ERROR	1		:EITHER ROUND FAILED OR WENT TO 766 X1(1,0)<---0 INTO 76
1081	006102	000401			BR	6\$		
1082	006104	104001			ERROR	1		
1083	006106			6\$:				
1084				:				
1085	006106			YYY3:				
(1)	006106	104414			LPERR			:SET UP THE LOOP ON ERROR ADDRESS.
1086	006110	004767	000200		JSR		PC,STCDF5	
1087	006114	077777		1\$:	77777			:ACO
1088	006116	177777			-1			
1089	006120	100000			100000			
1090	006122	000000			0			
1091	006124	000000		2\$:	0			:RES
1092	006126	000000			0			
1093	006130	177777			-1			
1094	006132	177777			-1			
1095	006134	077777		3\$:	77777			:ERROR RES.
1096	006136	177777			-1			
1097	006140	177777			-1			
1098	006142	177777			-1			
1099	006144	040200		4\$:	40200			:FPS BEFORE EXECUTION.
1100	006146	040206			40206			:FPS AFTER EXECUTION.
1101	006150	177777			-1			:FEC
1102	006152	040204			40204			:ERROR FPS.
1103	006154	104001		5\$:	ERROR	1		
1104	006156	000401			BR	6\$		
1105	006160	104001			ERROR	1		:BUT EZBT X ST421 TO 062 INTO 262
1106	006162			6\$:				


```
1107  
1108 006162  
  (1) 006162 104414  
1109 006164 004767 000124  
1110 006170 077777  
1111 006172 177777  
1112 006174 100000  
1113 006176 000000  
1114 006200 000000  
1115 006202 000000  
1116 006204 177777  
1117 006206 177777  
1118 006210 077777  
1119 006212 177777  
1120 006214 177777  
1121 006216 177777  
1122 006220 040200  
1123 006222 040206  
1124 006224 177777  
1125 006226 140206  
1126 006230 104001  
1127 006232 000401  
1128 006234 104001  
1129 006236  
1130  
1131 006236  
  (1) 006236 104414  
1132 006240 004767 000050  
1133 006244 177777  
1134 006246 177777  
1135 006250 100000  
1136 006252 000000  
1137 006254 100000  
1138 006256 000000  
1139 006260 177777  
1140 006262 177777  
1141 006264 000000  
1142 006266 000000  
1143 006270 177777  
1144 006272 177777  
1145 006274 047200  
1146 006276 147216  
1147 006300 000010  
1148 006302 047206  
1149 006304 104001  
1150 006306 000401  
1151 006310 104001  
1152 006312 000535  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160
```

Y4:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,STCDF5
1\$: 77777 ;ACO
-1
100000
0
2\$: 0 ;RES
0
-1
-1
3\$: 77777 ;ERROR RES.
-1
-1
-1
4\$: 40200 ;FPS BEFORE EXECUTION.
40206 ;FPS AFTER EXECUTION.
-1 ;FEC
140206 ;ERROR FPS.
5\$: ERROR 1
BR 6\$
ERROR 1 ;BUT FIV ST262 TO 123 INTO 103
6\$:
Y5:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,STCDF5
1\$: 177777 ;ACO
-1
100000
0
2\$: 100000 ;RES
0
-1
-1
3\$: 0 ;ERROR RES.
0
-1
-1
4\$: 47200 ;FPS BEFORE EXECUTION.
147216 ;FPS AFTER EXECUTION.
10 ;FEC
47206 ;ERROR FPS.
5\$: ERROR 1 ;BUT FIV ST262 FAIL TO 103 INT 123
BR 6\$
ERROR 1 ;BUT FLAG ST 147 X TO ST 361 INTO 365
6\$: BR YYDONE
;THIS SUBROUTINE, STCDF5, IS USED TO SET UP THE OPERANDS, EXECUTE
;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
;TO IT IS MADE THUS:
: :
: JSR PC,@#STCDF5
: ACARG: .WORD X,X,X,X ;AC OPERAND
: RES: .WORD X,X,X,X ;EXPECTED RESULT
: ERRES: .WORD X,X,X,X ;ERROR RESULT

```

1161      :
1162      :
1163      :
1164      :
1165      :
1166      :
1167      :
1168      :
1169      :
1170      :
1171      :
1172      :
1173      :
1174      :
1175      :
1176      :
1177      :
1178      :
1179      :
1180      :
1181      :
1182      :
1183      :
1184      :
1185      006314 012601          STCDFS: MOV      (SP)+,R1      ;PICK UP THE POINTER TO THE OPERANDS.
1186      006316 012700 000200      MOV      #200,R0      ;ENTER DOUBLE FLOATING MODE.
1187      006322 170100          LDFPS   R0
1188      006324 010100          MOV      R1,R0      ;LOAD ACO.
1189      006326 172410          LDD     (R0),ACO
1190      006330 012700 177777      MOV      #-1,R0      ;FILL THE OUTPUT BUFFER WITH -1'S.
1191      006334 012702 006576      MOV      #STCDT,R2
1192      006340 012703 000004      MOV      #4,R3
1193      006344 010022          1$:    MOV      R0,(R2)+
1194      006346 077302          SOB     R3,1$
1195      006350 016100 000030      MOV      30(R1),R0      ;LOAD THE FPS.
1196      006354 170100          LDFPS   R0
1197      006356 012737 006370 001236      MOV      #2$,@#STMP2
1198      006364 012700 006576          MOV      #STCDT,R0      ;SET UP THE DESTINATION ADDRESS.
1199      006370 176010          2$:    STCDF   ACO,(R0)      ;TEST INSTRUCTION.
1200
1201      006372 170204          STFPS   R4      ;GET THE FPS.
1202      006374 170305          STST    R5      ;GET THE FEC.
1203      006376 010102          MOV      R1,R2      ;SAVE THE DATA IN CASE OF ERROR.
1204      006400 010237 001240      MOV      R2,@#STMP3
1205      006404 062702 000010      ADD     #10,R2
1206      006410 010237 001244      MOV      R2,@#STMP5
1207      006414 012737 006576 001242      MOV      #STCDT,@#STMP4
1208      006422 010437 001250      MOV      R4,@#STMP7
1209      006426 016137 000032 001252      MOV      32(R1),@#STMP10
1210
1211      006434 010102          MOV      R1,R2      ;CHECK THE RESULT.
1212      006436 062702 000010      ADD     #10,R2
1213      006442 012703 006576      MOV      #STCDT,R3
1214      006446 012700 000004      MOV      #4,R0
1215      006452 022223          3$:    CMP     (R2)+,(R3)+
1216      006454 001014          BNE     15$      ;BRANCH IF INCORRECT.
    
```

:FPS BEFORE EXECUTION
 :FPS AFTER EXECUTION
 :EXPECTED FEC
 :ERROR FPS.
 :DATA ERROR.
 :FPS ERROR.
 :RETURN ADDRESS

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


```
1217 006456 077003 SOB R0,3$
1218
1219 006460 016102 000032 MOV 32(R1),R2
1220 006464 020204 CMP R2,R4 ;IS THE FPS CORRECT?
1221 006466 001025 BNE 20$ ;BRANCH IF FPS INCORRECT.
1222 006470 005702 TST R2 ;IF EXPECTED FPS IS NEGATIVE, THEN
1223 006472 100003 BPL 4$ ;GO AHEAD AND CHECK THE FEC.
1224 006474 026105 000034 CMP 34(R1),R5
1225 006500 001027 BNE 25$ ;BRANCH IF FEC IS INCORRECT.
1226 006502 000161 000046 4$: JMP 46(R1) ;RETURN.
1227
1228 ;RESULT INCORRECT:
1229 006506 010102 15$: MOV R1,R2 ;SEE IF ERROR WAS ANTICIPATED.
1230 006510 062702 000020 ADD #20,R2
1231 006514 012703 006576 MOV #STCDT,R3
1232 006520 012700 000004 MOV #4,R0
1233 006524 022223 16$: CMP (R2)+,(R3)+ ;BRANCH IF NOT ANTICIPATED.
1234 006526 001003 BNE 17$
1235 006530 077003 SOB R0,16$
1236 006532 000161 000040 JMP 40(R1) ;IF ERROR WAS ANTICIPATED RETURN.
1237 ;OTHERWISE REPORT RESULT INCORRECT HERE.
1238 006536
1239 006536 104001 17$: ERROR 1 ;DATA ERROR
1240 006540 000760 BR 4$
1241
1242 ;FPS INCORRECT:
1243 006542 020461 000034 20$: CMP R4,34(R1) ;WAS THE ERROR ANTICIPATED.
1244 006546 001002 BNE 21$ ;BRANCH IF NOT ANTICIPATED.
1245 006550 000161 000044 JMP 44(R1) ;IF IT WAS ANTICIPATED RETURN.
1246
1247 ;THE FPS ERROR WAS NOT ANTICIPATED SO REPORT FPS INCORRECT HERE.
1248 006554 21$:
1249 006554 104001 22$: ERROR 1 ;FPS X
1250 006556 000751 BR 4$
1251
1252 ;REPORT FEC INCORRECT:
1253 006560 016137 000036 001256 25$: MOV 36(R1),@#STMP12
1254 006566 010537 001254 MOV R5,@#STMP11
1255 006572 104001 26$: ERROR 1 ;FEC X
1256 006574 000742 BR 4$
1257 006576 177777 177777 177777 STCDT: -1,-1,-1,-1
006604 177777
1258 006606 YYYDONE:
(1) 006606 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1264 ;*****
(3) ;*TEST 14 STCFD WITH ILLEGAL ACCUMULATOR TEST
(4) ;*
(4) ;*THIS TEST STCFD WITH ILLEGAL AC 6.
(4) ;*
(3) ;*****
(2) 006610 000004 TST14: SCOPE
1265
```

```
1266 006612          ZZZ1:          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 006612 104414          MOV          #40000,R0          ;DISSABLE INTERRUPTS.
1267 006614 012700 040000          LDFPS        R0
1268 006620 170100          MOV          #ZZZ2,@#$TMP2
1269 006622 012737 006630 001236 ZZZ2: STCFD     AC0,AC6          ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.
1270 006630 176006
1271
1272 006632 170204          STFPS        R4          ;GET FPS.
1273 006634 170305          STST         R5          ;GET FEC.
1274 006636 020427 140000          CMP          R4,#140000      ;IS FPS CORRECT?
1275 006642 001004          BNE          ZZZ10          ;BRANCH IF INCORRECT FPS.
1276 006644 022705 000002          CMP          #2,R5          ;IS FEC CORRECT?
1277
1278 006650 001010          BNE          ZZZ15          ;BRANCH IF INCORRECT.
1279 006652 000415          BR           ZZZDONE
1280
1281          ;REPORT FPS INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
1282 006654 010437 001242          ZZZ10: MOV        R4,@#$TMP4
1283 006660 012737 140000 001240          MOV        #140000,@#$TMP3
1284 006666 104001          1$: ERROR    1          ;BUT FDST ST767 X TO 567 INTO 577
1285 006670 000406          BR           ZZZDONE
1286
1287          ;REPORT FEC INCORRECT AFTER USE OF ILLEGAL ACCUMULATOR.
1288 006672 010537 001242          ZZZ15: MOV        R5,@#$TMP4
1289 006676 012737 000002 001240          MOV        #2,@#$TMP3
1290 006704 104001          1$: ERROR    1          ;FEC<---2 ST577 X
1291 006706          ZZZDONE:
(1) 006706 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)          ;SEE IF THE USER HAS EXPRESSED
(1)          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)          ;THE USER TYPED CONTROL G?).
1292
1298          ;*****
(3)          ;*TEST 15          CLRD TEST
(4)          ;*
(4)          ;*THIS IS A TEST OF THE CRLF AND CLRD INSTRUCTIONS.
(4)          ;*
(3)          ;*****
(2) 006710 000004          TST15: SCOPE
1299 006712          AAB1:
(1) 006712 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1300 006714 012700 007100          MOV          #AABTP1,R0      ;SET UP OUTPUT BUFFER
1301 006720 012701 007070          MOV          #AABBF0,R1
1302 006724 012702 000004          MOV          #4,R2
1303 006730 012021          1$: MOV        (R0)+,(R1)+
1304 006732 077202          SOB         R2,1$
1305 006734 012700 007070          MOV          #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS.
1306 006740 012701 000213          MOV          #213,R1        ;SET UP FPS.
1307 006744 170101          LDFPS        R1
1308 006746 012737 006754 001236          MOV          #2$,@#$TMP2
1309 006754 170410          2$: CLRD     (R0)          ;TEST INSTRUCTION.
1310
1311 006756 170205          STFPS        R5          ;GET FPS.
1312 006760 012702 000004          MOV          #4,R2          ;SEE IF RESULT CLEAR, 0.
1313 006764 012701 007070          MOV          #AABBF0,R1
```



```
1314 006770 005721 3$: TST (R1)+
1315 006772 001010 BNE AAB2 ;BRANCH IF RESULT INCORRECT, NOT 0.
1316 006774 077203 SOB R2,3$
1317 006776 022705 000204 CMP #204,R5 ;SEE IF FPS IS CORRECT.
1318 007002 001014 BNE AAB3 ;BRANCH IF INCORRECT.
1319 007004 020027 007070 CMP R0,#AABBF0 ;SEE IF R0 IS CORRECT.
1320 007010 001020 BNE AAB4 ;BRANCH IF R0 IS INCORRECT.
1321 007012 000442 BR AABDONE
1322
1323 ;RESULT NOT 0, REPORT ERROR.
1324 007014 012737 007070 001240 AAB2: MOV #AABBF0,@#$TMP3
1325 007022 012737 007110 001242 MOV #AABTP2,@#$TMP4
1326 007030 104001 1$: ERROR 1 ;BAD DATA = 0 X 11+ZERO ST770 X
1327 007032 000432 BR AABDONE
1328
1329 ;REPORT FPS INCORRECT:
1330 007034 010437 001242 AAB3: MOV R4,@#$TMP4
1331 007040 012737 000204 001240 MOV #204,@#$TMP3
1332 007046 104001 1$: ERROR 1 ;BAD FPS
1333 007050 000423 BR AABDONE
1334
1335 ;REPORT R0 INCORRECT.
1336 007052 010037 001242 AAB4: MOV R0,@#$TMP4
1337 007056 012737 007070 001240 MOV #AABBF0,@#$TMP3
1338 007064 104001 1$: ERROR 1
1339 007066 000414 BR AABDONE
1340
1341 ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
1342 007070 073475 AABBF0: 73475
1343 007072 067707 67707
1344 007074 127347 127347
1345 007076 056770 56770
1346 ;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
1347 007100 073475 AABTP1: 73475
1348 007102 067707 67707
1349 007104 127347 127347
1350 007106 056770 56770
1351 ;THIS IS THE EXPECTED DATA, RESULT:
1352 007110 000000 AABTP2: 0
1353 007112 000000 0
1354 007114 000000 0
1355 007116 000000 0
1356 007120 AABDONE:
(1) 007120 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1357
1363 ;*****
(3) ;*TEST 16 CLR D WITH ILLEGAL ACCUMULATOR TEST
(4) ;*
(4) ;*THIS IS A TEST OF CLR D WITH ILLEGAL AC7.
(4) ;*
(3) ;*****
(2) 007122 000004 TST16: SCOPE
```

```
1364 007124 CCB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 007124 104414 MOV #40200,R0 ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
1365 007126 012700 040200 LDFPS R0
1366 007132 170100 MOV #CCB2,@#$TMP2
1367 007134 012737 007142 001236 CCB2: CLRD AC7 ;TEST INSTRUCTION.
1368 007142 170407
1369
1370 007144 170204 STFPS R4 ;GET FPS.
1371 007146 170305 STST R5 ;GET FEC.
1372 007150 020427 140200 CMP R4,#140200 ;IS THE FPS CORRECT?
1373 007154 001004 BNE CCB10 ;BRANCH IF FPS IS INCORRECT.
1374 007156 022705 000002 CMP #2,R5 ;IS THE FEC CORRECT?
1375 007162 001010 BNE CCB15 ;BRANCH IF FEC IS INCORRECT.
1376 007164 000415 BR CCBDONE
1377
1378 ;REPORT INCORRECT FPS:
1379 007166 010437 001242 CCB10: MOV R4,@#$TMP4
1380
1381 007172 012737 140200 001240 1$: MOV #140200,@#$TMP3
1382 007200 104001 ERROR 1 ;BUT FDST ST 700X TO 607 INTO 677
1383 007202 000406 BR CCBDONE
1384
1385 ;REPORT INCORRECT FEC:
1386 007204 010537 001242 CCB15: MOV R5,@#$TMP4
1387 007210 012737 000002 001240 MOV #2,@#$TMP3
1388 007216 104001 1$: ERROR 1 ;FEC---2 ST 677 X
1389 007220 CCBDONE:
(1) 007220 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1398
1399 ;*****
(3) ;*TEST 17 NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
(4) ;*
(4) ;*THIS IS A TEST OF THE SPECIAL
(4) ;*DEST FLOWS USING THE NEGD INST
(4) ;*WITH MODE ZERO AND ILLEGAL
(4) ;*AC7.
(4) ;*
(3) ;*****
(2) 007222 000004 TST17: SCOPE
1400
1401 007224 VVB1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 007224 104414 MOV #40200,R0 ;SET UP THE FPS, FID=1 AND FD=1.
1402 007226 012700 040200 LDFPS R0
1403 007232 170100 MOV #VVB2,@#$TMP2
1404 007234 012737 007242 001236 VVB2: NEGD AC7 ;TEST INSTRUCTION.
1405
1406 007242 170707
1407
1408 007244 170204 STFPS R4 ;GET FPS.
1409 007246 170305 STST R5 ;GET FEC.
1410
1411 007250 022704 140200 CMP #140200,R4 ;IS FPS CORRECT?
```



```
1412 007254 001004          BNE      VVB10          ;BRANCH IF FPS IS INCORRECT.
1413 007256 022705 000002    CMP      #2,R5          ;IS FEC CORRECT?
1414 007262 001010          BNE      VVB15          ;BRANCH IF FEC IS INCORRECT.
1415 007264 000415          BR       VVBDONE
1416
1417                          ;REPORT INCORRECT FPS:
1418 007266 012737 140200 001240 VVB10:  MOV      #140200,@#$TMP3
1419 007274 010437 001242    MOV      R4,@#$TMP4
1420 007300 104002          1$:     ERROR      2          ;FPS BAD
1421 007302 000406          BR       VVBDONE
1422
1423                          ;REPORT FEC INCORRECT:
1424 007304 012737 000002 001240 VVB15:  MOV      #2,@#$TMP3
1425 007312 010537 001242    MOV      R5,@#$TMP4
1426 007316 104002          1$:     ERROR      2          ;FEC BAD
1427
1428 007320          VVBDONE:
(1) 007320 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                          ;SEE IF THE USER HAS EXPRESSED
(1)                          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                          ;THE USER TYPED CONTROL G?).
1429
1437                          ;*****
(3)                          ;*TEST 20      NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
(4)                          ;*
(4)                          ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
(4)                          ;*SOURCE FLOWS. THE NEGD INSTRUCTION
(4)                          ;*IS USED TO TEST MODE 0
(4)                          ;*
(3)                          ;*****
(2) 007322 000004          TST20:  SCOPE
1438
1439 007324          DDB1:
(1) 007324 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
1440 007326 012700 000200    MOV      #200,R0        ;SET FD MODE.
1441 007332 170100          LDFPS      R0
1442 007334 012700 007476    MOV      #DDBTP1,R0     ;SET UP ACO.
1443 007340 172410          LDD      (R0),AC0       ;SET ACO = 0
1444 007342 005000          CLR      R0             ;CLEAR THE FPS.
1445 007344 170100          LDFPS      R0
1446 007346 012700 007506    MOV      #DDBTP2,R0     ;LOAD ACO TO BE A FLOATING 0.
1447 007352 172410          LDF      (R0),AC0       ;SET ACO=ZERO
1448                          ;FLOAT
1449 007354 012700 000201    MOV      #201,R0        ;SET FD MODE.
1450 007360 170100          LDFPS      R0
1451 007362 012737 007370 001236 MOV      #DDB2,@#$TMP2
1452
1453 007370 170700          DDB2:  NEGD      AC0          ;TEST INSTRUCTION.
1454
1455 007372 170205          STFPS      R5            ;GET FPS.
1456 007374 012700 000200    MOV      #200,R0        ;SET FD MODE.
1457 007400 170100          LDFPS      R0
1458 007402 012700 007516    MOV      #DDBBF0,R0     ;GET THE RESULT OUT OF ACO.
1459 007406 174010          STD      ACO,(R0)       ;SEE IF THE RESULT IS CORRECT.
1460
```

```
1461 007410 012701 000004      MOV    #4,R1
1462 007414 005720      1$:   TST    (R0)+
1463 007416 001005      BNE    DDB5          ;BRANCH IF THE RESULT IS INCORRECT.
1464 007420 077103      SOB    R1,1$
1465 007422 022705 000204      CMP    #204,R5      ;IS THE FPS CORRECT?
1466 007426 001014      BNE    DDB6          ;BRANCH IF THE FPS IS INCORRECT.
1467 007430 000442      BR     DDBDONE
1468
1469      ;RESULT INCORRECT, REPORT FAILURE:
1470 007432 012737 007506 001242 DDB5:  MOV    #DDBTP2,@#$TMP4 ;EXPECT DO
1471 007440 012737 007526 001240      MOV    #DDBTP3,@#$TMP3 ;PREV FO IMPURE
1472 007446 012737 007516 001244      MOV    #DDBBF0,@#$TMP5 ;GOT
1473 007454 104002      1$:   ERROR  2
1474 007456 000427      BR     DDBDONE
1475
1476      ;REPORT FPS INCORRECT:
1477 007460 012737 000204 001240 DDB6:  MOV    #204,@#$TMP3
1478 007466 010537 001242      MOV    R5,@#$TMP4
1479 007472 104002      1$:   ERROR  2
1480 007474 000420      BR     DDBDONE
1481
1482      ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
1483 007476 101112 DDBTP1: 101112
1484 007500 131415      131415
1485 007502 161710      161710
1486 007504 111213      111213
1487 007506 000000 DDBTP2: 0
1488 007510 000000      0
1489 007512 000000      0
1490 007514 000000      0
1491
1492 007516 177777 DDBBF0: -1
1493 007520 177777      -1
1494 007522 177777      -1
1495 007524 177777      -1
1496 007526 000000 DDBTP3: 0
1497 007530 000000      0
1498 007532 161710      161710
1499 007534 111213      111213
1500
1501 007536      DDBDONE:
(1) 007536 104413      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                  ;SEE IF THE USER HAS EXPRESSED
(1)                  ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                  ;THE USER TYPED CONTROL G?).
1502
1503      ;*****
(3)      ;*TEST 21      NEG, ABSF AND TSTF SOURCE MODE 1 TEST
(4)      ;*
(4)      ;*THIS IS A TEST THE NEG, ABSF AND TSTF
(4)      ;*SOURCE FLOWS. THE NEG, INSTRUCTION
(4)      ;*IS USED TO TEST MODE 1
(4)      ;*
(3)      ;*****
(2) 007540 000004 TST21: SCOPE
```



```

1504
1505 007542          EEB1: LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
      (1) 007542 104414      MOV #EEBTP1,R0 ;SET UP THE DATA BUFFER.
1506 007544 012700 007652      MOV #EEBF1,R1
1507 007550 012701 007702      MOV #4,R2
1508 007554 012702 000004      1$: MOV (R0)+,(R1)+
1509 007560 012021      SOB R2,1$
1510 007562 077202      MOV #200,R0 ;SET FD MODE.
1511 007564 012700 000200      LDFPS R0
1512 007570 170100      MOV #EEBF1,R0 ;SET UP THE OPERAND ADDRESS.
1513 007572 012700 007702      MOV #EEB2,@#$TMP2
1514 007576 012737 007612 001236      MOV #EEB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR.
1515 007604 012737 007712 000004      EEB2: NEG D (R0) ;TEST INSTRUCTION.
1516 007612 170710
1517
1518 007614 170205      STFPS R5 ;GET FPS.
1519 007616 012701 007702      MOV #EEBF1,R1 ;SEE IF RESULT IS CORRECT.
1520 007622 012702 000004      MOV #4,R2
1521 007626 005721      1$: TST (R1)+
1522 007630 001046      BNE EEB15 ;BRANCH IF NOT CORRECT.
1523 007632 077203      SOB R2,1$
1524
1525 007634 020027 007702      CMP R0,#EEBF1 ;IS R0 CORRECT?
1526 007640 001055      BNE EEB20 ;BRANCH IF NOT CORRECT.
1527 007642 022705 000204      CMP #204,R5 ;IS THE FPS CORRECT?
1528 007646 001061      BNE EEB25 ;BRANCH IF NOT CORRECT.
1529 007650 000466      BR EEBDONE
1530
1531 ;THESE ARE TEST DATA TABLES AND A BUFFER.
1532 007652 000177      EEBTP1: 177
1533 007654 167574      167574
1534 007656 137271      137271
1535 007660 107675      107675
1536 007662 000000      EEBTP2: 0
1537 007664 000000      0
1538 007666 000000      0
1539 007670 000000      0
1540 007672 177777      EEBBF0: -1
1541 007674 177777      -1
1542 007676 177777      -1
1543 007700 177777      -1
1544 007702 177777      EEBBF1: -1
1545 007704 177777      -1
1546 007706 177777      -1
1547 007710 177777      -1
1548
1549 ;IF A TRAP TO 4 OCCURS COME HERE:
1550 007712 011602      EEB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTR.
1551 007714 020227 007614      CMP R2,#EEB2+2
1552 007720 001405      BEQ 1$ ;BRANCH IF YES.
1553 007722 020227 007616      CMP R2,#EEB2+4
1554 007726 001402      BEQ 1$ ;BRANCH IF YES.
1555 007730 000137 036166      JMP @#CPSPUR ;OTHERWISE GO REPORT A SPURIOUS TRAP TO 4.
1556 ;REPORT A FAILURE IN THE FDST FLOWS RESULTED IN AN ODD ADDRESS TRAP TO 4.
1557 007734 022626      1$: CMP (SP)+,(SP)+ ;RESET THE STACK.
1558 007736 010237 001236      MOV R2,@#$TMP2
  
```

```

1559 007742 104002      2$:  ERROR 2      ;ODD ADRES
1560 007744 000430      BR      EEBDONE ;BUT FDSTX IN ST 771
1561
1562      ;REPORT RESULT INCORRECT.
1563 007746 012737 007662 001242 EEB15: MOV #EEBTP2,@#STMP4
1564 007754 012737 007652 001240      MOV #EEBTP1,@#STMP3
1565 007762 012737 007702 001244      MOV #EEBBF1,@#STMP5
1566 007770 104002      1$:  ERROR 2      ;BAD DATA X11*0 ST 312X
1567 007772 000415      BR      EEBDONE
1568
1569      ;RO INCORRECT:
1570 007774 012737 007702 001240 EEB20: MOV #EEBBF1,@#STMP3
1571 010002 010037 001242      MOV RO,@#STMP4
1572 010006 104002      1$:  ERROR 2      ;RO BADX
1573 010010 000406      BR      EEBDONE
1574
1575      ;REPORT FPS INCORRECT:
1576 010012 010537 001240 EEB25: MOV R5,@#STMP3
1577 010016 012737 000204 001244      MOV #204,@#STMP5
1578 010024 104002      1$:  ERROR 2      ;FPS X
1579
1580      EEBDONE:
(1) 010026 104413      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
(1)      ;SEE IF THE USER HAS EXPRESSED
(1)      ;THE DESIRE TO CHANGE THE SOFTWARE
(1)      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)      ;THE USER TYPED CONTROL G?).
1581
1582      ;*****
(3)      ;*TEST 22      NEG, ABSF AND TSTF SOURCE MODE 2 TEST
(4)      ;*
(4)      ;*THIS IS A TEST THE NEG, ABSF AND TSTF
(4)      ;*SOURCE FLOWS. THE ABSD INSTRUCTION
(4)      ;*IS USED TO TEST MODE 2
(4)      ;*
(3)      ;*****
(2) 010030 000004      TST22: SCOPE
1583
1584 010032      FFB1:
(1) 010032 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
1585 010034 012700 010142      MOV #FFBTP1,R0      ;SET UP THE DATA BUFFER.
1586 010040 012701 010172      MOV #FFBBF1,R1
1587 010044 012702 000004      MOV #4,R2
1588 010050 012021      1$:  MOV (R0)+,(R1)+
1589 010052 077202      SOB R2,1$
1590 010054 012700 000200      MOV #200,R0      ;SET FD.
1591 010060 170100      LDFPS RO
1592 010062 012700 010172      MOV #FFBBF1,R0      ;SET UP THE OPERAND ADDRESS.
1593 010066 012737 010102 001236      MOV #FFB2,@#STMP2
1594 010074 012737 010202 000004      MOV #FFB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
1595
1596 010102 170620      FFB2:  ABSD (R0)+      ;TEST INSTRUCTION.
1597
1598 010104 170205      STFPS R5      ;GET FPS.
1599 010106 012701 010172      MOV #FFBBF1,R1      ;CHECK RESULT.
1600 010112 012702 000004      MOV #4,R2
    
```



```
1601 010116 005721          1$: TST (R1)+
1602 010120 001046          BNE FFB15          ;BRANCH IF INCORRECT.
1603 010122 077203          SOB R2,1$
1604
1605 010124 020027 010202    CMP R0,#FFBBF1+10 ;IS R0 CORRECT?
1606 010130 001055          BNE FFB20          ;BRANCH IF INCORRECT.
1607 010132 022705 000204    CMP #204,R5        ;IS THE FPS CORRECT?
1608 010136 001061          BNE FFB25          ;BRANCH IF INCORRECT.
1609 010140 000466          BR FFBDONE
1610
1611          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
1612 010142 000177          FFBTP1: 177
1613 010144 167574          167574
1614 010146 137271          137271
1615 010150 107675          107675
1616 010152 000000          FFBTP2: 0
1617 010154 000000          0
1618 010156 000000          0
1619 010160 000000          0
1620 010162 177777          FFBBF0: -1
1621 010164 177777          -1
1622 010166 177777          -1
1623 010170 177777          -1
1624 010172 177777          FFBBF1: -1
1625 010174 177777          -1
1626 010176 177777          -1
1627 010200 177777          -1
1628
1629          ;IF A TRAP TO 4 OCCURS COME HERE.
1630 010202 011602          FFB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
1631 010204 020227 010104    CMP R2,#FFB2+2
1632 010210 001405          BEQ 1$ ;BRANCH IF YES.
1633 010212 020227 010106    CMP R2,#FFB2+4
1634 010216 001402          BEQ 1$ ;BRANCH IF YES.
1635 010220 000137 036166    JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1636          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
1637 010224 022626          1$: CMP (SP)+,(SP)+
1638 010226 010237 001236    MOV R2,@#STMP2
1639 010232 104002          2$: ERROR 2 ;ODD ADRES
1640 010234 000430          BR FFBDONE ;BUT FDSTX IN ST 771
1641
1642          ;REPORT RESULT INCORRECT:
1643 010236 012737 010152 001240 FFB15: MOV #FFBTP2,@#STMP3
1644 010244 012737 010142 001242    MOV #FFBTP1,@#STMP4
1645 010252 012737 010172 001244    MOV #FFBBF1,@#STMP5
1646 010260 104002          1$: ERROR 2 ;BAD DATA X11*0 ST 312X
1647 010262 000415          BR FFBDONE
1648
1649          ;REPORT R0 INCORRECT:
1650 010264 012737 010176 001240 FFB20: MOV #FFBBF1+4,@#STMP3
1651 010272 010037 001242    MOV R0,@#STMP4
1652 010276 104002          1$: ERROR 2 ;R0 BADX
1653 010300 000406          BR FFBDONE
1654
1655          ;REPORT FPS INCORRECT:
1656 010302 010537 001240    FFB25: MOV R5,@#STMP3
```

```

1657 010306 012737 000204 001244 MOV #204,@#$TMP5
1658 010314 104002 1$: ERROR 2 ;FPS X
1659
1660 010316 FFBDONE:
(1) 010316 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1661
(3) ;*****
(4) ;*TEST 23 NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
(4) ;*
(4) ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
(4) ;*SOURCE FLOWS. THE ABSD INSTRUCTION
(4) ;*IS USED TO TEST MODE 4
(4) ;*
(3) ;*****
(2) 010320 000004 TST23: SCOPE
1662
1663 010322 GGB1:
(1) 010322 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1664 010324 012700 010432 MOV #GGBTP1,R0 ;SET UP THE DATA BUFFER.
1665 010330 012701 010452 MOV #GGBBF0,R1
1666 010334 012702 000004 MOV #4,R2
1667 010340 012021 1$: MOV (R0)+,(R1)+
1668 010342 077202 SOB R2,1$
1669 010344 012700 000200 MOV #200,R0 ;SET FD.
1670 010350 170100 LDFPS R0
1671 010352 012700 010462 MOV #GGBBF1,R0 ;SET UP THE OPERAND ADDRESS.
1672 010356 012737 010372 001236 MOV #GGB2,@#$TMP2
1673 010364 012737 010472 000004 MOV #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
1674
1675 010372 170640 GGB2: ABSD -(R0) ;TEST INSTRUCTION.
1676
1677 010374 170205 STFPS R5 ;GET FPS.
1678 010376 012701 010452 MOV #GGBBF0,R1 ;CHECK RESULT.
1679 010402 012702 000004 MOV #4,R2
1680 010406 005721 1$: TST (R1)+
1681 010410 001046 BNE GGB15 ;BRANCH IF INCORRECT.
1682 010412 077203 SOB R2,1$
1683
1684 010414 020027 010452 CMP R0,#GGBBF0 ;IS R0 CORRECT?
1685 010420 001055 BNE GGB20 ;BRANCH IF INCORRECT.
1686 010422 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
1687 010426 001061 BNE GGB25 ;BRANCH IF INCORRECT.
1688 010430 000466 BR GGBDONE
1689
1690 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
1691 010432 000177 GGBTP1: 177
1692 010434 117273 117273
1693 010436 147576 147576
1694 010440 177071 177071
1695 010442 000000 GGBTP2: 0
1696 010444 000000 0
1697 010446 000000 0
1698 010450 000000 0
    
```


1699 010452 177777
1700 010454 177777
1701 010456 177777
1702 010460 177777
1703 010462 177777
1704 010464 177777
1705 010466 177777
1706 010470 177777

GGBBF0: -1
-1
-1
-1
GGBBF1: -1
-1
-1
-1

1707
1708
1709 010472 011602
1710 010474 020227 010374
1711 010500 001405
1712 010502 020227 010376
1713 010506 001402
1714 010510 000137 036166
1715
1716 010514 022626
1717 010516 010237 001236
1718 010522 104002
1719 010524 000430

:IF A TRAP TO 4 OCCURS COME HERE.
GGB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2,#GGB2+2
BEQ 1\$;BRANCH IF YES.
CMP R2,#GGB2+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 2 ;ODD ADRES
BR GGBDONE ;BUT FDSTX IN ST 771

1720
1721
1722 010526 012737 010442 001240
1723 010534 012737 010432 001242
1724 010542 012737 010452 001244
1725 010550 104002
1726 010552 000415

:REPORT RESULT INCORRECT:
GGB15: MOV #GGBTP2,@#STMP3
MOV #GGBTP1,@#STMP4
MOV #GGBBF0,@#STMP5
1\$: ERROR 2 ;BAD DATA X11*0 ST 312X
BR GGBDONE

1727
1728
1729 010554 012737 010452 001240
1730 010562 010037 001242
1731 010566 104002
1732 010570 000406

:REPORT R0 INCORRECT:
GGB20: MOV #GGBBF01,@#STMP3
MOV R0,@#STMP4
1\$: ERROR 2 ;R0 BADX
BR GGBDONE

1733
1734
1735 010572 010537 001240
1736 010576 012737 000204 001244
1737 010604 104002

:REPORT FPS INCORRECT:
GGB25: MOV R5,@#STMP3
MOV #204,@#STMP5
1\$: ERROR 2 ;FPS X

1738
1739 010606
(1) 010606 104413
(1)
(1)
(1)
(1)

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

1740
(3)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 010610 000004
1741

: *TEST 24 NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
: *
: *THIS IS A TEST THE NEGF, ABSF AND TSTF
: *SOURCE FLOWS. THE ABSD INSTRUCTION
: *IS USED TO TEST MODE 3
: *
: *****
TST24: SCOPE

```
1742 010612          HHB1: LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 010612 104414    MOV          #HHBTP1,R0      ;SET UP THE DATA BUFFER.
1743 010614 012700 010722    MOV          #HHBBF0,R1
1744 010620 012701 010752    MOV          #10,R2
1745 010624 012702 000010    1$: MOV        (R0)+,(R1)+
1746 010630 012021    SOB          R2,1$
1747 010632 077202    MOV          #200,R0      ;SET FD.
1748 010634 012700 000200    LDFPS       R0
1749 010640 170100    MOV          #HHBBF1,R0   ;SET UP THE OPERAND ADDRESS.
1750 010642 012700 010762    MOV          #HHB2,@#$TMP2
1751 010646 012737 010662 001236    MOV          #HHB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
1752 010654 012737 010772 000004
1753
1754 010662 170630    HHB2: ABSD     @(R0)+      ;TEST INSTRUCTION.
1755
1756 010664 170205    STFPS       R5           ;GET FPS.
1757 010666 012701 010752    MOV          #HHBBF0,R1   ;CHECK RESULT.
1758 010672 012702 000004    MOV          #4,R2
1759 010676 005721    1$: TST        (R1)+
1760 010700 001052    BNE         HHB15       ;BRANCH IF INCORRECT.
1761 010702 077203    SOB          R2,1$
1762 010704 020027 010764    CMP         R0,#HHBBF1+2 ;IS R0 CORRECT?
1763 010710 001061    BNE         HHB20       ;BRANCH IF INCORRECT.
1764 010712 022705 000204    CMP         #204,R5     ;IS THE FPS CORRECT?
1765 010716 001065    BNE         HHB25       ;BRANCH IF INCORRECT.
1766 010720 000472    BR          HHBDONE
1767
1768 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
1769 010722 000177    HHBTP1: 177
1770 010724 147576    147576
1771 010726 177071    177071
1772 010730 107576 010752 177777    107576,HHBBF0,-1,-1,-1
1773 010736 177777 177777
1774 010742 000000 000000 000000 HHBTP2: 0,0,0,0
1775 010750 000000
1776 010752 177777    HHBBF0: -1
1777 010754 177777    -1
1778 010756 177777    -1
1779 010760 177777    -1
1780 010762 177777    HHB1: -1
1781 010764 177777    -1
1782 010766 177777    -1
1783 010770 177777    -1
1784 ;IF A TRAP TO 4 OCCURS COME HERE.
1784 010772 011602    HHB10: MOV        (SP),R2   ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
1785 010774 020227 010664    CMP         R2,#HHB2+2
1786 011000 001405    BEQ         1$          ;BRANCH IF YES.
1787 011002 020227 010666    CMP         R2,#HHB2+4
1788 011006 001402    BEQ         1$          ;BRANCH IF YES.
1789 011010 000137 036166    JMP         @#CPSPUR    ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1790 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
1791 011014 022626    1$: CMP        (SP)+,(SP)+
1792 011016 010237 001236    MOV        R2,@#$TMP2
1793 011022 104002    2$: ERROR      2          ;ODD ADRES
1794 011024 000430    BR          HHBDONE    ;BUT FDSTX IN ST 771
```



```
1795
1796
1797 011026 012737 010742 001240 ;REPORT RESULT INCORRECT:
1798 011034 012737 010722 001242 HHB15: MOV #HHBTP2,@#$TMP3
1799 011042 012737 010752 001244 MOV #HHBTP1,@#$TMP4
1800 011050 104002 1$: ERROR 2 ;BAD DATA X11*0 ST 3127
1801 011052 000415 BR HHB DONE
1802
1803 ;REPORT R0 INCORRECT:
1804 011054 012737 010764 001240 HHB20: MOV #HHBBF1+2,@#$TMP3
1805 011062 010037 001242 MOV R0,@#$TMP4
1806 011066 104002 1$: ERROR 2 ;R0 INCORRECT.
1807 011070 000406 BR HHB DONE
1808 ;REPORT FPS INCORRECT:
1809 011072 010537 001240 HHB25: MOV R5,@#$TMP3
1810 011076 012737 000204 001244 MOV #204,@#$TMP5
1811 011104 104002 1$: ERROR 2 ;FPSX
1812
1813 HHB DONE:
(1) 011106 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1814 ;*****
(3) ;*TEST 25 NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
(4) ;*
(4) ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
(4) ;*SOURCE FLOWS. THE NEG D INSTRUCTION
(4) ;*IS USED TO TEST MODE 5
(4) ;*
(3) ;*****
(2) 011110 000004 TST25: SCOPE
1815
1816 011112 IIB1:
(1) 011112 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1817 011114 012700 011222 MOV #IIBTP1,R0 ;SET UP THE DATA BUFFER.
1818 011120 012701 011252 MOV #IIBBF0,R1
1819 011124 012702 000010 MOV #10,R2
1820 011130 012021 1$: MOV (R0)+,(R1)+
1821 011132 077202 SOB R2,1$
1822 011134 012700 000200 MOV #200,R0 ;SET FD.
1823 011140 170100 LDFPS R0
1824 011142 012700 011264 MOV #IIBBF1+2,R0 ;SET UP THE OPERAND ADDRESS.
1825 011146 012737 011162 001236 MOV #IIB2,@#$TMP2
1826 011154 012737 011272 000004 MOV #IIB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
1827
1828 011162 170750 IIB2: NEG D @-(R0) ;TEST INSTRUCTION.
1829
1830 011164 170205 STFPS R5 ;GET FPS.
1831 011166 012701 011252 MOV #IIBBF0,R1 ;CHECK RESULT.
1832 011172 012702 000004 MOV #4,R2
1833 011176 005721 1$: TST (R1)+
1834 011200 001052 BNE IIB15 ;BRANCH IF INCORRECT.
1835 011202 077203 SOB R2,1$
1836 011204 020027 011262 CMP R0,#IIBBF1 ;IS R0 CORRECT?
```

```
1837 011210 001061          BNE      IIB20          ;BRANCH IF INCORRECT.
1838 011212 022705 000204    CMP      #204,R5       ;IS THE FPS CORRECT?
1839 011216 001065          BNE      IIB25          ;BRANCH IF INCORRECT.
1840 011220 000472          BR       IIBDONE
1841
1842          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
1843 011222 000176          IIBTP1: 176
1844 011224 177074          177074
1845 011226 127374          127374
1846 011230 157677 011252 177777 157677,IIBBF0,-1,-1,-1
      011236 177777 177777
1847 011242 000000          IIBTP2: 0
1848 011244 000000          0
1849 011246 000000          0
1850 011250 000000          0
1851 011252 177777          IIBBF0: -1
1852 011254 177777          -1
1853 011256 177777          -1
1854 011260 177777          -1
1855 011262 177777          IIBBF1: -1
1856 011264 177777          -1
1857 011266 177777          -1
1858 011270 177777          -1
1859
1860          ;IF A TRAP TO 4 OCCURS COME HERE.
1861 011272 011602          IIB10: MOV      (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
1862 011274 020227 011164    CMP      R2,#IIB2+2
1863 011300 001405          BEQ      1$              ;BRANCH IF YES.
1864 011302 020227 011166    CMP      R2,#IIB2+4
1865 011306 001402          BEQ      1$              ;BRANCH IF YES.
1866 011310 000137 036166    JMP      @#CPSPUR        ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1867          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
1868 011314 022626          1$:  CMP      (SP)+,(SP)+
1869 011316 010237 001236    MOV      R2,@#STMP2
1870 011322 104002          2$:  ERROR   2              ;ODD ADRES
1871 011324 000430          BR       IIBDONE        ;BUT FDSTX IN ST 771
1872
1873          ;REPORT RESULT INCORRECT:
1874 011326 012737 011242 001240 IIB15: MOV      #IIBTP2,@#STMP3
1875 011334 012737 011222 001242    MOV      #IIBTP1,@#STMP4
1876 011342 012737 011252 001244    MOV      #IIBBF0,@#STMP5
1877 011350 104002          1$:  ERROR   2              ;BAD DATA X11*0 ST 3127
1878 011352 000415          BR       IIBDONE
1879
1880          ;REPORT R0 INCORRECT:
1881 011354 012737 011262 001240 IIB20: MOV      #IIBBF1,@#STMP3
1882 011362 010037 001242    MOV      R0,@#STMP4
1883 011366 104002          1$:  ERROR   2              ;R0 BADX
1884 011370 000406          BR       IIBDONE
1885          ;REPORT FPS INCORRECT:
1886 011372 010537 001240 IIB25: MOV      R5,@#STMP3
1887 011376 012737 000204 001244    MOV      #204,@#STMP5
1888 011404 104002          1$:  ERROR   2              ;FPSX
1889
1890          IIBDONE:
      (1) 011406 104413          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?).

(1)
(1)
(1)
(1)
1891
1892
(3)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 011410 000004
1893
1894 011412
(1) 011412 104414
1895 011414 012700 011524
1896 011420 012701 011546
1897 011424 012702 000004
1898 011430 012021
1899 011432 077202
1900 011434 012700 000200
1901 011440 170100
1902 011442 012700 011537
1903 011446 012737 011462 001236
1904 011454 012737 011566 000004
1905
1906 011462 170660 000007
1907
1908 011466 170205
1909 011470 012701 011546
1910 011474 012702 000004
1911 011500 005721
1912 011502 001047
1913 011504 077203
1914 011506 020027 011537
1915 011512 001043
1916 011514 022705 000204
1917 011520 001053
1918 011522 000467
1919
1920
1921 011524 000177
1922 011526 161524
1923 011530 131273
1924 011532 107174 000000
1925 011536 000000
1926 011540 000000
1927 011542 000000
1928 011544 000000
1929 011546 177777
1930 011550 177777
1931 011552 177777
1932 011554 177777
1933 011556 177777

```
*****  
*TEST 26      NEGF, ABSF AND TSTF SOURCE MODE 6 TEST  
*  
*THIS IS A TEST THE NEGF, ABSF AND TSTF  
*SOURCE FLOWS. THE ABSD INSTRUCTION  
*IS USED TO TEST MODE 6  
*  
*****  
TST26: SCOPE  
  
JJB1:      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.  
           MOV        #JJBTP1,R0      ;SET UP THE DATA BUFFER.  
           MOV        #JJBFB0,R1  
           MOV        #4,R2  
1$:        MOV        (R0)+,(R1)+  
           SOB        R2,1$  
           MOV        #200,R0      ;SET FD.  
           LDFPS     R0  
           MOV        #JJBFB0-7,R0    ;SET UP THE OPERAND ADDRESS.  
           MOV        #JJB2,@#STMP2  
           MOV        #JJB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.  
  
JJB2:      ABSD      7(R0)      ;TEST INSTRUCTION.  
  
           STFPS     R5      ;GET FPS.  
           MOV        #JJBFB0,R1      ;CHECK RESULT.  
           MOV        #4,R2  
1$:        TST        (R1)+  
           BNE        JJB15      ;BRANCH IF INCORRECT.  
           SOB        R2,1$  
           CMP        R0,#JJBFB0-7    ;IS R0 CORRECT?  
           BNE        JJB15      ;BRANCH IF INCORRECT.  
           CMP        #204,R5      ;IS THE FPS CORRECT?  
           BNE        JJB20      ;BRANCH IF INCORRECT.  
           BR         JJBDONE  
  
;THESE ARE TEST DATA TABLES AND DATA BUFFER.  
JJBTP1: 177  
         161524  
         131273  
         107174,  
JJBTP2: 0  
         0  
         0  
         0  
JJBFB0: -1  
         -1  
         -1  
         -1  
JJBFB1: -1
```

```
1934 011560 177777 -1
1935 011562 177777 -1
1936 011564 177777 -1
1937
1938 ;IF A TRAP TO 4 OCCURS COME HERE.
1939 011566 011602 JJB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
1940 011570 020227 011464 CMP R2,#JJB2+2
1941 011574 001405 BEQ 1$ ;BRANCH IF YES.
1942 011576 020227 011466 CMP R2,#JJB2+4
1943 011602 001402 BEQ 1$ ;BRANCH IF YES.
1944 011604 000137 036166 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
1945 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
1946 011610 022626 1$: CMP (SP)+,(SP)+
1947 011612 010237 001236 MOV R2,@#STMP2
1948 011616 104002 2$: ERROR 2 ;ODD ADRES
1949 011620 000430 BR JJB DONE ;BUT FDSTX IN ST 771
1950
1951 ;REPORT RESULT INCORRECT:
1952 011622 012737 011536 001240 JJB15: MOV #JJBTP2,@#STMP3
1953 011630 012737 011524 001242 MOV #JJBTP1,@#STMP4
1954 011636 012737 011546 001244 MOV #JJBFB0,@#STMP5
1955 011644 104002 1$: ERROR 2 ;BAD DATA X11*0 ST 3127
1956 011646 000415 BR JJB DONE
1957
1958 ;REPORT R0 INCORRECT:
1959 011650 012737 011537 001240 JJB20: MOV #JJBFB0-7,@#STMP3
1960 011656 010037 001242 MOV R0,@#STMP4
1961 011662 104002 1$: ERROR 2 ;R0 BADX
1962 011664 000406 BR JJB DONE
1963 ;REPORT FPS INCORRECT:
1964 011666 010537 001240 JJB25: MOV R5,@#STMP3
1965 011672 012737 000204 001244 MOV #204,@#STMP5
1966 011700 104002 1$: ERROR 2 ;FPSX
1967 011702 JJB DONE:
(1) 011702 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
1968 ;*****
(3) ;*TEST 27 NEGf, ABSF AND TSTF SOURCE MODE 7 TEST
(4) ;*
(4) ;*THIS IS A TEST THE NEGf, ABSF AND TSTF
(4) ;*SOURCE FLOWS. THE ABSD INSTRUCTION
(4) ;*IS USED TO TEST MODE 6
(4) ;*
(3) ;*****
(2) 011704 000004 TST27: SCOPE
1969
1970 011706 KKB1:
(1) 011706 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
1971 011710 012700 012020 MOV #KKBTP1,R0 ;SET UP THE DATA BUFFER.
1972 011714 012701 012050 MOV #KKBFB0,R1
1973 011720 012702 000010 MOV #10,R2
1974 011724 012021 1$: MOV (R0)+,(R1)+
1975 011726 077202 SOB R2,1$
```



```

CJKDD8 KEF11-A DIAG PART 2          MACY11 30A(1052) 12-MAR-80 07:25 M 5
CJKDD8.P11 12-MAR-80 07:22          T27      NEG, ABSF AND TSTF SOURCE MODE 7 TEST          PAGE 1-47          SEQ 0064

1976 011730 012700 000200          MOV      #200,R0          ;SET FD.
1977 011734 170100          LDFPS   R0
1978 011736 012700 012051          MOV      #KKBBF1-7,R0    ;SET UP THE OPERAND ADDRESS.
1979 011742 012737 011756 001236    MOV      #KKB2,@#$TMP2
1980 011750 012737 012070 000004    MOV      #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
1981
1982 011756 170770 000007          KKB2:   NEGD      @7(R0)          ;TEST INSTRUCTION.
1983
1984 011762 170205          STFPS   R5          ;GET FPS.
1985 011764 012701 012050          MOV      #KKBBF0,R1          ;CHECK RESULT.
1986 011770 012702 000004          MOV      #4,R2
1987 011774 005721          1$:    TST      (R1)+
1988 011776 001052          BNE     KKB15          ;BRANCH IF INCORRECT.
1989 012000 077203          SOB     R2,1$
1990 012002 020027 012051          CMP     R0,#KKBBF1-7    ;IS R0 CORRECT?
1991 012006 001061          BNE     KKB20          ;BRANCH IF INCORRECT.
1992 012010 022705 000204          CMP     #204,R5        ;IS THE FPS CORRECT?
1993 012014 001056          BNE     KKB20          ;BRANCH IF INCORRECT.
1994 012016 000472          BR      KKBDONE
1995
1996          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
1997 012020 000177          KKBTP1: 177
1998 012022 167574          167574
1999 012024 137271          137271
2000 012026 107675 012050 177777    107675,KKBBF0,-1,-1,-1
2001 012034 177777 177777
2002 012040 000000          KKBTP2: 0
2003 012042 000000          0
2004 012044 000000          0
2005 012046 000000          0
2006 012050 177777          KKBBF0: -1
2007 012052 177777          -1
2008 012054 177777          -1
2009 012056 177777          -1
2010 012060 177777          KKBBF1: -1
2011 012062 177777          -1
2012 012064 177777          -1
2013 012066 177777          -1
2014
2015          ;IF A TRAP TO 4 OCCURS COME HERE.
2016 012070 011602          KKB10: MOV      (SP),R2          ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2017 012072 020227 011760          CMP     R2,#KKB2+2
2018 012076 001405          BEQ     1$          ;BRANCH IF YES.
2019 012100 020227 011762          CMP     R2,#KKB2+4
2020 012104 001402          BEQ     1$          ;BRANCH IF YES.
2021 012106 000137 036166          JMP     @#CPSPUR      ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2022          ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2023 012112 022626          1$:    CMP     (SP)+,(SP)+
2024 012114 010237 001236          MOV     R2,@#$TMP2
2025 012120 104002          2$:    ERROR   2          ;ODD ADRES
2026 012122 000430          BR      KKBDONE      ;BUT FDSTX IN ST 771
2027
2028          ;REPORT RESULT INCORRECT:
2029 012124 012737 012040 001240    KKB15: MOV      #KKBTP2,@#$TMP3
2030 012132 012737 012020 001242    MOV     #KKBTP1,@#$TMP4

```

```

2031 012140 012737 012050 001244      MOV      #KKBBF0,@#$TMP5
2032 012146 104002                1$:      ERROR      2      ;BAD DATA X11*0 ST 3127
2033 012150 000415                BR        KKBDONE
2034
2035      ;REPORT R0 INCORRECT:
2036 012152 012737 012051 001240  KKB20:  MOV      #KKBBF1-7,@#$TMP3
2037 012160 010037 001242                MOV      R0,@#$TMP4
2038 012164 104002                1$:      ERROR      2      ;R0 BADX
2039 012166 000406                BR        KKBDONE
2040      ;REPORT FPS INCORRECT:
2041 012170 010537 001240  KKB25:  MOV      R5,@#$TMP3
2042 012174 012737 000204 001244  MOV      #204,@#$TMP5
2043 012202 104002                1$:      ERROR      2      ;FPSX
2044
2045      KKBDONE:
(1) 012204 104413                RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
(1)                                     ;SEE IF THE USER HAS EXPRESSED
(1)                                     ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                                     ;THE USER TYPED CONTROL G?).
2046      ;*****
(3)      ;*TEST 30      NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
(4)      ;*
(4)      ;*THIS IS A TEST THE NEGF, ABSF AND TSTF
(4)      ;*SOURCE FLOWS. THE NEGD INSTRUCTION
(4)      ;*IS USED TO TEST MODE 6
(4)      ;*
(3)      ;*****
(2) 012206 000004      TST30:  SCOPE
2047 012210      LLB1:
(1) 012210 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
2048 012212 012700 012310      MOV      #LLBTP1,R0      ;SET UP THE DATA BUFFER.
2049 012216 012701 012330      MOV      #LLBBF0,R1
2050 012222 012702 000004      MOV      #4,R2
2051 012226 012021      1$:      MOV      (R0)+,(R1)+
2052 012230 077202      SOB      R2,1$
2053 012232 012700 000200      MOV      #200,R0      ;SET FD.
2054 012236 170100      LDFPS    R0
2055 012240 012737 012254 001236      MOV      #LLB2,@#$TMP2
2056 012246 012737 012350 000004      MOV      #LLB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2057
2058 012254 170767 000050      LLB2:  NEGD      LLBBF0      ;TEST INSTRUCTION.
2059
2060 012260 170205      STFPS    R5      ;GET FPS.
2061 012262 012701 012330      MOV      #LLBBF0,R1      ;CHECK RESULT.
2062 012266 012702 000004      MOV      #4,R2
2063 012272 005721      1$:      TST      (R1)+
2064 012274 001043      BNE      LLB15      ;BRANCH IF INCORRECT.
2065 012276 077203      SOB      R2,1$
2066 012300 022705 000204      CMP      #204,R5      ;IS THE FPS CORRECT?
2067 012304 001052      BNE      LLB25      ;BRANCH IF INCORRECT.
2068 012306 000457      BR        LLBDONE
2069
2070      ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2071 012310 000127      LLBTP1: 127
2072 012312 137475                137475
    
```


2073 012314 147372
2074 012316 117057
2075 012320 000000
2076 012322 000000
2077 012324 000000
2078 012326 000000
2079 012330 177777
2080 012332 177777
2081 012334 177777
2082 012336 177777
2083 012340 177777
2084 012342 177777
2085 012344 177777
2086 012346 177777
2087
2088
2089 012350 011602 012256
2090 012352 020227 012256
2091 012356 001405
2092 012360 020227 012260
2093 012364 001402
2094 012366 000137 036166
2095
2096 012372 022626
2097 012374 010237 001236
2098 012400 104002
2099 012402 000421
2100
2101
2102 012404 012737 012320 001240
2103 012412 012737 012310 001242
2104 012420 012737 012330 001244
2105 012426 104002
2106 012430 000406
2107
2108 012432 010537 001240
2109 012436 012737 000204 001244
2110 012444 104002
2111
2112 012446
(1) 012446 104413
(1)
(1)
(1)
(1)
(1)
2113
(3)
(4)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 012450 000004
2114
2115 012452

147372
117057
LLBTP2: 0
0
0
0
LLBBF0: -1
-1
-1
-1
LLBBF1: -1
-1
-1
-1
-1
-1
:IF A TRAP TO 4 OCCURS COME HERE.
LLB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
CMP R2,#LLB2+2
BEQ 1\$;BRANCH IF YES.
CMP R2,#LLB2+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 2 ;ODD ADRES
BR LLBDONE ;BUT FDSTX IN ST 771
:REPORT RESULT INCORRECT:
LLB15: MOV #LLBTP2,@#STMP3
MOV #LLBTP1,@#STMP4
MOV #LLBBF0,@#STMP5
1\$: ERROR 2 ;BAD DATA X11*0 ST 3127
BR LLBDONE
:REPORT FPS INCORRECT:
LLB25: MOV R5,@#STMP3
MOV #204,@#STMP5
1\$: ERROR 2 ;FPSX
LLBDONE:
RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
;SEE IF THE USER HAS EXPRESSED
;THE DESIRE TO CHANGE THE SOFTWARE
;VIRTUAL CONSOLE SWITCH REGISTER (HAS
;THE USER TYPED CONTROL G?).
:*****
:*TEST 31 NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST
:*
:*THIS IS A TEST THE NEGF, ABSF AND TSTF
:*SOURCE FLOWS. THE ABSD INSTRUCTION
:*IS USED TO TEST MODE 7
:*
:*****
TST31: SCOPE
MMB1:

```
(1) 012452 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2116 012454 012700 012552 MOV #MMBTP1,R0 ;SET UP THE DATA BUFFER.
2117 012460 012701 012602 MOV #MMBBF0,R1
2118 012464 012702 000010 MOV #10,R2
2119 012470 012021 1$: MOV (R0)+,(R1)+
2120 012472 077202 SOB R2,1$
2121 012474 012700 000200 MOV #200,R0 ;SET FD.
2122 012500 170100 LDFPS R0
2123 012502 012737 012516 001236 MOV #MMB2,@#$TMP2
2124 012510 012737 012622 000004 MOV #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
2125
2126 012516 170677 000070 MMB2: ABSD @MMBBF1 ;TEST INSTRUCTION.
2127
2128 012522 170205 STFPS R5 ;GET FPS.
2129 012524 012701 012602 MOV #MMBBF0,R1 ;CHECK RESULT.
2130 012530 012702 000004 MOV #4,R2
2131 012534 005721 1$: TST (R1)+
2132 012536 001047 BNE MMB15 ;BRANCH IF INCORRECT.
2133 012540 077203 SOB R2,1$
2134 012542 022705 000204 CMP #204,R5 ;IS THE FPS CORRECT?
2135 012546 001056 BNE MMB25 ;BRANCH IF INCORRECT.
2136 012550 000463 BR MMBDONE
2137
2138 ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
2139 012552 000137 MMBTP1: 137
2140 012554 045607 045607
2141 012556 101230 101230
2142 012560 045607 012602 177777 45607,MMBBF0,-1,-1,-1
2143 012566 177777 177777
2144 012572 000000 MMBTP2: 0
2145 012574 000000 0
2146 012576 000000 0
2147 012600 000000 0
2148 012602 177777 MMBBF0: -1
2149 012604 177777 -1
2150 012606 177777 -1
2151 012610 177777 -1
2152 012612 177777 MMBBF1: -1
2153 012614 177777 -1
2154 012616 177777 -1
2155 012620 177777 -1
2156
2157 012622 011602 ;IF A TRAP TO 4 OCCURS COME HERE.
2158 012624 020227 012520 MMB10: MOV (SP),R2 ;SEE IF THE TRAP OCCURRED ON THE TEST INSTRUCTION.
2159 012630 001405 CMP R2,#MMB2+2
2160 012632 020227 012522 BEQ 1$ ;BRANCH IF YES.
2161 012636 001402 CMP R2,#MMB2+4
2162 012640 000137 036166 BEQ 1$ ;BRANCH IF YES.
2163 JMP @#CPSPUR ;OTHERWISE GO REPORT SPURIOUS TRAP TO 4.
2164 012644 022626 ;REPORT AN FDST FLOW FAILURE RESULTED IN A TRAP TO 4.
2165 012646 010237 001236 1$: CMP (SP)+,(SP)+
2166 012652 104002 MOV R2,@#$TMP2
2167 012654 000421 2$: ERROR 2 ;ODD ADRES
BR MMBDONE ;BUT FDSTX IN ST 771
2168
2169 ;REPORT RESULT INCORRECT:
```



```

2170 012656 012737 012572 001240 MMB15: MOV #MMBTP2,@#$TMP3
2171 012664 012737 012552 001242 MOV #MMBTP1,@#$TMP4
2172 012672 012737 012602 001244 MOV #MMBBF0,@#$TMP5
2173 012700 104002 1$: ERROR 2 ;BAD DATA x11*0 ST 3127
2174 012702 000406 BR MMBDONE
2175 ;REPORT FPS INCORRECT:
2176 012704 010537 001240 MMB25: MOV R5,@#$TMP3
2177 012710 012737 000204 001244 MOV #204,@#$TMP5
2178 012716 104002 1$: ERROR 2 ;FPSX
2179
2180 012720 MMBDONE:
(1) 012720 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2187 ;*****
(3) ;*TEST 32 SPECIAL DEST, MODE 0, TEST
(4) ;*
(4) ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4) ;*MODE 0 USING THE NEGD INSTR.
(4) ;*
(3) ;*****
(2) 012722 000004 TST32: SCOPE
2188
2189 012724 NNB1:
(1) 012724 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2190 012726 012700 000200 MOV #200,R0 ;SET FD.
2191 012732 170100 LDFPS R0
2192 012734 012700 013022 MOV #NNBTP1,R0 ;SET UP ACO.
2193 012740 172410 LDD (R0),ACO
2194 012742 012737 012750 001236 MOV #NNB2,@#$TMP2
2195
2196 012750 170700 NNB2: NEGD ACO ;TEST INSTRUCTION.
2197
2198 012752 170205 STFPS R5 ;GET FPS.
2199 012754 012700 000200 MOV #200,R0 ;SET FD.
2200 012760 170100 LDFPS R0
2201 012762 012700 013042 MOV #NNBBF0,R0 ;GET THE RESULT.
2202 012766 174010 STD ACO,(R0)
2203 012770 012700 013042 MOV #NNBBF0,R0 ;IS THE RESULT CORRECT?
2204 012774 012701 013032 MOV #NNBTP2,R1
2205 013000 012702 000004 MOV #4,R2
2206 013004 022021 1$: CMP (R0)+,(R1)+
2207 013006 001021 BNE NNB10 ;BRANCH IF INCORRECT.
2208 013010 077203 SOB R2,1$
2209 013012 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
2210 013016 001033 BNE NNB15 ;BRANCH IF INCORRECT.
2211 013020 000440 BR NNBDONE
2212
2213 ;THESE ARE DATA TABLES AND A DATA BUFFER.
2214 013022 013572 NNBTP1: 013572
2215 013024 046013 46013
2216 013026 057246 57246
2217 013030 013570 013570
2218 013032 113572 NNBTP2: 113572
    
```

```

2219 013034 046013 46013
2220 013036 057246 57246
2221 013040 013570 013570
2222 013042 000000 NBBF0: 0
2223 013044 000000 0
2224 013046 000000 0
2225 013050 000000 0
2226
2227 ;REPORT RESULT INCORRECT:
2228 013052 012737 013042 001240 NNB10: MOV #NNBF0,@#$TMP3
2229 013060 012737 013032 001242 MOV #NNBTP2,@#$TMP4
2230 013066 023737 013022 013042 CMP @NNBTP1,@NNBF0
2231 013074 001002 BNE NNB11
2232 013076 104001 1$: ERROR 1 ;E10*200X ST 336
2233 013100 000410 BR NNBDONE
2234
2235 ;REPORT RESULT INCORRECT:
2236 013102 NNB11:
2237 013102 104001 1$: ERROR 1 ;BAD DATA NEGF
2238 013104 000406 BR NNBDONE
2239
2240 ;REPORT FPS INCORRECT:
2241 013106 010537 001242 001240 NNB15: MOV R5,@#$TMP4
2242 013112 012737 000210 001240 MOV #210,@#$TMP3
2243 013120 104001 1$: ERROR 1 ;FPSX
2244
2245 NNBDONE:
(1) 013122 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2246 ::*****
(3) ;*TEST 33 SPECIAL DEST, MODE 1, TEST
(4) ;*
(4) ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4) ;*MODE 1 USING THE NEGD INSTR.
(4) ;*
(3) ;:*****
(2) 013124 000004 TST33: SCOPE
2247
2248 013126 104414 OOB1:
(1) 013126 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2249 013130 012701 013240 MOV #OOBTP1,R1 ;SET UP THE DATA BUFFER.
2250 013134 012700 013250 MOV #OOBTP2,R0
2251 013140 012702 000004 MOV #4,R2
2252 013144 012021 1$: MOV (R0)+,(R1)+
2253 013146 077202 SOB R2,1$
2254 013150 012700 013240 MOV #OOBTP1,R0
2255 013154 042710 100000 BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
2256 013160 012737 013174 001236 MOV #OOB2,@#$TMP2
2257 013166 012701 000200 MOV #200,R1 ;SET FD.
2258 013172 170101 LDFPS R1
2259
2260 013174 170710 OOB2: NEGD (R0) ;TEST INSTRUCTION.
2261 013176 170205 STFPS R5 ;GET FPS.
    
```



```

2262 013200 012701 013240      MOV      #00BTP1,R1      ;IS THE RESULT CORRECT.
2263 013204 012702 013250      MOV      #00BTP2,R2
2264 013210 012703 000004      MOV      #4,R3
2265 013214 022122      1$:    CMP      (R1)+,(R2)+
2266 013216 001020      BNE      00B10          ;BRANCH IF INCORRECT.
2267 013220 077303      SOB      R3,1$
2268 013222 022700 013240      CMP      #00BTP1,R0      ;IS R0 CORRECT.
2269 013226 001024      BNE      00B15          ;BRANCH IF INCORRECT.
2270 013230 022705 000210      CMP      #210,R5        ;IS THE FPS CORRECT?
2271 013234 001030      BNE      00B20          ;BRANCH IF INCORRECT.
2272 013236 000435      BR       00BDONE
2273
2274      ;THESE ARE DATA TABLES AND A DATA BUFFER.
2275 013240 023245      00BTP1: 023245
2276 013242 026720      26720
2277 013244 122324      122324
2278
2279 013246 052672      52672
2280 013250 123245      00BTP2: 123245
2281 013252 026720      26720
2282 013254 122324      122324
2283 013256 052672      52672
2284
2285      ;REPORT RESULT INCORRECT:
2286 013260 012737 013240 001240      00B10: MOV      #00BTP1,@#$TMP3
2287 013266 012737 013250 001242      MOV      #00BTP2,@#$TMP4
2288 013274 104002      1$:    ERROR  2          ;BAD DATA
2289 013276 000415      BR       00BDONE
2290
2291      ;REPORT R0 INCORRECT:
2292 013300 012737 013240 001240      00B15: MOV      #00BTP1,@#$TMP3
2293 013306 010037 001242      MOV      R0,@#$TMP4
2294 013312 104002      1$:    ERROR  2          ;SPEC DESTX
2295 013314 000406      BR       00BDONE          ;ROX
2296
2297      ;REPORT FPS INCORRECT:
2298 013316 012737 000210 001240      00B20: MOV      #210,@#$TMP3
2299 013324 010537 001242      MOV      R5,@#$TMP4
2300 013330 104002      1$:    ERROR  2
2301
2302      00BDONE:
(1) 013332 104413      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                                     ;SEE IF THE USER HAS EXPRESSED
(1)                                     ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                                     ;THE USER TYPED CONTROL G?).
2303      ;*****
(3)      ;*TEST 34      SPECIAL DEST, MODE 2, TEST
(4)      ;*
(4)      ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4)      ;*MODE 2 USING THE NEGD INSTR.
(4)      ;*
(3)      ;*****
(2) 013334 000004      TST34: SCOPE
2304 013336      PPB1:
(1) 013336 104414      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
    
```

```

2305
2306 013340 012701 013450      MOV    #PPBTP1,R1      ;SET UP THE DATA BUFFER.
2307 013344 012700 013460      MOV    #PPBTP2,R0
2308 013350 012702 000004      MOV    #4,R2
2309 013354 012021      1$:  MOV    (R0)+,(R1)+
2310 013356 077202      SOB    R2,1$
2311 013360 012700 013450      MOV    #PPBTP1,R0
2312 013364 042710 100000      BIC    #100000,(R0)    ;MAKE OPERAND POSITIVE.
2313 013370 012737 013404 001236  MOV    #PPB2,@#$TMP2
2314 013376 012701 000200      MOV    #200,R1        ;SET FD.
2315 013402 170101      LDFPS  R1
2316
2317 013404 170720      PPB2:  NEG    (R0)+      ;TEST INSTRUCTION.
2318
2319 013406 170205      STFPS  R5              ;GET FPS.
2320 013410 012701 013450      MOV    #PPBTP1,R1      ;IS THE RESULT CORRECT.
2321 013414 012702 013460      MOV    #PPBTP2,R2
2322 013420 012703 000004      MOV    #4,R3
2323 013424 022122      1$:  CMP    (R1)+,(R2)+
2324 013426 001020      BNE    PPB10          ;BRANCH IF INCORRECT.
2325 013430 077303      SOB    R3,1$
2326 013432 022700 013460      CMP    #PPBTP1+10,R0   ;IS R0 CORRECT.
2327 013436 001024      BNE    PPB15          ;BRANCH IF INCORRECT.
2328 013440 022705 000210      CMP    #210,R5        ;IS THE FPS CORRECT?
2329 013444 001030      BNE    PPB20          ;BRANCH IF INCORRECT.
2330 013446 000435      BR     PPBDONE
2331
2332      ;THESE ARE DATA TABLES AND A DATA BUFFER.
2333 013450 023245      PPBTP1: 023245
2334 013452 026720      26720
2335 013454 122324      122324
2336 013456 052672      52672
2337 013460 123245      PPBTP2: 123245
2338 013462 026720      26720
2339 013464 122324      122324
2340 013466 052672      52672
2341
2342      ;REPORT RESULT INCORRECT:
2343 013470 012737 013450 001240  PPB10: MOV    #PPBTP1,@#$TMP3
2344 013476 012737 013460 001242  MOV    #PPBTP2,@#$TMP4
2345 013504 104001      1$:  ERROR 1          ;BAD DATA
2346 013506 000415      BR     PPBDONE
2347
2348      ;REPORT R0 INCORRECT:
2349 013510 012737 013460 001240  PPB15: MOV    #PPBTP1+10,@#$TMP3
2350 013516 010037 001242  MOV    R0,@#$TMP4
2351 013522 104001      1$:  ERROR 1          ;SPEC DESTX ROX
2352 013524 000406      BR     PPBDONE
2353
2354      ;REPORT FPS INCORRECT:
2355 013526 012737 000210 001240  PPB20: MOV    #210,@#$TMP3
2356 013534 010537 001242  MOV    R5,@#$TMP4
2357 013540 104001      1$:  ERROR 1
2358
2359 013542      PPBDONE:
(1) 013542 104413      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
  
```



```
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2360 :*****
(3) :*TEST 35 SPECIAL DEST, MODE 4, TEST
(4) :*
(4) :*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4) :*MODE 4 USING THE NEGD INSTR.
(4) :*
(3) :*****
(2) 013544 000004 TST35: SCOPE
2361 013546 QQB1:
(1) 013546 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2362 013550 012701 013662 MOV #QQBTP1,R1 ;SET UP THE DATA BUFFER.
2363 013554 012700 013702 MOV #QQBTP2,R0
2364 013560 012702 000004 MOV #4,R2
2365 013564 012021 1$: MOV (R0)+,(R1)+
2366 013566 077202 SOB R2,1$
2367 013570 012700 013672 MOV #QQBTP1+10,R0
2368 013574 042760 100000 177770 BIC #100000,-10(R0) ;MAKE OPERAND POSITIVE.
2369 013602 012737 013616 001236 MOV #QQB2,@#$TMP2
2370 013610 012701 000200 MOV #200,R1 ;SET FD.
2371 013614 170101 LDFPS R1
2372
2373 013616 170740 QQB2: NEGD -(R0) ;TEST INSTRUCTION.
2374
2375 013620 170205 STFPS R5 ;GET FPS.
2376 013622 012701 013662 MOV #QQBTP1,R1 ;IS THE RESULT CORRECT.
2377 013626 012702 013702 MOV #QQBTP2,R2
2378 013632 012703 000004 MOV #4,R3
2379 013636 022122 1$: CMP (R1)+,(R2)+
2380 013640 001024 BNE QQB10 ;BRANCH IF INCORRECT.
2381 013642 077303 SOB R3,1$
2382
2383 013644 022700 013662 CMP #QQBTP1,R0 ;IS R0 CORRECT.
2384 013650 001030 BNE QQB15 ;BRANCH IF INCORRECT.
2385 013652 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
2386 013656 001034 BNE QQB20 ;BRANCH IF INCORRECT.
2387 013660 000441 BR QQBDONE
2388
2389 ;THESE ARE DATA TABLES AND A DATA BUFFER.
2390 013662 023245 QQBTP1: 023245
2391 013664 026720 26720
2392 013666 122324 122324
2393 013670 052672 52672
2394 013672 177777 177777 177777 .WORD -1,-1,-1,-1
2395 013700 177777
2396 013702 123245 QQBTP2: 123245
2397 013704 026720 26720
2398 013706 122324 122324
2399 013710 052672 52672
2400
2401 013712 012737 013662 001240 ;REPORT RESULT INCORRECT:
2402 013720 012737 013702 001242 QQB10: MOV #QQBTP1,@#$TMP3
MOV #QQBTP2,@#$TMP4
```

```
2403 013726 104001 1$: ERROR 1 ;BAD DATA
2404 013730 000415 BR QQBDONE
2405
2406 :REPORT R0 INCORRECT:
2407 013732 012737 013662 001240 QQB15: MOV #QQBTP1,@#$TMP3
2408 013740 010037 001242 MOV R0,@#$TMP4
2409 013744 104001 1$: ERROR 1 ;SPEC DESTX R0X
2410 013746 000406 BR QQBDONE
2411
2412
2413 :REPORT FPS INCORRECT:
2414 013750 012737 000210 001240 QQB20: MOV #210,@#$TMP3
2415 013756 010537 001242 MOV R5,@#$TMP4
2416 013762 104001 1$: ERROR 1
2417
2418 013764 QQBDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) 013764 104413 ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2419
2420 :*****
(3) ;*TEST 36 SPECIAL DEST, MODE 3, TEST
(4) ;*
(4) ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4) ;*MODE 3 USING THE NEGD INSTR.
(4) ;*
(3) :*****
(2) 013766 000004 TST36: SCOPE
2421
2422 013770 RRB1:
(1) 013770 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2423 013772 012701 014110 MOV #RRBTP1,R1 ;SET UP THE DATA BUFFER.
2424 013776 012700 014120 MOV #RRBTP2,R0
2425 014002 012702 000004 MOV #4,R2
2426 014006 012021 1$: MOV (R0)+,(R1)+
2427 014010 077202 SOB R2,1$
2428 014012 012700 014130 MOV #RRBTP3,R0
2429 014016 012710 014110 MOV #RRBTP1,(R0)
2430 014022 042737 100000 014110 BIC #100000,@#RRBTP1 ;MAKE THE OPERAND POSITIVE.
2431 014030 012737 014044 001236 MOV #RRB2,@#$TMP2
2432 014036 012701 000200 MOV #200,R1 ;SET FD.
2433 014042 170101 LDFPS R1
2434
2435 014044 170730 RRB2: NEGD @(R0)+ ;TEST INSTRUCTION.
2436
2437 014046 170205 STFPS R5 ;GET FPS.
2438 014050 012701 014110 MOV #RRBTP1,R1 ;IS THE RESULT CORRECT.
2439 014054 012702 014120 MOV #RRBTP2,R2
2440 014060 012703 000004 MOV #4,R3
2441 014064 022122 1$: CMP (R1)+,(R2)+
2442 014066 001021 BNE RRB10 ;BRANCH IF INCORRECT.
2443 014070 077303 SOB R3,1$
2444 014072 022700 014132 CMP #RRBTP3+2,R0 ;IS R0 CORRECT.
2445 014076 001025 BNE RRB15 ;BRANCH IF INCORRECT.
```



```

2446 014100 022705 000210      CMP      #210,R5      ;IS THE FPS CORRECT?
2447 014104 001031              BNE      RRB20       ;BRANCH IF INCORRECT.
2448 014106 000436              BR       RRBDONE
2449
2450      ;THESE ARE DATA TABLES AND A DATA BUFFER.
2451 014110 023245      RRBTP1: 023245
2452 014112 026720              26720
2453 014114 122324              122324
2454 014116 052672              52672
2455 014120 123245      RRBTP2: 123245
2456 014122 026720              26720
2457 014124 123324              123324
2458 014126 052672              52672
2459 014130 014110      RRBTP3: RRBTP1
2460
2461      ;REPORT RESULT INCORRECT:
2462 014132 012737 014110 001240 RRB10: MOV      #RRBTP1,@#$TMP3
2463 014140 012737 014120 001242      MOV      #RRBTP2,@#$TMP4
2464 014146 104001      1$:      ERROR 1      ;BAD DATA
2465 014150 000415              BR       RRBDONE
2466
2467      ;REPORT R0 INCORRECT:
2468 014152 012737 014132 001240 RRB15: MOV      #RRBTP3+2,@#$TMP3
2469 014160 010037 001242      MOV      R0,@#$TMP4
2470 014164 104001      1$:      ERROR 1      ;SPEC DESTX ROX
2471 014166 000406              BR       RRBDONE
2472
2473      ;REPORT FPS INCORRECT:
2474 014170 012737 000210 001240 RRB20: MOV      #210,@#$TMP3
2475 014176 010537 001242      MOV      R5,@#$TMP4
2476 014202 104001      1$:      ERROR 1
2477
2478 014204      RRBDONE:
    (1) 014204 104413      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
    (1)      ;SEE IF THE USER HAS EXPRESSED
    (1)      ;THE DESIRE TO CHANGE THE SOFTWARE
    (1)      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
    (1)      ;THE USER TYPED CONTROL G?).
2479
2480      ;*****
    (3)      ;*TEST 37      SPECIAL DEST, MODE 5, TEST
    (4)      ;*
    (4)      ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
    (4)      ;*MODE 5 USING THE NEGD INSTR.
    (4)      ;*
    (3)      ;*****
    (2) 014206 000004      TST37:  SCOPE
2481 014210      SSB1:
    (1) 014210 104414      LPERR
2482 014212 012701 014332      MOV      #SSBTP1,R1      ;SET UP THE LOOP ON ERROR ADDRESS.
2483 014216 012700 014342      MOV      #SSBTP2,R0      ;SET UP THE DATA BUFFER.
2484 014222 012702 000004      MOV      #4,R2
2485 014226 012021      1$:      MOV      (R0)+,(R1)+
2486 014230 077202      SOB      R2,1$
2487 014232 012700 014354      MOV      #SSBTP3+2,R0
2488 014236 012760 014332 177776      MOV      #SSBTP1,-2(R0)
    
```

```

2489 014244 042737 100000 014332 BIC #100000,@#SSBTP1 ;MAKE THE OPERAND POSITIVE.
2490 014252 012737 014266 001236 MOV #SSB2,@#STMP2
2491 014260 012701 000200 MOV #200,R1 ;SET FD.
2492 014264 170101 LDFPS R1
2493
2494 014266 170750 SSB2: NEGD @-(R0) ;TEST INSTRUCTION.
2495
2496 014270 170205 STFPS R5 ;GET FPS.
2497 014272 012701 014332 MOV #SSBTP1,R1 ;IS THE RESULT CORRECT.
2498 014276 012702 014342 MOV #SSBTP2,R2
2499 014302 012703 000004 MOV #4,R3
2500 014306 022122 1$: CMP (R1)+,(R2)+
2501 014310 001021 BNE SSB10 ;BRANCH IF INCORRECT.
2502 014312 077303 SOB R3,1$
2503 014314 022700 014352 CMP #SSBTP3,R0 ;IS R0 CORRECT.
2504 014320 001025 BNE SSB15 ;BRANCH IF INCORRECT.
2505 014322 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
2506 014326 001031 BNE SSB20 ;BRANCH IF INCORRECT.
2507 014330 000436 BR SSBDONE
2508
2509 ;THESE ARE DATA TABLES AND A DATA BUFFER.
2510 014332 023245 SSBTP1: 023245
2511 014334 026720 26720
2512 014336 122324 122324
2513 014340 052672 52672
2514 014342 123245 SSBTP2: 123245
2515 014344 026270 26270
2516 014346 122324 122324
2517 014350 052672 52672
2518 014352 014332 SSBTP3: SSBTP1
2519
2520 ;REPORT RESULT INCORRECT:
2521 014354 012737 014332 001240 SSB10: MOV #SSBTP1,@#STMP3
2522 014362 012737 014342 001242 MOV #SSBTP2,@#STMP4
2523 014370 104001 1$: ERROR 1 ;BAD DATA
2524 014372 000415 BR SSBDONE
2525
2526 ;REPORT R0 INCORRECT:
2527 014374 012737 014352 001240 SSB15: MOV #SSBTP3,@#STMP3
2528 014402 010037 001242 MOV R0,@#STMP4
2529 014406 104001 1$: ERROR 1 ;SPEC DESTX R0X
2530 014410 000406 BR SSBDONE
2531
2532 ;REPORT FPS INCORRECT:
2533 014412 012737 000210 001240 SSB20: MOV #210,@#STMP3
2534 014420 010537 001242 MOV R5,@#STMP4
2535 014424 104001 1$: ERROR 1
2536
2537 SSBDONE:
(1) 014426 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2538 ;*****
(3) ;*TEST 40 SPECIAL DEST, FLOATING MODE 2, TEST
    
```



```
(4)
(4)
(4)
(4)
(3)
(2) 014430 000004
2539 014432
(1) 014432 104414
2540 014434 012701 014544
2541 014440 012700 014554
2542 014444 012702 000004
2543 014450 012021
2544 014452 077202
2545 014454 012700 014544
2546 014460 042710 100000
2547 014464 012737 014500 001236
2548 014472 012701 000000
2549 014476 170101
2550
2551 014500 170720
2552
2553 014502 170205
2554 014504 012701 014544
2555 014510 012702 014554
2556 014514 012703 000004
2557 014520 022122
2558 014522 001020
2559 014524 077303
2560 014526 022700 014550
2561 014532 001024
2562 014534 022705 000010
2563 014540 001030
2564 014542 000435
2565
2566
2567 014544 023245
2568 014546 026720
2569 014550 122324
2570 014552 052672
2571 014554 123245
2572 014556 026720
2573 014560 122324
2574 014562 052672
2575
2576
2577 014564 012737 014544 001240
2578 014572 012737 014554 001242
2579 014600 104001
2580 014602 000415
2581
2582
2583 014604 012737 014550 001240
2584 014612 010037 001242
2585 014616 104001
2586 014620 000406
2587

;*****
;THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
;MODE 2 USING THE NEGF INSTR.
;*****
TST40: SCOPE
TTB1:
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #TTBTP1,R1 ;SET UP THE DATA BUFFER.
MOV #TTBTP2,R0
MOV #4,R2
1$: MOV (R0)+,(R1)+
SOB R2,1$
MOV #TTBTP1,R0
BIC #100000,(R0) ;MAKE OPERAND POSITIVE.
MOV #TTB2,@#$TMP2
MOV #000,R1 ;SET FD.
LDFPS R1
TTB2: NEG (R0)+ ;TEST INSTRUCTION.
STFPS R5 ;GET FPS.
MOV #TTBTP1,R1 ;IS THE RESULT CORRECT.
MOV #TTBTP2,R2
MOV #4,R3
1$: CMP (R1)+,(R2)+
BNE TTB10 ;BRANCH IF INCORRECT.
SOB R3,1$
CMP #TTBTP1+4,R0 ;IS R0 CORRECT.
BNE TTB15 ;BRANCH IF INCORRECT.
CMP #010,R5 ;IS THE FPS CORRECT?
BNE TTB20 ;BRANCH IF INCORRECT.
BR TTBDONE

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

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

;REPORT R0 INCORRECT:
TTB15: MOV #TTBTP1+4,@#$TMP3
MOV R0,@#$TMP4
1$: ERROR 1 ;SPEC DESTX ROX
BR TTBDONE
```

```

2588 ;REPORT FPS INCORRECT:
2589 014622 012737 000010 001240 ITB20: MOV #010,@#$TMP3
2590 014630 010537 001242 MOV R5,@#$TMP4
2591 014634 104001 1$: ERROR 1
2592
2593 014636 TTBDONE:
(1) 014636 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
2594 ;*****
(3) ;*TEST 41 SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST
(4) ;*
(4) ;THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
(4) ;*MODE 2(IMMEDIATE) USING THE NEGD INSTR.
(4) ;*
(3) ;*****
(2) 014640 000004 TST41: SCOPE
2595 014642 UUB1:
(1) 014642 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2596 014644 012700 014770 MOV #UUBTP2,R0
2597 014650 012701 014716 MOV #UUBTP1,R1 ;SET UP THE DATA BUFFER.
2598 014654 012702 000004 MOV #4,R2
2599 014660 012021 1$: MOV (R0)+,(R1)+
2600 014662 077202 SOB R2,1$
2601 014664 012700 014716 MOV #UUBTP1,R0
2602 014670 042737 100000 014716 BIC #100000,@#UUBTP1 ;MAKE THE OPERAND POSITIVE.
2603 014676 012737 014714 001236 MOV #UUB2,@#$TMP2
2604 014704 012701 000200 MOV #200,R1 ;SET FD.
2605 014710 170101 LDFPS R1
2606 014712 005001 CLR R1
2607
2608 014714 170727 UUB2: NEGD (R7)+ ;TEST INSTRUCTION.
2609 014716 005201 005201 005201 UUBTP1: 5201,5201,5201,5201
014724 005201
2610 ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
2611 014726 170205 STFPS R5 ;GET FPS.
2612 014730 012703 014716 MOV #UUBTP1,R3 ;IS THE RESULT CORRECT.
2613 014734 012702 014770 MOV #UUBTP2,R2
2614 014740 012704 000004 MOV #4,R4
2615 014744 022322 1$: CMP (R3)+,(R2)+
2616 014746 001014 BNE UUB10 ;BRANCH IF INCORRECT.
2617 014750 077403 SOB R4,1$
2618 014752 022701 000003 CMP #3,R1 ;WAS R1 INCREMENTED CORRECTLY.
2619 014756 001027 BNE UUB15 ;BRANCH IF INCORRECT.
2620 014760 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
2621 014764 001015 BNE UUB20 ;BRANCH IF INCORRECT.
2622 014766 000436 BR UUBDONE
2623
2624 ;THESE ARE DATA TABLE.
2625 014770 105201 UUBTP2: 105201
2626 014772 005201 5201
2627 014774 005201 5201
2628 014776 005201 5201
2629
    
```



```

2630 ;REPORT RESULT INCORRECT:
2631 015000 012737 014716 001240 UUB10: MOV #UUBTP1,@#$TMP3
2632 015006 012737 014770 001242 MOV #UUBTP2,@#$TMP4
2633 015014 104001 1$: ERROR 1 ;BAD DATA
2634 015016 000422 BR UBDONE
2635
2636 ;REPORT FPS INCORRECT:
2637 015020 012737 000210 001240 UUB20: MOV #210,@#$TMP3
2638 015026 010537 001242 MOV R5,@#$TMP4
2639 015032 104001 1$: ERROR 1 ;FPS
2640 015034 000413 BR UBDONE
2641
2642 ;REPORT PC INCORRECTLY INCREMENTED DURING EXECUTION.
2643 015036 162701 000003 UUB15: SUB #3,R1
2644 015042 006301 ASL R1
2645 015044 012702 014720 MOV #UUBTP1+2,R2
2646 015050 010237 001240 MOV R2,@#$TMP3
2647 015054 160102 SUB R1,R2
2648 015056 010237 001242 MOV R2,@#$TMP4
2649 015062 104001 1$: ERROR 1 ;PC BAD CONSTAND B GR7X
2650
2651 015064 UBDONE:
    (1) 015064 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
    (1) ;SEE IF THE USER HAS EXPRESSED
    (1) ;THE DESIRE TO CHANGE THE SOFTWARE
    (1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
    (1) ;THE USER TYPED CONTROL G?).
2652 ;*****
    (3) ;*TEST 42 SPECIAL DEST, MODE 6, TEST
    (4) ;*
    (4) ;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
    (4) ;*MODE 6 USING THE NEGD INSTR.
    (4) ;*
    (3) ;*****
    (2) 015066 000004 T42: SCOPE
2653 015070 XA01:
    (1) 015070 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2654 015072 012701 015214 MOV #XXBTP1,R1 ;SET UP THE DATA BUFFER.
2655 015076 012700 015224 MOV #XXBTP2,R0
2656 015102 012702 000004 MOV #4,R2
2657 015106 012021 1$: MOV (R0)+,(R1)+
2658 015110 077202 SOB R2,1$
2659 015112 012700 010013 MOV #XXBTP1-5201,R0
2660 015116 042737 100000 015214 BIC #100000,@#XXBTP1;MAKE OPERAND POSITIVE.
2661 015124 012737 015142 001236 MOV #XXB2,@#$TMP2
2662 015132 012701 000200 MOV #200,R1 ;SET FD.
2663 015136 170101 LDFPS R1
2664
2665 015140 005001 XXB2: CLR R1
2666 015142 170760 005201 NEG R1 ;TEST INSTRUCTION.
2667
2668 015146 170205 STFPS R5 ;GET FPS.
2669 015150 005701 TST R1
2670 015152 001030 BNE XXB25 ;WAS THE PC CORRECT AFTER EXECUTION?
2671 015154 012701 015214 MOV #XXBTP1,R1 ;IS THE RESULT CORRECT.
2672 015160 012702 015224 MOV #XXBTP2,R2
    
```

```
2673 015164 012703 000004  
2674 015170 022122  
2675 015172 001030  
2676 015174 077303  
2677 015176 022700 010013  
2678 015202 001034  
2679 015204 022705 000210  
2680 015210 001040  
2681 015212 000445  
2682  
2683  
2684 015214 023245  
2685 015216 026720  
2686 015220 122324  
2687 015222 052672  
2688 015224 123245  
2689 015226 026720  
2690 015230 122324  
2691 015232 052672  
2692  
2693  
2694  
2695 015234 012737 015144 001242  
2696 015242 012737 015146 001240  
2697 015250 104001  
2698 015252 000425  
2699  
2700  
2701 015254 012737 015214 001240  
2702 015262 012737 015224 001242  
2703 015270 104001  
2704 015272 000415  
2705  
2706  
2707 015274 012737 010013 001240  
2708 015302 010037 001242  
2709 015306 104001  
2710 015310 000406  
2711  
2712  
2713  
2714 015312 012737 000210 001240  
2715 015320 010537 001242  
2716 015324 104001  
2717  
2718 015326  
(1) 015326 104413  
(1)  
(1)  
(1)  
(1)  
(1)  
2719  
2720  
(3)  
(4)  
(4)
```

```
1$: MOV #4,R3  
CMP (R1)+,(R2)+  
BNE XXB10 ;BRANCH IF INCORRECT.  
SOB R3,1$  
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,@#$TMP4  
MOV #XXB2+4,@#$TMP3  
1$: ERROR 1 ;PC NOT INCREMENTED BY 2.  
BR XXBDONE
```

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

```
;REPORT R0 INCORRECT:  
XXB15: MOV #XXBTP1-5201,@#$TMP3  
MOV R0,@#$TMP4  
1$: ERROR 1 ;SPEC DESTX R0X  
BR XXBDONE
```

```
;REPORT FPS INCORRECT:  
XXB20: MOV #210,@#$TMP3  
MOV R5,@#$TMP4  
1$: ERROR 1
```

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

```
::*****  
;*TEST 43 SPECIAL DEST, MODE 7, TEST  
;*:  
;*THIS IS A TEST OF THE NEGF ABSF AND TSTF DESTINATION FLOWS
```



```
(4) ;*MODE 7 USING THE NEGD INSTR.
(4) ;*
(3) ;*****
(2) 015330 000004 TST43: SCOPE
2721
2722 015332 YYB1:
(1) 015332 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
2723 015334 012701 015464 MOV #YYBTP1,R1 ;SET UP THE DATA BUFFER.
2724 015340 012700 015474 MOV #YYBTP2,R0
2725 015344 012702 000004 MOV #4,R2
2726 015350 012021 1$: MOV (R0)+,(R1)+
2727 015352 077202 SOB R2,1$
2728 015354 012700 010303 MOV #YYBTP3-5201,R0
2729 015360 012760 015464 005201 MOV #YYBTP1,5201(R0)
2730 015366 042737 100000 015464 BIC #100000,@#YYBTP1 ;MAKE THE OPERAND POSITIVE.
2731 015374 012737 015412 001236 MOV #YYB2,@#$TMP2
2732 015402 012701 000200 MOV #200,R1 ;SET FD.
2733 015406 170101 LDFPS R1
2734
2735 015410 005001 YYB2: CLR R1
2736 015412 170770 005201 NEGD @5201(R0) ;TEST INSTRUCTION.
2737
2738 015416 170205 STFPS R5 ;GET FPS.
2739 015420 005701 TST R1 ;WAS THE PC CORRECT AFTER EXECUTION?
2740 015422 001031 BNE YYB25
2741 015424 012701 015464 MOV #YYBTP1,R1 ;IS THE RESULT CORRECT.
2742 015430 012702 015474 MOV #YYBTP2,R2
2743 015434 012703 000004 MOV #4,R3
2744 015440 022122 1$: CMP (R1)+,(R2)+
2745 015442 001031 BNE YYB10 ;BRANCH IF INCORRECT.
2746 015444 077303 SOB R3,1$
2747 015446 022700 010303 CMP #YYBTP3-5201,R0 ;IS R0 CORRECT.
2748 015452 001035 BNE YYB15 ;BRANCH IF INCORRECT.
2749 015454 022705 000210 CMP #210,R5 ;IS THE FPS CORRECT?
2750 015460 001041 BNE YYB20 ;BRANCH IF INCORRECT.
2751 015462 000446 BR YYBDONE
2752
2753 ;THESE ARE DATA TABLES AND A DATA BUFFER.
2754 015464 023245 YYBTP1: 023245
2755 015466 026720 26720
2756 015470 122324 122324
2757 015472 052672 52672
2758 015474 123245 YYBTP2: 123245
2759 015476 026720 26720
2760 015500 123324 123324
2761 015502 052672 52672
2762 015504 015464 YYBTP3: YYBTP1
2763
2764 ;REPORT PC INCORRECT AFTER EXECUTION.
2765 015506 016737 177702 001242 YYB25: MOV YYB2+2,@#$TMP4
2766 015514 016737 177676 001240 MOV YYB2+4,@#$TMP3
2767 015522 104002 1$: ERROR 2 ;PC NOT INCREMENTED BY 2.
2768 015524 000425 BR YYBDONE
2769
2770 ;REPORT RESULT INCORRECT:
2771 015526 012737 015464 001240 YYB10: MOV #YYBTP1,@#$TMP3
```

```

2772 015534 012737 015474 001242      MOV    #YYBTP2,@#$TMP4
2773 015542 104002      1$:   ERROR 2          ;BAD DATA
2774 015544 000415      BR     YYBDONE
2775
2776      ;REPORT R0 INCORRECT:
2777 015546 012737 010303 001240  YYB15: MOV    #YYBTP3-5201,@#$TMP3
2778 015554 010037 001242      MOV    R0,@#$TMP4
2779 015560 104002      1$:   ERROR 2          ;SPEC DESTX ROX
2780 015562 000406      BR     YYBDONE
2781
2782      ;REPORT FPS INCORRECT:
2783 015564 012737 000210 001240  YYB20: MOV    #210,@#$TMP3
2784 015572 010537 001242      MOV    R5,@#$TMP4
2785 015576 104002      1$:   ERROR 2
2786
2787 015600      YYBDONE:
    (1) 015600 104413      RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
    (1)                                     ;SEE IF THE USER HAS EXPRESSED
    (1)                                     ;THE DESIRE TO CHANGE THE SOFTWARE
    (1)                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
    (1)                                     ;THE USER TYPED CONTROL G?).
2793      ;*****
    (3)      ;*TEST 44      NEGD, ABSD AND TSTD TEST
    (4)      ;*
    (4)      ;*THIS IS A TEST OF THE NEGD ABSD AND TSTD INSTRUCTIONS.
    (4)      ;*
    (3)      ;*****
    (2) 015602 000004  TST44: SCOPE
2794      ;TEST NEGD WITH POS NONZERO OPERAND
2795 015604      WWB1:
    (1) 015604 104414      LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
2796 015606 004767 000634      JSR    PC,NATSUB
2797 015612 000000      1$:   0          ;FLAG=NEGD.
2798 015614 016341      2$:   16341      ;OPERAND.
2799 015616 055772      55772
2800 015620 021133      21133
2801 015622 055447      55447
2802 015624 116341      3$:   116341      ;RESULT.
2803 015626 055772      55772
2804 015630 021133      21133
2805 015632 055447      55447
2806 015634 016341      4$:   16341      ;ERROR RES.
2807 015636 055772      55772
2808 015640 021133      21133
2809 015642 055447      55447
2810 015644 000207      5$:   207          ;FPS BEFORE EXECUTION.
2811 015646 000210      210          ;FPS AFTER EXECUTION.
2812 015650 000200      200          ;ERROR FPS.
2813 015652 177777      -1          ;FEC
2814 015654 104002      6$:   ERROR 2      ;E10<---E10*200X ST 336
2815 015656 000401      BR     7$
2816 015660 104002      7$:   ERROR 2      ;BUT ENBT ST 336X WENT TO 053 INTO 453
2817 015662
2818
2819 015662      ;TEST NEGD WITH NEG OPERAND.
    (1) 015662 104414      WWB2:
    LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
    
```



```

2820 015664 004767 000556      JSR      PC,NATSUB
2821 015670 000000      1$:      0          ;FLAG=NEGD.
2822 015672 152525      2$:      152525     ;OPERAND.
2823 015674 053545          53545
2824 015676 055565          55565
2825 015700 057505          57505
2826 015702 052525      3$:      52525     ;RESULT.
2827 015704 053545          53545
2828 015706 055565          55565
2829 015710 057505          57505
2830 015712 152525      4$:      152525     ;ERROR RES.
2831 015714 053545          53545
2832 015716 055565          55565
2833 015720 057505          57505
2834 015722 000217      5$:      217         ;FPS BEFORE EXECUTION.
2835 015724 000200          200         ;FPS AFTER EXECUTION.
2836 015726 000210          210         ;ERROR FPS.
2837 015730 177777          -1          ;FEC
2838 015732 104002      6$:      ERROR      2     ;E10<---E10*200X S336
2839 015734 000401          BR          7$
2840 015736 104002          ERROR      2     ;BUT ENBT X ST336 TO 453 INTO 053
2841 015740
2842
2843 015740      ;TEST ABSD WITH POSITIVE OPERAND
      WMB3:
      LPERR
2844 015742 004767 000500      JSR      PC,NATSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
2845 015746 000001      1$:      1          ;FLAG=ABSD.
2846 015750 060705      2$:      60705     ;OPERAND.
2847 015752 124735          124735
2848 015754 060124          60124
2849 015756 073560          73560
2850 015760 060705      3$:      60705     ;RESULT.
2851 015762 124735          124735
2852 015764 060124          60124
2853 015766 073560          73560
2854 015770 160705      4$:      160705     ;ERROR RES.
2855 015772 124735          124735
2856 015774 060124          60124
2857 015776 073560          73560
2858 016000 000217      5$:      217         ;FPS BEFORE EXECUTION.
2859 016002 000200          200         ;FPS AFTER EXECUTION.
2860 016004 000210          210         ;ERROR FPS.
2861 016006 177777          -1          ;EITHER BUT OP1B
2862 016010 104002      6$:      ERROR      2     ;BUT ST 055 TO 336 INTO 335
2863 016012 000401          BR          7$
2864 016014 104002          ERROR      2     ;OR BUT ENBT ST 335 TO 452 INTO 052
2865 016016
2866
2867 016016      ;TEST ABSD WITH NEG. OPERAND
      WMB4:
      LPERR
      JSR      PC,NATSUB      ;SET UP THE LOOP ON ERROR ADDRESS.
2868 016020 004767 000422      JSR      PC,NATSUB
2869 016024 000001      1$:      1          ;FLAG=ABSD.
2870 016026 154345      2$:      154345     ;OPERAND.
2871 016030 076567          76567
2872 016032 032123          32123
2873 016034 043234          43234
    
```


2928	016206	034510		34510		
2929	016210	045621		45621		
2930	016212	000207		5\$: 207		:FPS BEFORE EXECUTION.
2931	016214	000210		210		:FPS AFTER EXECUTION.
2932	016216	000200		200		:ERROR FPS.
2933	016220	177777		-1		
2934	016222	104002		6\$: ERROR 2		:BUT OPB1 ST 055 TO 335 INTO 334
2935	016224	000401		BR 7\$		
2936	016226	104002		ERROR 2		:BUT ENBT ST 334 TO 053 INTO 453
2937	016230			7\$:		
2938				:TEST TSTD 0 OP		
2939	016230			WWB7:		
(1)	016230	104414		LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
2940	016232	004767	000210	JSR PC,NATSUB		
2941	016236	000002		1\$: 2		:FLAG=TSTD.
2942	016240	000175		2\$: 175		:OPERAND.
2943	016242	176737		176737		
2944	016244	071727		71727		
2945	016246	037574		37574		
2946	016250	000175		3\$: 175		:RESULT.
2947	016252	176737		176737		
2948	016254	071727		71727		
2949	016256	037574		37574		
2950	016260	000000		4\$: 0		:ERROR RES.
2951	016262	000000		0		
2952	016264	000000		0		
2953	016266	000000		0		
2954	016270	000200		5\$: 200		:FPS BEFORE EXECUTION.
2955	016272	000204		204		:FPS AFTER EXECUTION.
2956	016274	000214		214		:ERROR FPS.
2957	016276	177777		-1		
2958	016300	104002		6\$: ERROR 2		:BUT OP1B ST 255 TO 311 OR 312 INTO 310
2959	016302	000401		BR 7\$		
2960	016304	104002		ERROR 2		:BUT ENBT ST 310 TO 402 INTO 002
2961	016306			7\$:		
2962				:TEST TSTD -0 OP FIUV=0		
2963	016306			WWB8:		
(1)	016306	104414		LPERR		:SET UP THE LOOP ON ERROR ADDRESS.
2964	016310	004767	000132	JSR PC,NATSUB		
2965	016314	000002		1\$: 2		:FLAG=TSTD.
2966	016316	100123		2\$: 100123		:OPERAND.
2967	016320	021012		21012		
2968	016322	034565		34565		
2969	016324	043210		43210		
2970	016326	100123		3\$: 100123		:RESULT.
2971	016330	021012		21012		
2972	016332	034565		34565		
2973	016334	043210		43210		
2974	016336	000000		4\$: 0		:ERROR RES.
2975	016340	000000		0		
2976	016342	000000		0		
2977	016344	000000		0		
2978	016346	040203		5\$: 40203		:FPS BEFORE EXECUTION.
2979	016350	040214		040214		:FPS AFTER EXECUTION.
2980	016352	140214		140214		:ERROR FPS.
2981	016354	177777		-1		

2982 016356 104002
 2983 016360 000401
 2984 016362 104002
 2985 016364
 2986
 2987 016364
 (1) 016364 104414
 2988 016366 004767 000054
 2989 016372 000002
 2990 016374 100137
 2991 016376 024613
 2992 016400 057024
 2993 016402 060137
 2994 016404 100137
 2995 016406 024613
 2996 016410 057024
 2997 016412 060137
 2998 016414 000000
 2999 016416 000000
 3000 016420 000000
 3001
 3002 016422 000000
 3003 016424 044200
 3004 016426 144214
 3005 016430 044214
 3006 016432 000014
 3007 016434 104002
 3008 016436 000401
 3009 016440 104002
 3010 016442
 3011 016442 000167 000414
 3012
 3013
 3014
 3015
 3016
 3017
 3018
 3019
 3020
 3021
 3022
 3023
 3024
 3025
 3026
 3027
 3028
 3029
 3030
 3031
 3032
 3033
 3034
 3035
 3036

```

6$: ERROR 2 ;+
   BR 7$
   ERROR 2 ;BUT FIUV ST 257 TO 355 INTO 255
7$:
:TEST TSTD -0 OP FIUV=1
wwb9:
   LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
   JSR PC,NATSUB
1$: 2 ;FLAG=TSTD.
2$: 100137 ;OPERAND.
   24613
   57024
   60137
3$: 100137 ;RESULT.
   24613
   57024
   60137
4$: 0 ;ERROR RES.
   0
   0
5$: 0
   44200 ;FPS BEFORE EXECUTION.
   144214 ;FPS AFTER EXECUTION.
   044214 ;ERROR FPS.
   14
6$: ERROR 2 ;+
   BR 7$
   ERROR 2 ;BUT FIUV ST 257 TO 255 INTO 355
7$:
   JMP wwbdone
  
```

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

```

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

:THE OPERAND IS SET UP IN NATBF1. THEN
 :THE EITHER THE TSTD, NEG, OR ABSD INSTRUCTION IS EXECUTED.
 :NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
 :IS TO BE EXECUTED: 0 = NEG, 1 = ABSD, 2 = TSTD.
 :THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
 :COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL


```

3037 ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
3038 ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
3039 ;TO THE ERROR CALL AT ERR2, OTHERWISE NATSUB ITSELF
3040 ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
3041 ;INSTRUCTION IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
3042 ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
3043 ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN NATSUB
3044 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
3045 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
3046 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
3047
3048
3049 016446 012601 NATSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
3050 016450 010102 MOV R1,R2 ;COPY THE OPERAND.
3051 016452 062702 000002 ADD #2,R2
3052 016456 012703 017050 MOV #NATBF1,R3
3053 016462 012704 000004 MOV #4,R4
3054 016466 012223 1$: MOV (R2)+,(R3)+
3055 016470 077402 SOB R4,1$
3056 016472 016100 000032 MOV 32(R1),R0 ;LOAD THE FPS.
3057 016476 170100 LDFPS R0
3058 016500 012700 017050 MOV #NATBF1,R0 ;SET UP THE OPERAND ADDRESS.
3059 016504 011102 MOV (R1),R2 ;GET THE FLAG TO DETERMINE WHICH
3060 016506 006302 ASL R2 ;INSTRUCTION TO EXECUTE.
3061 016510 006302 ASL R2 ;0 = NEG, 1 = ABSD, 2 = TSTD
3062 016512 012703 016526 MOV #NATINS,R3
3063 016516 060203 ADD R2,R3
3064 016520 010337 001236 MOV R3,@#STMP2
3065 016524 000113 JMP (R3) ;GO EXECUTE THE INSTRUCTION.
3066 016526 170710 NATINS: NEG (R0)
3067 016530 000403 BR 2$
3068 016532 170610 ABSD (R0)
3069 016534 000401 BR 2$
3070 016536 170510 TSTD (R0)
3071
3072 016540 170204 2$: STFPS R4 ;GET THE FPS.
3073 016542 170305 STST R5 ;GET THE FEC.
3074 016544 010102 MOV R1,R2
3075 016546 062702 000002 ADD #2,R2
3076 016552 010237 001240 MOV R2,@#STMP3
3077 016556 062702 000010 ADD #10,R2
3078 016562 010237 001244 MOV R2,@#STMP5
3079 016566 012737 017050 001242 MOV #NATBF1,@#STMP4
3080 016574 010437 001250 MOV R4,@#STMP7
3081 016600 016137 000034 001252 MOV 34(R1),@#STMP10
3082 016606 010100 MOV R1,R0 ;WAS THE RESULT CORRECT?
3083 016610 062700 000012 ADD #12,R0
3084 016614 012702 017050 MOV #NATBF1,R2
3085 016620 012703 000004 MOV #4,R3
3086 016624 022022 3$: CMP (R0)+,(R2)+
3087 016626 001014 BNE 10$ ;BRANCH IF INCORRECT.
3088 016630 077303 SOB R3,3$
3089 016632 026104 000034 CMP 34(R1),R4 ;WAS THE FPS CORRECT?
3090 016636 001032 BNE 15$ ;BRANCH IF INCORRECT.
3091 016640 005761 000034 TST 34(R1) ;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
3092 016644 100003 BPL 4$

```

```
3093 016646 026105 000040      CMP    40(R1),R5      ;WAS THE FEC CORRECT.
3094 016652 001037             BNE    20$           ;BRANCH IF INCORRECT.
3095 016654 000161 000050      4$:   JMP    50(R1)     ;RETURN.
3096
3097                               ;THE RESULT WAS INCORRECT BUT WAS THIS FAILURE ANTICIPATED?
3098                               ;SEE IF THE RESULT WAS ANTICIPATED:
3099 016660             10$:
3100 016660 011105             MOV    (R1),R5
3101 016662 006305             ASL   R5
3102 016664 006305             ASL   R5
3103 016666 062705 017000      ADD   #NATER1,R5
3104 016672 010100             MOV   R1,R0
3105 016674 062700 000022      ADD   #22,R0
3106 016700 012702 017050      MOV   #NATBF1,R2
3107 016704 012703 000004      MOV   #4,R3
3108 016710 022022             11$:   CMP   (R0)+,(R2)+
3109 016712 001003             BNE   12$           ;BRANCH IF NOT ANTICIPATED.
3110 016714 077303             SOB   R3,11$
3111
3112                               ;THE ERROR WAS ANTICIPATED SO RETURN.
3113 016716 000161 000042             JMP   42(R1)
3114
3115                               ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
3116 016722 000115             12$:   JMP   (R5)           ;GO TO THE PROPER ERROR CALL.
3117
3118                               ;THE FPS WAS INCORRECT.
3119 016724 026105 000036             15$:   CMP   36(R1),R5      ;WAS THIS ERROR ANTICIPATED?
3120 016730 001002             BNE   16$           ;BRANCH IF NOT ANTICIPATED.
3121
3122                               ;THE FPS ERROR WAS ANTICIPATED SO RETURN.
3123 016732 000161 000046             JMP   46(R1)
3124
3125                               ;THE FPS FAILURE WAS NOT ANTICIPATED SO REPORT IT HERE.
3126 016736 011102             16$:   MOV   (R1),R2
3127 016740 006302             ASL   R2
3128 016742 006302             ASL   R2
3129 016744 062702 017016      ADD   #NATER2,R2
3130 016750 000112             JMP   (R2)           ;GO TO THE PROPER ERROR CALL.
3131
3132                               ;REPORT THAT THE FEC WAS INCORRECT.
3133 016752 016137 000040 001256      20$:   MOV   40(R1),@#STMP12
3134 016760 010537 001254             MOV   R5,@#STMP11
3135 016764 011102             MOV   (R1),R2
3136 016766 006302             ASL   R2
3137 016770 006302             ASL   R2
3138 016772 062702 017032      ADD   #NATER3,R2
3139 016776 000112             JMP   (R2)           ;GO TO THE PROPER ERROR CALL.
3140
3141                               ;THESE ARE THE ERROR CALLS FOR EACH INDIVIDUAL INSTRUCTION AND CONDITION.
3142 017000 104002      NATER1: ERROR 2 ;NEGD BAD DATA
3143 017002 000403      BR    NATRET
3144 017004 104002      ERROR 2 ;ABSD BAD DATA
3145 017006 000401      BR    NATRET
3146 017010 104002      ERROR 2 ;TSTD BAD DATA
3147 017012 000161 000050      NATRET: JMP 50(R1)
3148
```



```
3149 ;FPS INCORRECT:
3150 017016 104002 NATER2: ERROR 2 ;NEGD FPSX
3151 017020 000774 BR NATRET
3152 017022 104002 ERROR 2 ;ABSD FPSX
3153 017024 000772 BR NATRET
3154 017026 104002 ERROR 2 ;TSTD FPSX
3155 017030 000770 BR NATRET
3156
3157 ;FEC INCORRECT:
3158 017032 104002 NATER3: ERROR 2 ;NEGD FECX
3159 017034 000766 BR NATRET
3160 017036 104002 ERROR 2 ;ABSD FECX
3161 017040 000764 BR NATRET
3162 017042 104002 ERROR 2 ;TSTD FECX
3163 017044 000762 BR NATRET
3164
3165 017046 177777
3166 017050 177777 177777 177777 NATBF1: .WORD -1
017056 177777 177777 .WORD -1,-1,-1,-1,-1
3167
3168 017062
(1) 017062 104413 WwBDONE:
(1) RSETUP
(1) ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
3169
3170
3177
3178 ;*****
(3) ;*TEST 45 SOURCE MODES, MODE 1 (FL=0), TEST
(4) ;*
(4) ;* THIS IS A TEST OF SOURCE MODE 1
(4) ;* USING THE LDFPS INSTR
(4) ;*
(3) ;*****
(2) 017064 000004 TST45: SCOPE
3179
3180
3181 017066
(1) 017066 104414 AAC1:
3182 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3183 017070 012700 017146 MOV #AACTP1,R0 ;SET UP TEST DATA IN BUFFER.
3184 017074 012710 147517 MOV #147517,(R0)
3185 017100 012737 147517 001240 MOV #147517,@#STMP3 ;SAVE DATA IN CASE OF ERROR.
3186 017106 012737 017122 001236 MOV #AAC2,@#STMP2
3187 017114 012737 017206 000004 MOV #AAC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
3188 017122 170110 AAC2: LDFPS (R0) ;TEST INSTRUCTION.
3189
3190 017124 170205 STFPS R5 ;GET FPS
3191
3192 017126 020027 017146 CMP R0,#AACTP1 ;IS R0 CORRECT?
3193 017132 001007 BNE AAC10 ;BR IF NOT.
3194 017134 022705 147517 CMP #147517,R5 ;IS FPS CORRECT?
3195 017140 001013 BNE AAC11 ;BR IF NOT.
3196 017142 000437 BR AACDONE
```

```
3197
3198 ;TEST BUFFER AND DATA:
3199 017144 177777 -1
3200 017146 147517 AAC10: 147517
3201 017150 177777 -1
3202
3203 ;REPORT R0 INCORRECT.
3204 017152 012737 017146 001240 AAC10: MOV #AAC10,@#STMP3
3205 017160 010037 001242 MOV R0,@#STMP4
3206 017164 104001 1$: ERROR 1 ;R0 BAD BUT FSRC FAILED
3207 017166 000425 BR AACDONE
3208
3209 ;REPORT FPS INCORRECT.
3210 017170 012737 147517 001240 AAC11: MOV #147517,@#STMP3 ;REPORT FPS INCORRECT.
3211 017176 010537 001242 MOV R5,@#STMP4
3212 017202 104001 1$: ERROR 1
3213 017204 000416 BR AACDONE
3214
3215 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3216 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3217 ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
3218 017206 AAC20:
3219 017206 011602 MOV (SP),R2
3220 017210 020227 017124 CMP R2,#AAC2+2
3221 017214 001405 BEQ 1$
3222 017216 020227 017126 CMP R2,#AAC2+4
3223 017222 001402 BEQ 1$
3224 017224 000137 036166 JMP @#CPSPUR
3225 017230 022626 1$: CMP (SP)+,(SP)+
3226 017232 010237 001236 MOV R2,@#STMP2
3227 017236 104001 2$: ERROR 1 ;ODD ADRES
3228 017240 000400 BR AACDONE ;BUT FDSTX IN ST 771
3229
3230 AACDONE:
3231 (1) 017242 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3232 (1) ;SEE IF THE USER HAS EXPRESSED
3233 (1) ;THE DESIRE TO CHANGE THE SOFTWARE
3234 (1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3235 (1) ;THE USER TYPED CONTROL G?).
3236
3237 ;*****
3238 ;*TEST 46 SOURCE MODES, MODE 2 (FL=0), TEST
3239 ;*
3240 ;* THIS IS A TEST OF SOURCE MODE 2
3241 ;* USING THE LDFPS INSTR
3242 ;*
3243 ;*****
3244 (3) TST46: SCOPE
3245 (4)
3246 (4)
3247 (4)
3248 (4)
3249 (3)
3250 (2) 017244 000004
3251 017246
3252 (1) 017246 104414 BBC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3253 017250 012700 017326 MOV #BBC1,R0 ;SET UP TEST DATA IN BUFFER.
3254 017254 012710 145212 MOV #145212,(R0)
3255 017260 012737 145212 001240 MOV #145212,@#STMP3 ;SAVE DATA IN CASE OF ERROR.
```



```

CJKddb KEF11-A DIAG PART 2          MACY11 30A(1052) 12-MAR-80 07:25 M 7 PAGE 1-73
CJKddb.P11 12-MAR-80 07:22          T46 SOURCE MODES, MODE 2 (FL=0), TEST                               SEQ 0090

3240 017266 012737 017302 001236      MOV    #BBC2,@#$TMP2
3241 017274 012737 017366 000004      MOV    #BBC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
3242
3243 017302 170120          BBC2:  LDFPS  (R0)+          ;TEST INSTRUCTION.
3244
3245 017304 170205          STFPS  R5                  ;GET FPS
3246
3247 017306 020027 017330          CMP    R0,#BBCTP1+2       ;IS R0 CORRECT?
3248 017312 001007          BNE    BBC10              ;BR IF NOT.
3249 017314 022705 145212          CMP    #145212,R5        ;IS THE FPS CORRECT?
3250 017320 001013          BNE    BBC11              ;BR IF NOT.
3251 017322 000436          BR     BBCDONE
3252
3253
3254          ;TEST BUFFER AND DATA:
3255 017324 177777          -1
3256 017326 177777          BBCTP1: .WORD  -1
3257 017330 177777          -1
3258
3259
3260          ;REPORT R0 INCORRECT.
3261 017332 012737 017330 001240      BBC10: MOV    #BBCTP1+2,@#$TMP3
3262 017340 010037 001242          MOV    R0,@#$TMP4
3263 017344 104001          1$:    ERROR  1          ;R0 BAD BUT FSRC FAILED
3264 017346 000424          BR     BBCDONE
3265
3266          ;REPORT FPS INCORRECT.
3267 017350 012737 145212 001240      BBC11: MOV    #145212,@#$TMP3 ;REPORT FPS INCORRECT.
3268 017356 010537 001242          MOV    R5,@#$TMP4
3269 017362 104001          1$:    ERROR  1
3270 017364 000415          BR     BBCDONE
3271
3272          ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3273          ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3274          ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
3275 017366          BBC20:
3276 017366 011602          MOV    (SP),R2
3277 017370 020227 017304          CMP    R2,#BBC2+2
3278 017374 001405          BEQ    1$
3279
3280 017376 020227 017306          CMP    R2,#BBC2+4
3281 017402 001402          BEQ    1$
3282 017404 000137 036166          JMP    @#CPSPUR
3283 017410 022626          1$:    CMP    (SP)+,(SP)+
3284 017412 010237 001236          MOV    R2,@#$TMP2
3285 017416 104001          2$:    ERROR  1          ;ODD ADRES
3286          ;BUT FDSTX IN ST 771
3287
3288 017420          BBCDONE:
3289 (1) 017420 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
3290 (1)          ;SEE IF THE USER HAS EXPRESSED
(1)          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)          ;THE USER TYPED CONTROL G?).

```

```
3291 (3) *****  
(4) *TEST 47 SOURCE MODES, MODE 4 (FL=0), TEST  
(4) *  
(4) * THIS IS A TEST OF SOURCE MODE 4  
(4) * USING THE LDFPS INSTR  
(3) *****  
(2) 017422 000004 TST47: SCOPE  
3292  
3293 017424 DDC1:  
(1) 017424 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
3294  
3295 017426 012700 017516 MOV #DDCTP1+2,R0 ;SET UP THE TEST DATA BUFFER.  
3296 017432 012760 105252 177776 MOV #105252,-2(R0)  
3297 017440 012737 105252 001240 MOV #105252,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.  
3298 017446 012737 017462 001236 MOV #DDC2,@#$TMP2  
3299 017454 012737 017562 000004 MOV #DDC20,@#ERRVEC  
3300 017462 170140 DDC2: LDFPS -(R0)  
3301 017464 170205 STFPS R5  
3302 017466 020027 017514 CMP R0,#DDCTP1  
3303 017472 001015 BNE DDC10  
3304 017474 022705 105252 CMP #105252,R5  
3305 017500 001021 BNE DDC11  
3306 017502 000444 BR DDCDONE  
3307  
3308 017504 177777 177777 177777 -1,-1,-1,-1  
017512 177777  
3309 017514 177777 DDCTP1: -1  
3310 017516 177777 177777 177777 -1,-1,-1,-1  
017524 177777  
3311  
3312 017526 012737 017514 001240 DDC10: MOV #DDCTP1,@#$TMP3  
3313 017534 010037 001242 MOV R0,@#$TMP4  
3314 017540 104001 1$: ERROR 1 ;R0 BAD BUT FSRC FAILED  
3315 017542 000424 BR DDCDONE  
3316 017544 012737 105252 001240 DDC11: MOV #105252,@#$TMP3 ;REPORT FPS INCORRECT.  
3317 017552 010537 001242 MOV R5,@#$TMP4  
3318 017556 104001 1$: ERROR 1  
3319 017560 000415 BR DDCDONE  
3320 017562 011602 DDC20: MOV (SP),R2  
3321 017564 020227 017464 CMP R2,#DDC2+2  
3322 017570 001405 BEQ 1$  
3323 017572 020227 017466 CMP R2,#DDC2+4  
3324 017576 001402 BEQ 1$  
3325 017600 000137 036166 JMP @#CPSPUR  
3326 017604 022626 1$: CMP (SP)+,(SP)+  
3327 017606 010237 001236 MOV R2,@#$TMP2  
3328 017612 104001 2$: ERROR 1 ;DDD ADRES  
3329 017614 DDCDONE:  
(1) 017614 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
(1) ;SEE IF THE USER HAS EXPRESSED  
(1) ;THE DESIRE TO CHANGE THE SOFTWARE  
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
(1) ;THE USER TYPED CONTROL G?).  
3330 *****  
(3) *TEST 50 SOURCE MODES, MODE 3 (FL=0), TEST
```



```
(4)
(4)
(4)
(4)
(3)
(2) 017616 000004
3331 017620
(1) 017620 104414
3332 017622 012700 017724
3333 017626 012710 017714
3334 017632 012767 103456 000054
3335 017640 012737 103456 001240
3336 017646 012737 017662 001236
3337 017654 012737 017772 000004
3338 017662 170130
3339 017664 170205
3340 017666 020027 017726
3341 017672 001021
3342 017674 022705 103456
3343 017700 001025
3344 017702 000450
3345
3346
3347
3348 017704 177777 177777 177777 ;TEST BUFFER AND DATA:
      017712 177777 -1,-1,-1,-1
3349 017714 177777
3350 017716 177777 177777 177777
3351 017724 017714 177777 177777
      017732 177777 000000
3352
3353
3354 ;REPORT RO INCORRECT.
3355 017736 012737 017726 001240 EEC10: MOV #EECTP2+2,@#STMP3
3356 017744 010037 001242 MOV R0,@#STMP4
3357 017750 104001 1$: ERROR 1 ;RO BAD BUT FSRC FAILED
3358 017752 000424 BR EECDONE
3359
3360 ;REPORT FPS INCORRECT.
3361 017754 012737 103456 001240 EEC11: MOV #103456,@#STMP3 ;REPORT FPS INCORRECT.
3362 017762 010537 001242 MOV R5,@#STMP4
3363 017766 104001 1$: ERROR 1
3364 017770 000415 BR EECDONE
3365 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3366 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3367 ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
3368 017772 011602
3369 017774 020227 017664
3370 020000 001405
3371 020002 020227 017666
3372 020006 001402
3373 020010 000137 036166
3374 020014 022626
3375 020016 010237 001236
3376 020022 104001
3377 020024
```

```
(1) 020024 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
3378 ;*****
(3) ;*TEST 51 SOURCE MODES, MODE 5 (FL=0), TEST
(4) ;*
(4) ;* THIS IS A TEST OF SOURCE MODE 5
(4) ;* USING THE LDFPS INSTR
(4) ;*
(3) ;*****
(2) 020026 000004 TST51: SCOPE
3379 020030 FFC1:
(1) 020030 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3380 020032 012700 020132 MOV #FFCTP2+2,R0 ;SET UP THE TEST DATA BUFFER.
3381 020036 012760 020120 177776 MOV #FFCTP1,-2(R0)
3382 020044 012737 045412 020120 MOV #45412,@#FFCTP1
3383 020052 012737 045412 001240 MOV #45412,@#STMP3 ;SAVE DATA IN CASE OF ERROR.
3384
3385 020060 012737 020030 001236 MOV #FFC1,@#STMP2
3386 020066 012737 020174 000004 FFC2: MOV #FFC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
3387 020074 170150 LDFPS @-(R0) ;TEST INSTRUCTION.
3388 020076 170205 STFPS R5 ;GET THE FPS.
3389 020100 020027 020130 CMP R0,#FFCTP2 ;IS R0 CORRECT?
3390 020104 001015 BNE FFC10 ;BR IF NOT.
3391 020106 022705 045412 CMP #45412,R5 ;IS THE FPS CORRECT?
3392 020112 001021 BNE FFC11 ;BR IF NOT.
3393 020114 000444 BR FFCDONE
3394
3395
3396 ;TEST BUFFER AND DATA:
3397 020116 177777 -1
3398 020120 177777 FFC1: -1
3399 020122 177777 177777 177777 -1,-1,-1
3400 020130 020120 177777 177777 FFC2: FFC1,-1,-1,-1
3401
3402
3403 ;REPORT R0 INCORRECT.
3404 020140 012737 020130 001240 FFC10: MOV #FFCTP2,@#STMP3
3405 020146 010037 001242 MOV R0,@#STMP4
3406 020152 104001 1$: ERROR 1 ;R0 BAD BUT FSRC FAILED
3407 020154 000424 BR FFCDONE
3408
3409 ;REPORT FPS INCORRECT.
3410 020156 012737 045412 001240 FFC11: MOV #45412,@#STMP3 ;REPORT FPS INCORRECT.
3411 020164 010537 001242 MOV R5,@#STMP4
3412 020170 104001 1$: ERROR 1
3413 020172 000415 BR FFCDONE
3414
3415 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3416 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3417 ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
3417 020174 011602 FFC20: MOV (SP),R2
3418 020176 020227 020076 CMP R2,#FFC2+2
3419 020202 001405 BEQ 1$
```



```

3420 020204 020227 020100          CMP     R2,#FFC2+4
3421 020210 001402          BEQ     1$
3422 020212 000137 036166          JMP     @#CPSPUR
3423 020216 022626          1$:    CMP     (SP)+,(SP)+
3424 020220 010237 001236          MOV     R2,@#$TMP2
3425 020224 104001          2$:    ERROR  1          ;ODD ADRES
3426 020226          FFCDONE:
(1) 020226 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)          ;SEE IF THE USER HAS EXPRESSED
(1)          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)          ;THE USER TYPED CONTROL G?).
3427          ;*****
(3)          ;*TEST 52          SOURCE MODES, MODE 6 (FL=0), TEST
(4)          ;*
(4)          ;* THIS IS A TEST OF SOURCE MODE 6
(4)          ;* USING THE LDFPS INSTR
(4)          ;*
(3)          ;*****
(2) 020230 000004          TST52: SCOPE
3428 020232          GGC1:
(1) 020232 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
3429 020234 012700 013123          MOV     #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER.
3430 020240 012737 046543 020324          MOV     #46543,@#GGCTP1
3431 020246 012737 046543 001240          MOV     #46543,@#$TMP3 ;SAVE DATA IN CASE OF ERROR.
3432 020254 012737 020272 001236          MOV     #GGC2,@#$TMP2
3433 020262 005001          CLR     R1
3434 020264 012737 020412 000004          MOV     #GGC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
3435 020272 170160 005201          GGC2: LDFPS  5201(R0)          ;TEST INSTRUCTION.
3436 020276 170204          STFPS  R4          ;GET THE FPS.
3437 020300 005701          TST     R1          ;WAS PC CORRECT AFTER EXECUTION?
3438 020302 001033          BNE     GGC25          ;BR IF NOT.
3439 020304 020027 013123          CMP     R0,#GGCTP1-5201 ;IS R0 CORRECT?
3440 020310 001012          BNE     GGC10          ;BR IF NOT.
3441 020312 022704 046543          CMP     #46543,R4          ;IS THE FPS CORRECT?
3442 020316 001016          BNE     GGC11          ;BR IF NOT.
3443 020320 000451          BR      GGCDONE
3444
3445
3446          ;TEST BUFFER AND DATA:
3447 020322 177777          -1
3448 020324 177777 177777 177777          GGC1: -1,-1,-1,-1
(1) 020332 177777          -1
3449 020334 177777
3450
3451          ;REPORT R0 INCORRECT.
3452 020336 012737 013123 001240          GGC10: MOV     #GGCTP1-5201,@#$TMP3
3453 020344 010037 001242          MOV     R0,@#$TMP4
3454 020350 104001          1$:    ERROR  1          ;R0 BAD BUT FSRC FAILED
3455 020352 000434          BR      GGCDONE
3456
3457          ;REPORT FPS INCORRECT.
3458 020354 012737 046543 001240          GGC11: MOV     #46543,@#$TMP3 ;REPORT FPS INCORRECT.
3459 020362 010437 001242          MOV     R4,@#$TMP4
3460 020366 104001          1$:    ERROR  1
3461 020370 000425          BR      GGCDONE
    
```

```
3462  
3463 ;REPORT PC INCORRECT AFTER INSTRUCTION.  
3464 020372 012737 020276 001240 GGC25: MOV #GGC2+4,@#STMP3  
3465 020400 012737 020274 001242 MOV #GGC2+2,@#STMP4  
3466 020406 104001 1$: ERROR 1 ;PC X  
3467 020410 000415 BR GGC25  
3468 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING  
3469 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT  
3470 ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.  
3471 020412 011602 GGC20: MOV (SP),R2  
3472 020414 020227 020274 CMP R2,#GGC2+2  
3473 020420 001405 BEQ 1$  
3474 020422 020227 020276 CMP R2,#GGC2+4  
3475 020426 001402 BEQ 1$  
3476 020430 000137 036166 JMP @#CPSPUR  
3477 020434 022626 1$: CMP (SP)+,(SP)+  
3478 020436 010237 001236 MOV R2,@#STMP2  
3479 020442 104001 2$: ERROR 1 ;ODD ADRES  
3480 020444 GGC20: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND  
(1) 020444 104413 ;SEE IF THE USER HAS EXPRESSED  
(1) ;THE DESIRE TO CHANGE THE SOFTWARE  
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS  
(1) ;THE USER TYPED CONTROL G?).  
3481 *****  
(3) *TEST 53 SOURCE MODES, MODE 7 (FL=0), TEST  
(4) *  
(4) * THIS IS A TEST OF SOURCE MODE 7  
(4) * USING THE LDFPS INSTR  
(4) *  
(3) *****  
(2) 020446 000004 TST53: SCOPE  
3482 020450 HHC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
(1) 020450 104414 MOV #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.  
3483 020452 012700 013357 MOV #HHCTP1,5201(R0)  
3484 020456 012760 020550 005201 MOV #4547,@#HHCTP1  
3485 020464 012737 004547 020550 MOV #4547,@#STMP3 ;SAVE DATA IN CASE OF ERROR.  
3486 020472 012737 004547 001240 MOV #HHC2,@#STMP2  
3487 020500 012737 020516 001236 CLR R1  
3488 020506 005001 MOV #HHC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.  
3489 020510 012737 020644 000004 HHC2: LDFPS @5201(R0) ;TEST INSTRUCTION.  
3490 020516 170170 005201 STFPS R4 ;GET THE FPS.  
3491 020522 170204 TST R1 ;WAS PC CORRECT AFTER EXECUTION?  
3492 020524 005701 BNE HHC25 ;BR IF NOT.  
3493 020526 001036 CMP R0,#HHCTP2-5201 ;IS R0 CORRECT?  
3494 020530 020027 013357 BNE HHC10 ;BR IF NOT.  
3495 020534 001015 CMP #4547,R4 ;IS THE FPS CORRECT?  
3496 020536 022704 004547 BNE HHC11 ;BR IF NOT.  
3497 020542 001021 BR HHC25  
3498 020544 000454  
3499  
3500  
3501 ;TEST BUFFER AND DATA:  
3502 020546 177777 -1  
3503 020550 177777 177777 177777 HHCTP1: .WORD -1,-1,-1,-1  
020556 177777
```



```

3504 020560 177777 177777 177777 HHCTP2: .WORD -1,-1,-1,-1
      020566 177777
3505
3506 ;REPORT R0 INCORRECT.
3507 020570 012737 013357 001240 HHC10: MOV #HHCTP2-5201,@#$TMP3
3508 020576 010037 001242 MOV R0,@#$TMP4
3509 020602 104001 1$: ERROR 1 ;R0 BAD BUT FSRC FAILED
3510 020604 000434 BR HHC DONE
3511
3512 ;REPORT FPS INCORRECT.
3513 020606 012737 004547 001240 HHC11: MOV #4547,@#$TMP3 ;REPORT FPS INCORRECT.
3514 020614 010437 001242 MOV R4,@#$TMP4
3515 020620 104001 1$: ERROR 1
3516 020622 000425 BR HHC DONE
3517
3518 ;REPORT PC INCORRECT AFTER INSTRUCTION.
3519 020624 012737 020522 001240 HHC25: MOV #HHC2+4,@#$TMP3
3520 020632 012737 020520 001242 MOV #HHC2+2,@#$TMP4
3521 020640 104001 1$: ERROR 1 ;PC X
3522 020642 000415 BR HHC DONE
3523 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3524 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3525 ;FAILURE. OTHERWISE GO TO THE SPURIOUS TRAP TO 4 HANDLING.
3526 020644 011602 HHC20: MOV (SP),R2
3527 020646 020227 020520 CMP R2,#HHC2+2
3528 020652 001405 BEQ 1$
3529 020654 020227 020522 CMP R2,#HHC2+4
3530 020660 001402 BEQ 1$
3531 020662 000137 036166 JMP @#CPSPUR
3532 020666 022626 1$: CMP (SP)+,(SP)+
3533 020670 010237 001236 MOV R2,@#$TMP2
3534 020674 104001 2$: ERROR 1 ;DDD ADDRESS
3535 020676 HHC DONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
      (1) 020676 104413 ;SEE IF THE USER HAS EXPRESSED
      (1) ;THE DESIRE TO CHANGE THE SOFTWARE
      (1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
      (1) ;THE USER TYPED CONTROL G?).
3536
3537
3544
3545 ;*****
      (3) ;*TEST 54 SOURCE MODES, MODE 2 GR7 (FL=1), TEST
      (4) ;*
      (4) ;* THIS IS A TEST OF THE LDCLD WITH
      (4) ;* IMMEDIATE ADDRESSING MODE
      (4) ;*
      (3) ;*****
      (2) 020700 000004 TST54: SCOPE
3546
3547 IIC1:
      (1) 020702 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3548 020704 012737 020730 001236 MOV #IIC2,@#$TMP2 ;SAVE DATA IN CASE OF ERROR.
3549 020712 012737 021002 000004 MOV #IIC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
3550 020720 012700 000300 MOV #300,R0
3551 020724 170100 LDFPS R0
    
```

```

3552 020726 005001 CLR R1
3553
3554 020730 177027 IIC2: LDCLD (R7)+,ACO ;TEST INSTRUCTION.
3555 020732 005201 5201
3556 020734 005201 5201
3557 020736 005201 5201
3558 020740 005201 5201
3559
3560 020742 020127 000003 CMP R1,#3 ;WAS PC CORRECT AFTER EXECUTION?
3561 020746 001421 BEQ IICDONE ;BR IF YES.
3562
3563
3564 ;REPORT PC INCORRECT AFTER INSTRUCTION.
3565 020750 012704 020734 IIC3: MOV #IIC2+4,R4
3566 020754 162701 000003 SUB #3,R1
3567 020760 006301 ASL R1
3568 020762 160104 SUB R1,R4
3569 020764 010437 001242 MOV R4,@#$TMP4
3570 020770 012737 020734 001240 MOV #IIC2+4,@#$TMP3
3571 020776 104001 1$: ERROR 1 ;BAD CONSTANT
3572 021000 000404 BR IICDONE
3573
3574 ;TRAP HERE THROUGH VECTOR FOUR. SEE IF THE TRAP WAS DURING
3575 ;EXECUTION OF THE FPS INSTRUCTION BEING TESTED. IF SO REPORT
3576 021002 011637 001236 IIC20: MOV (SP),@#$TMP2
3577 021006 022626 CMP (SP)+,(SP)+
3578 021010 104001 1$: ERROR 1 ;BAD CONSTANT ODD ADD
3579
3580 021012 IICDONE:
(1) 021012 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
3581
3588
3589
(3) ;*****
(4) ;*TEST 55 SOURCE MODES, MODE 2 (FL=1), TEST
(4) ;*
(4) ;* THIS IS A TEST OF THE LDCLD INSTR
(4) ;* WITH MODE 2.
(4) ;*
(3) ;*****
(2) 021014 000004 TST55: SCOPE
3590
3591 021016 TCC1:
(1) 021016 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3592 021020 016737 000014 001236 MOV TCC2,@#$TMP2 ;SAVE DATA IN CASE OF ERROR.
3593 021026 012700 000300 MOV #300,R0
3594 021032 170100 LDFPS R0
3595 021034 012700 021130 MOV #TCCBF0,R0 ;SET UP THE TEST DATA BUFFER.
3596 021040 177020 TCC2: LDCLD (R0)+,ACO ;TEST INSTRUCTION.
3597
3598 021042 170204 STFPS R4 ;GET THE FPS.
3599 021044 012701 021140 MOV #TCCBF1,R1 ;GET THE RESULT.
3600 021050 012702 000200 MOV #200,R2
    
```



```
3601 021054 170102          LDFPS R2
3602 021056 174011          STD   ACO,(R1)
3603 021060 020027 021134    CMP   R0,#TCCBF0+4 ;IS R0 CORRECT?
3604 021064 001407          BEQ   TCC3
3605          ;REPORT R0 INCORRECT.
3606 021066 010037 001242    MOV   R0,@#STMP4
3607 021072 012737 021134 001240  MOV   #TCCBF0+4,@#STMP3
3608 021100 104001          1$:  ERROR 1 ;BAD CONST
3609 021102 000422          BR    TCCDONE
3610
3611 021104 022704 000300    TCC3: CMP   #300,R4 ;IS THE FPS CORRECT?
3612 021110 001417          BEQ   TCCDONE
3613
3614          ;REPORT FPS INCORRECT.
3615 021112 010437 001242    MOV   R4,@#STMP4
3616 021116 012737 000300 001240  MOV   #300,@#STMP3
3617 021124 104001          1$:  ERROR 1 ;FPS X
3618 021126 000410          BR    TCCDONE
3619
3620          ;TEST BUFFER AND DATA:
3621          TCCBF0: .WORD 01234,67076,54321,012345
3622 021130 001234 067076 054321
3623 021136 012345
3624 021140 177777 177777 177777 TCCBF1: -1,-1,-1,-1
3625 021146 177777
3626
3627          TCCDONE:
3628 (1) 021150 104413          RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
3629 (1)          ;SEE IF THE USER HAS EXPRESSED
3630 (1)          ;THE DESIRE TO CHANGE THE SOFTWARE
3631 (1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
3632 (1)          ;THE USER TYPED CONTROL G?).
3633
3634
3635          ;*****
3636 (3)          ;*TEST 56 LDCIF AND LDCLF TEST
3637 (4)          ;*
3638 (4)          ;* THIS IS A TEST OF THE LDCIF AND
3639 (4)          ;* THE LDCLF INSTRUCTIONS.
3640 (4)          ;*
3641 (3)          ;*****
3642 (2) 021152 000004          TST56: SCOPE
3643
3644          ;ZERO OPERAND FL=0
3645
3646          KKC1:
3647 (1) 021154 104414          LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
3648 021156 004737 022306    JSR   PC,@#LDCFSUB ;GO EXECUTE INSTRUCTION.
3649
3650 021162 000000 000000    1$:  .WORD 0,0 ;FSRC OPERAND.
3651 021166 000000 000000    2$:  .WORD 0,0 ;EXPECTED RESULT.
3652 021172 177777 177777    3$:  .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
3653 021176 000000          4$:  0 ;FPS BEFORE EXECUTION.
3654 021200 000004          4 ;FPS AFTER EXECUTION.
```

3648	021202	177777			-1				:ANTICIPATED ERRONEOUS FPS.
3649	021204	104001		5\$:	ERROR	1			:REPORT RESULT INCORRECT.
3650	021206	000401			BR	6\$			
3651	021210	104001			ERROR	1			:REPORT FPS INCORRECT.
3652	021212			6\$:					
3653				:ZERO	OPERAND	FL=0			
3654									
3655	021212			KKC2:					
(1)	021212	104414			LPERR				:SET UP THE LOOP ON ERROR ADDRESS.
3656	021214	004737	022306		JSR	PC,@#LDCFSUB			:GO EXECUTE THE INSTRUCTION.
3657									
3658	021220	000000	177777	1\$:	.WORD	0,-1			:FSRC OPERAND.
3659	021224	000000	000000	2\$:	.WORD	0,0			:EXPECTED RESULT.
3660	021230	004177	177400	3\$:	4177,177400				:ANTICIPATED ERRONEOUS RESULT.
3661	021234	000000		4\$:	0				:FPS BEFORE EXECUTION.
3662	021236	000004			4				:FPS AFTER EXECUTION.
3663	021240	177777			-1				:ANTICIPATED ERRONEOUS FPS.
3664	021242	104001		5\$:	ERROR	1			:(BUT FL) ST
3665	021244	000401			BR	6\$:277 TO 300
3666	021246	104001			ERROR	1			:INTO 301
3667	021250			6\$:					
3668				:ZERO	OPERAND	FL=1			
3669									
3670	021250			KKC3:					
(1)	021250	104414			LPERR				:SET UP THE LOOP ON ERROR ADDRESS.
3671	021252	004737	022306		JSR	PC,@#LDCFSUB			:GO EXECUTE THE INSTRUCTION.
3672									
3673	021256	000000	000000	1\$:	.WORD	0,0			:FSRC OPERAND.
3674	021262	000000	000000	2\$:	.WORD	0,0			:EXPECTED RESULT.
3675	021266	177777	177777	3\$:	.WORD	-1,-1			:ANTICIPATED ERRONEOUS RESULT.
3676	021272	000100		4\$:	100				:FPS BEFORE EXECUTION.
3677	021274	000104			104				:FPS AFTER EXECUTION.
3678	021276	000004			4				:ANTICIPATED ERRONEOUS FPS.
3679	021300	104001		5\$:	ERROR	1			:REPORT RESULT INCORRECT.
3680	021302	000401			BR	6\$			
3681	021304	104001			ERROR	1			:FL WAS CLR'ED
3682	021306			6\$:					
3683				:OPERAND	POSITIVE	FL=0			
3684	021306			KKC4:					
(1)	021306	104414			LPERR				:SET UP THE LOOP ON ERROR ADDRESS.
3685	021310	004737	022306		JSR	PC,@#LDCFSUB			:GO EXECUTE THE INSTRUCTION.
3686	021314	040000	000000	1\$:	.WORD	4000,0			:FSRC OPERAND.
3687	021320	043600	000000	2\$:	.WORD	43600,0			:EXPECTED RESULT.
3688	021324	047600	000000	3\$:	.WORD	47600,0			:ANTICIPATED ERRONEOUS RESULT.
3689	021330	000017		4\$:	17				:FPS BEFORE EXECUTION.
3690	021332	000000			0				:FPS AFTER EXECUTION.
3691	021334	177777			-1				:ANTICIPATED ERRONEOUS FPS.
3692	021336	104001		5\$:	ERROR	1			:ST 107 BAD
3693	021340	000401			BR	6\$:CONSTANT 231 INSD
3694	021342	104001			ERROR	1			:215
3695	021344			6\$:					
3696				:OPERAND=1,	FL=0				
3697	021344			KKC5:					
(1)	021344	104414			LPERR				:SET UP THE LOOP ON ERROR ADDRESS.
3698	021346	004737	022306		JSR	PC,@#LDCFSUB			:GO EXECUTE THE INSTRUCTION.
3699	021352	000001	000000	1\$:	.WORD	1,0			:FSRC OPERAND.

3700 021356 040200 000000
3701 021362 044200 000000
3702 021366 000017
3703 021370 000000
3704 021372 177777
3705 021374 104001
3706 021376 000401
3707 021400 104001
3708 021402
3709
3710
3711
3712 021402
(1) 021402 104414
3713 021404 004737 022306
3714 021410 000252 000000
3715 021414 042052 000000
3716 021420 046052 000000
3717 021424 000000
3718 021426 000000
3719 021430 177777
3720 021432 104000
3721 021434 000401
3722 021436 104001
3723 021440
3724
3725
3726 021440
(1) 021440 104414
3727 021442 004737 022306
3728 021446 140000 000000
3729 021452 143600 000000
3730 021456 043600 000000
3731 021462 000007
3732 021464 000010
3733 021466 177777
3734 021470 104001
3735 021472 000401
3736 021474 104001
3737 021476
3738
3739
3740 021476
(1) 021476 104414
3741 021500 004737 022306
3742 021504 177777 000000
3743 021510 140200 000000
3744 021514 144000 000400
3745 021520 000000
3746 021522 000010
3747 021524 177777
3748 021526 104001
3749 021530 000401
3750 021532 104001
3751 021534
3752

2\$: .WORD 40200,0 ; EXPECTED RESULT.
3\$: .WORD 44200,0 ; ANTICIPATED ERRONEOUS RESULT.
4\$: 17 ; FPS BEFORE EXECUTION.
0 ; FPS AFTER EXECUTION.
-1 ; ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 ; REPORT RESULT INCORRECT.
BR 6\$
ERROR 1 ; REPORT FPS INCORRECT.
6\$:
; OPERAND= PATTERN FL=0
KKC6:
LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCFSUB ; GO EXECUTE THE INSTRUCTION.
1\$: .WORD 252,0 ; FSRC OPERAND.
2\$: .WORD 42052,0 ; EXPECTED RESULT.
3\$: .WORD 46052,0 ; ANTICIPATED ERRONEOUS RESULT.
4\$: 0 ; FPS BEFORE EXECUTION.
0 ; FPS AFTER EXECUTION.
-1 ; ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 ; REPORT RESULT INCORRECT.
BR 6\$
ERROR 1 ; REPORT FPS INCORRECT.
6\$:
; OPERAND=-40000 FL=0
KKC7:
LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCFSUB ; GO EXECUTE THE INSTRUCTION.
1\$: .WORD -40000,0 ; FSRC OPERAND.
2\$: .WORD 143600,0 ; EXPECTED RESULT.
3\$: .WORD 43600,0 ; ANTICIPATED ERRONEOUS RESULT.
4\$: 7 ; FPS BEFORE EXECUTION.
10 ; FPS AFTER EXECUTION.
-1 ; ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 ; (SET SIGN) ST 146
BR 6\$
ERROR 1 ; REPORT FPS INCORRECT.
6\$:
; OPERAND=-1 FL=0
KKC8:
LPERR ; SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCFSUB ; GO EXECUTE THE INSTRUCTION.
1\$: .WORD -1,0 ; FSRC OPERAND.
2\$: .WORD 140200,0 ; EXPECTED RESULT.
3\$: .WORD 144000,400 ; ANTICIPATED ERRONEOUS RESULT.
4\$: 0 ; FPS BEFORE EXECUTION.
10 ; FPS AFTER EXECUTION.
-1 ; ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 ; ST 372 TO 152 INTO
BR 6\$; 112 (BUF XNBT)
ERROR 1 ; REPORT FPS INCORRECT.
6\$:

```
3753 ;OPERAND=PATTERN FL=0
3754 021534 KKC9: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 021534 104414 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
3755 021536 004737 022306 1$: .WORD 125252,0 ;FSRC OPERAND.
3756 021542 125252 000000 2$: .WORD 143652,126000 ;EXPECTED RESULT.
3757 021546 143652 126000 3$: .WORD 43652,126000 ;ANTICIPATED ERRONEOUS RESULT.
3758 021552 043652 126000 4$: 7 ;FPS BEFORE EXECUTION.
3759 021556 000007 10 ;FPS AFTER EXECUTION.
3760 021560 000010 -1 ;ANTICIPATED ERRONEOUS FPS.
3761 021562 177777 5$: ERROR 1 ;REPORT RESULT INCORRECT.
3762 021564 104001 BR 6$
3763 021566 000401 ERROR 1 ;REPORT FPS INCORRECT.
3764 021570 104001 6$:
3765 021572
3766
3767 ;OPERAND POS FL-1
3768 021572 KKC10: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 021572 104414 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
3769 021574 004737 022306 1$: .WORD 40000,0 ;FSRC OPERAND.
3770 021600 040000 000000 2$: .WORD 47600,0 ;EXPECTED RESULT.
3771 021604 047600 000000 3$: .WORD 43600,0 ;ANTICIPATED ERRONEOUS RESULT.
3772 021610 043600 000000 4$: 117 ;FPS BEFORE EXECUTION.
3773 021614 000117 100 ;FPS AFTER EXECUTION.
3774 021616 000100 -1 ;ANTICIPATED ERRONEOUS FPS.
3775 021620 177777 5$: ERROR 1 ;ST 107 CONSTANT
3776 021622 104001 BR 6$ ;BAD 237 INST 217
3777 021624 000401 ERROR 1 ;REPORT FPS INCORRECT.
3778 021626 104001 6$:
3779 021630
3780
3781 ;OPERAND=1 FL=1
3782 021630 KKC11: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 021630 104414 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
3783 021632 004737 022306 1$: .WORD 0,1 ;FSRC OPERAND.
3784 021636 000000 000001 2$: .WORD 40200,0 ;EXPECTED RESULT.
3785 021642 040200 000000 3$: .WORD 34200,0 ;ANTICIPATED ERRONEOUS RESULT.
3786 021646 034200 000000 4$: 100 ;FPS BEFORE EXECUTION.
3787 021652 000100 100 ;FPS AFTER EXECUTION.
3788 021654 000100 -1 ;ANTICIPATED ERRONEOUS FPS.
3789 021656 177777 5$: ERROR 1 ;REPORT RESULT INCORRECT.
3790 021660 104001 BR 6$
3791 021662 000401 ERROR 1 ;REPORT FPS INCORRECT.
3792 021664 104001 6$:
3793 021666
3794
3795 ;OPERAND= PATTERN FL=1
3796 021666 KKC12: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 021666 104414 JSR PC,@#LDCFSUB ;GO EXECUTE THE INSTRUCTION.
3797 021670 004737 022306 1$: .WORD 0,252 ;FSRC OPERAND.
3798 021674 000000 000252 2$: .WORD 42052,0 ;EXPECTED RESULT.
3799 021700 042052 000000 3$: .WORD 36052,0 ;ANTICIPATED ERRONEOUS RESULT.
3800 021704 036052 000000 4$: 111 ;FPS BEFORE EXECUTION.
3801 021710 000111 100 ;FPS AFTER EXECUTION.
3802 021712 000100 -1 ;ANTICIPATED ERRONEOUS FPS.
3803 021714 177777 5$: ERROR 1 ;REPORT RESULT INCORRECT.
3804 021716 104001
```


3805	021720	000401		BR	6\$		
3806	021722	104001		ERROR	1		;REPORT FPS INCORRECT.
3807	021724			6\$:			
3808							
3809				:OPERAND=-40000,0	FL=1		
3810	021724			KKC13:			
(1)	021724	104414		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
3811	021726	004737	022306	JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
3812	021732	140000	000000	1\$:	.WORD -40000,0		;FSRC OPERAND.
3813	021736	147600	000000	2\$:	.WORD 147600,0		;EXPECTED RESULT.
3814	021742	047600	000000	3\$:	.WORD 47600,0		;ANTICIPATED ERRONEOUS RESULT.
3815	021746	000107		4\$:	107		;FPS BEFORE EXECUTION.
3816	021750	000110			110		;FPS AFTER EXECUTION.
3817	021752	177777			-1		;ANTICIPATED ERRONEOUS FPS.
3818	021754	104001		5\$:	ERROR 1		;SET SIGN
3819	021756	000401		BR	6\$		
3820	021760	104001		ERROR	1		;REPORT FPS INCORRECT.
3821	021762			6\$:			
3822							
3823				:OPERAND=-1,-1	FL=1		
3824	021762			KKC14:			
(1)	021762	104414		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
3825	021764	004737	022306	JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
3826	021770	177777	177777	1\$:	.WORD -1,-1		;FSRC OPERAND.
3827	021774	140200	000000	2\$:	.WORD 140200,0		;EXPECTED RESULT.
3828	022000	150000	000000	3\$:	.WORD 150000,0		;ANTICIPATED ERRONEOUS RESULT.
3829	022004	000100		4\$:	100		;FPS BEFORE EXECUTION.
3830	022006	000110			110		;FPS AFTER EXECUTION.
3831	022010	177777			-1		;ANTICIPATED ERRONEOUS FPS.
3832	022012	104001		5\$:	ERROR 1		; (BUT XNBT)
3833	022014	000401		BR	6\$		
3834	022016	104001		ERROR	1		;REPORT FPS INCORRECT.
3835	022020			6\$:			
3836							
3837				:OPERAND=-PATTERN	FL=1, ROUND MODE		
3838	022020			KKC15:			
(1)	022020	104414		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
3839	022022	004737	022306	JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
3840	022026	125252	125252	1\$:	.WORD 125252,125252		;FSRC OPERAND.
3841	022032	147652	125253	2\$:	.WORD 147652,125253		;EXPECTED RESULT.
3842	022036	047652	125253	3\$:	.WORD 47652,125253		;ANTICIPATED ERRONEOUS RESULT.
3843	022042	000105		4\$:	105		;FPS BEFORE EXECUTION.
3844	022044	000110			110		;FPS AFTER EXECUTION.
3845	022046	177777			-1		;ANTICIPATED ERRONEOUS FPS.
3846	022050	104001		5\$:	ERROR 1		;REPORT RESULT INCORRECT.
3847	022052	000401		BR	6\$		
3848	022054	104001		ERROR	1		;REPORT FPS INCORRECT.
3849	022056			6\$:			
3850							
3851				:OPERAND=77777,177500	FL=1, ROUND MODE		
3852	022056			KKC16:			
(1)	022056	104414		LPERR			;SET UP THE LOOP ON ERROR ADDRESS.
3853	022060	004737	022306	JSR	PC,@#LDCFSUB		;GO EXECUTE THE INSTRUCTION.
3854	022064	077777	177500	1\$:	.WORD 77777,177500		;FSRC OPERAND.
3855	022070	047777	177777	2\$:	.WORD 47777,177777		;EXPECTED RESULT.
3856	022074	047777	177776	3\$:	.WORD 47777,177776		;ANTICIPATED ERRONEOUS RESULT.

3909 022254 100000 000000
 3910 022260 150000 000000
 3911 022264 147600 000000
 3912 022270 000107
 3913 022272 000110
 3914 022274 177777
 3915 022276 104001
 3916 022300 000401
 3917 022302 104001
 3918 022304 000506
 3919
 3920
 3921
 3922
 3923
 3924
 3925
 3926
 3927
 3928
 3929
 3930
 3931
 3932
 3933
 3934
 3935
 3936
 3937
 3938
 3939
 3940
 3941
 3942
 3943
 3944
 3945
 3946
 3947
 3948
 3949
 3950

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 1 ;REPORT RESULT INCORRECT.
 BR 6\$
 ERROR 1 ;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 1;DATA ERROR
    BR CONT
    ERR2: ERROR 1;FPS ERROR
    CONT: ;RETURN ADDRESS
    
```

;THE OPERANDS ARE SET UP (USING ACO 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
 ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
 ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
 ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.

3951 022306 012601
 3952 022310 016100 000014
 3953 022314 170100
 3954 022316 012737 022326 001236
 3955 022324 010100
 3956
 3957 022326 177010
 3958
 3959 022330 170204
 3960 022332 012700 022512
 3961 022336 012702 000200
 3962 022342 170102
 3963 022344 174010
 3964

```

LDCFSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
        MOV 14(R1),R0 ;SET THE FPS.
        LDFPS R0
        MOV #1$,@#TMP2
        MOV R1,R0
1$: LDCIF (R0),ACO ;TEST INSTRUCTION LDCIF OR LDCLF.
        STFPS R4 ;GET FPS.
        MOV #LDCT,R0 ;GET THE RESULT.
        MOV #200,R2
        LDFPS R2
        STD ACO,(R0)
    
```

```

3965 022346 012702 022512      MOV      #LDCT,R2      ;SEE IF THE RESULT WAS CORRECT.
3966 022352 010237 001242      MOV      R2,@#STMP4
3967 022356 010137 001240      MOV      R1,@#STMP3
3968 022362 010103              MOV      R1,R3
3969 022364 062703 000004      ADD      #4,R3
3970 022370 010337 001244      MOV      R3,@#STMP5
3971 022374 010437 001250      MOV      R4,@#STMP7
3972 022400 016137 000016 001252  MOV      16(R1),@#STMP10
3973 022406 010100              MOV      R1,R0
3974 022410 062700 000004      ADD      #4,R0
3975 022414 012703 000002      MOV      #2,R3
3976 022420 022022      2$:      CMP      (R0)+,(R2)+
3977 022422 001006              BNE     10$           ;BR IF INCORRECT.
3978 022424 077303              SOB     R3,2$
3979
3980 022426 026104 000016      CMP      16(R1),R4     ;SEE IF THE FPS WAS CORRECT.
3981 022432 001020              BNE     15$           ;BR IF INCORRECT.
3982 022434 000161 000030      3$:      JMP      30(R1)       ;RETURN.
3983
3984      ;RESULT IN CORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
3985 022440 012702 022512      10$:     MOV      #LDCT,R2
3986 022444 010100              MOV      R1,R0
3987 022446 062700 000010      ADD      #10,R0
3988 022452 012703 000002      MOV      #2,R3
3989 022456 022022      11$:     CMP      (R0)+,(R2)+
3990 022460 001003              BNE     13$
3991 022462 077303              SOB     R3,11$
3992 022464 000161 000022      JMP      22(R1)
3993
3994      ;THE FAILURE WAS NOT ANTICIPATED SO REPORT THE ERROR HERE.
3995 022470      13$:
3996
3997 022470 104001      14$:     ERROR  1           ;BAD RES
3998 022472 000760              BR      3$
3999
4000
4001
4002      ;THE FPS WAS INCORRECT SO SEE IF IT WAS ANTICIPATED.
4003 022474 026104 000020      15$:     CMP      20(R1),R4
4004 022500 001002              BNE     16$
4005 022502 000161 000026      JMP      26(R1)
4006
4007      ;FPS ERROR NOT ANTICIPATED SO REPORT IT HERE.
4008 022506      16$:
4009 022506 104001      17$:     ERROR  1           ;BAD FPS
4010 022510 000751              BR      3$
4011
4012      ;DATA BUFFER:
4013 022512 000000 000000 000000  LDCT:   .WORD  0,0,0,0
4014 022520 000000
4015 022522      KKCDONE:
(1) 022522 104413              RSETUP
(1)
(1)
(1)
    ;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?).

```
(1)
4016
4022
4023
(3)
(4)
(4)
(4)
(3)
(2) 022524 000004
4024
4025 022526
(1) 022526 104414
4026 022530 004737 023324
4027 022534 000000 000000
4028 022540 000000 000000 000000
      022546 000000
4029 022550 177777 177777 177777
      022556 177777
4030 022560 000213
4031 022562 000204
4032 022564 177777
4033 022566 104001
4034 022570 000401
4035 022572 104001
4036 022574
4037
4038 022574
(1) 022574 104414
4039 022576 004737 023324
4040 022602 000000 177777
4041 022606 000000 000000 000000
      022614 000000
4042 022616 004177 177400 000000
      022624 000000
4043 022626 000200
4044 022630 000204
4045 022632 177777
4046 022634 104001
4047 022636 000401
4048 022640 104001
4049 022642
4050
4051
4052 022642
(1) 022642 104414
4053 022644 004737 023324
4054 022650 000000 000000
4055 022654 000000 000000 000000
      022662 000000
4056 022664 177777 177777 177777
      022672 177777
4057 022674 000211
4058 022676 000204
4059 022700 177777
4060 022702 104001
```

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

TST57: SCOPE
:OPERAND=0 FL=0, FD=1
LLC1:
LPERR :SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCDSUB :GO EXECUTE THE INSTRUCTION.
1\$: .WORD 0,0 :FSRC OPERAND.
2\$: .WORD 0,0,0,0 :EXPECTED RESULT.
3\$: .WORD -1,-1,-1,-1 :ANTICIPATED ERRONEOUS RESULT.
4\$: 213 :FPS BEFORE EXECUTION.
204 :FPS AFTER EXECUTION.
-1 :ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 :REPORT RESULT INCORRECT.
BR 6\$
ERROR 1 :REPORT FPS INCORRECT.
6\$:
:OPERAND=0 FL=0, FD=1
LLC2:
LPERR :SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCDSUB :GO EXECUTE THE INSTRUCTION.
1\$: .WORD 0,-1 :FSRC OPERAND.
2\$: .WORD 0,0,0,0 :EXPECTED RESULT.
3\$: .WORD 4177,177400,0,0 :ANTICIPATED ERRONEOUS RESULT.
4\$: 200 :FPS BEFORE EXECUTION.
204 :FPS AFTER EXECUTION.
-1 :ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 : (BUT FL)S+277
BR 6\$:TO 300 INTO 301
ERROR 1 :REPORT FPS INCORRECT.
6\$:
:OPERAND=0 FL=1 FD=1
LLC3:
LPERR :SET UP THE LOOP ON ERROR ADDRESS.
JSR PC,@#LDCDSUB :GO EXECUTE THE INSTRUCTION.
1\$: .WORD 0,0 :FSRC OPERAND.
2\$: .WORD 0,0,0,0 :EXPECTED RESULT.
3\$: .WORD -1,-1,-1,-1 :ANTICIPATED ERRONEOUS RESULT.
4\$: 211 :FPS BEFORE EXECUTION.
204 :FPS AFTER EXECUTION.
-1 :ANTICIPATED ERRONEOUS FPS.
5\$: ERROR 1 :REPORT RESULT INCORRECT.

```

4061 022704 000401          BR      6$
4062 022706 104001          ERROR   1          ;REPORT FPS INCORRECT.
4063 022710          6$:
4064
4065          ;OPERAND=40000 FL=0   FD=1
4066 022710          LLC4:
   (1) 022710 104414          LPERR
4067 022712 004737 023324          JSR     PC,@#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
4068 022716 040000 000000          .WORD  40000,0      ;GO EXECUTE THE INSTRUCTION.
4069 022722 043600 000000 000000 1$:      .WORD  43600,0,0,0 ;FSRC OPERAND.
   022730 000000          2$:      .WORD  43600,0,0,0 ;EXPECTED RESULT.
4070 022732 047600 000000 000000 3$:      .WORD  47600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
   022740 000000          4$:      217
4071 022742 000217          200
4072 022744 000200          -1
4073 022746 177777          5$:      ERROR   1          ;FPS BEFORE EXECUTION.
4074 022750 104001          BR      6$          ;FPS AFTER EXECUTION.
4075 022752 000401          ERROR   1          ;ANTICIPATED ERRONEOUS FPS.
4076 022754 104001          BR      6$          ;ST 107 BAD CONST
4077 022756          ERROR   1          ;REPORT FPS INCORRECT.
4078
4079          6$:
4080          ;OPERAND=-40000 FL=0   FD=1
4081 022756          LLC5:
   (1) 022756 104414          LPERR
4082 022760 004737 023324          JSR     PC,@#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
4083 022764 140000 000000          .WORD  -40000,0     ;GO EXECUTE THE INSTRUCTION.
4084 022770 143600 000000 000000 1$:      .WORD  143600,0,0,0 ;FSRC OPERAND.
   022776 000000          2$:      .WORD  143600,0,0,0 ;EXPECTED RESULT.
4085 023000 043600 000000 000000 3$:      .WORD  43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
   023006 000000          4$:      200
4086 023010 000200          210
4087 023012 000210          -1
4088 023014 177777          5$:      ERROR   1          ;FPS BEFORE EXECUTION.
4089 023016 104001          BR      6$          ;FPS AFTER EXECUTION.
4090 023020 000401          ERROR   1          ;ANTICIPATED ERRONEOUS FPS.
4091 023022 104001          BR      6$          ;(SET SIGN) ST 176
4092 023024          ERROR   1          ;REPORT FPS INCORRECT.
4093
4094          6$:
4095          ;OPERAND=40000,0      FL=1   FD=1
4096 023024          LLC6:
   (1) 023024 104414          LPERR
4097 023026 004737 023324          JSR     PC,@#LDCDSUB ;SET UP THE LOOP ON ERROR ADDRESS.
4098 023032 040000 000000          .WORD  40000,0     ;GO EXECUTE THE INSTRUCTION.
4099 023036 047600 000000 000000 1$:      .WORD  47600,0,0,0 ;FSRC OPERAND.
   023044 000000          2$:      .WORD  47600,0,0,0 ;EXPECTED RESULT.
4100 023046 043600 000000 000000 3$:      .WORD  43600,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
   023054 000000          4$:      317
4101 023056 000317          300
4102 023060 000300          -1
4103 023062 177777          5$:      ERROR   1          ;FPS BEFORE EXECUTION.
4104 023064 104001          BR      6$          ;FPS AFTER EXECUTION.
4105 023066 000401          ERROR   274        ;ANTICIPATED ERRONEOUS FPS.
4106 023070 104274          BR      6$          ;ST 107 BAD CONS
4107 023072          ERROR   274        ;REPORT FPS INCORRECT.
          6$:
          ;OPERAND=0,1      FL=1   FD=1

```



```

4108 023072          LLC7:
(1) 023072 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4109 023074 004737 023324 JSR          PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
4110 023100 000000 000001 1$: .WORD 0,1 ;FSRC OPERAND.
4111 023104 040200 000000 000000 2$: .WORD 40200,0,0,0 ;EXPECTED RESULT.
4112 023112 000000          3$: .WORD 34200,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
4113 023122 000000          4$: 300 ;FPS BEFORE EXECUTION.
4114 023124 000300          300 ;FPS AFTER EXECUTION.
4115 023126 000300          -1 ;ANTICIPATED ERRONEOUS FPS.
4116 023130 177777          5$: ERROR 1 ;REPORT FPS INCORRECT.
4117 023132 104001          BR 6$
4118 023134 000401          ERROR 1 ;REPORT FPS INCORRECT.
4119 023136 104001          6$:
4120 023140          ;OPERAND=77777,177777 FL=1 FD=1
4121 023142          LLC8:
(1) 023140 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4123 023142 004737 023324 JSR          PC,@#LDCDSUB ;GO EXECUTE THE INSTRUCTION.
4124 023146 077777 177777 1$: .WORD 77777,177777 ;FSRC OPERAND.
4125 023152 047777 177777 177000 2$: .WORD 47777,177777,177000,0 ;EXPECTED RESULT.
4126 023160 000000          3$: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
4127 023170 177777          4$: 317 ;FPS BEFORE EXECUTION.
4128 023172 000317          300 ;FPS AFTER EXECUTION.
4129 023174 000300          -1 ;ANTICIPATED ERRONEOUS FPS.
4130 023176 177777          5$: ERROR 1 ;REPORT RESULT INCORRECT.
4131 023200 104001          BR 6$
4132 023202 000401          ERROR 1 ;REPORT FPS INCORRECT.
4133 023204 104001          6$:
4134 023206          ;OPERAND=-PATTERN FL=1 FD=1
4135 023206          LLC9:
(1) 023206 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4138 023210 004767 000110 JSR          PC,LDCDSUB ;GO EXECUTE THE INSTRUCTION.
4139 023214 177777 177526 1$: .WORD -1,-252 ;FSRC OPERAND.
4140 023220 142052 000000 000000 2$: .WORD 142052,0,0,0 ;EXPECTED RESULT.
4141 023226 000000          3$: .WORD 136052,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
4142 023230 136052 000000 000000 4$: 307 ;FPS BEFORE EXECUTION.
4143 023236 000000          310 ;FPS AFTER EXECUTION.
4144 023240 000307          -1 ;ANTICIPATED ERRONEOUS FPS.
4145 023242 000310          5$: ERROR 1 ;REPORT RESULT INCORRECT.
4146 023244 177777          BR 6$
4147 023246 104001          ERROR 1 ;REPORT FPS INCORRECT.
4148 023250 000401          6$:
4149 023252 104001          ;OPERAND=PATTERN FL=1 FD=1 FT=1
4150 023254          LLC10:
(1) 023254 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4152 023256 004767 000042 JSR          PC,LDCDSUB ;GO EXECUTE THE INSTRUCTION.
4153 023262 012345 067012 1$: .WORD 12345,67012 ;FSRC OPERAND.
    
```

```

4154 023266 047247 025560 050000 2$: .WORD 47247,025560,050000,0 ;EXPECTED RESULT.
      023274 000000
4155 023276 177777 177777 177777 3$: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
      023304 177777
4156 023306 000352 4$: 352 ;FPS BEFORE EXECUTION.
4157 023310 000340 ;FPS AFTER EXECUTION.
4158 023312 177777 ;ANTICIPATED ERRONEOUS FPS.
4159 023314 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
4160 023316 000401 BR 6$
4161 023320 104001 ERROR 1 ;REPORT FPS INCORRECT.
4162 023322 000502 6$: BR LLCDONE

```

```

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

```

```

4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194

```

```

      JSR      PC,@#LDCDSUB
      ACARG:  .WORD  X,X           ;AC OPERAND
      RES:    .WORD  X,X,X,X      ;EXPECTED RESULT
      ERRES:  .WORD  X,X,X,X      ;ERROR RESULT
      FPSB:   .WORD  X           ;FPS BEFORE EXECUTION
      FPSA:   .WORD  X           ;FPS AFTER EXECUTION
      ERFPS:  .WORD  X           ;ERROR FPS.
      ERR1:   ERROR 1;DATA ERROR.
      BR      CONT
      ERR2:   ERROR 1;FPS ERROR.
      CONT:   ;RETURN ADDRESS

```

```

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

```

```

4195 023324 012601 LDCDSUB: MOV (SP)+,R1 ;GET A POINTER TO THE ARGUMENTS.
4196 023326 016100 000024 MOV 24(R1),R0 ;SET THE FPS.
4197 023332 170100 LDFPS R0
4198 023334 012737 023344 001236 MOV #1$,@#$TMP2
4199 023342 010100 MOV R1,R0
4200 023344 177010 1$: LDCID (R0),ACO ;TEST INSTRUCTION, LDCID OR LDCLD.
4201
4202 023346 170204 STFPS R4 ;GET FPS.
4203 023350 012700 022512 MOV #LDCT,R0 ;GET THE RESULT.
4204 023354 012702 000200 MOV #200,R2
4205 023360 170102 LDFPS R2
4206 023362 174010 STD ACO,(R0)
4207

```



```
4208 ;SEE IF THE RESULT IS CORRECT.
4209 023364 012702 022512 MOV #LDCT,R2
4210 023370 010237 001242 MOV R2,@#STMP4
4211 023374 010137 001240 MOV R1,@#STMP3
4212 023400 010103 MOV R1,R3
4213 023402 062703 000004 ADD #4,R3
4214 023406 010337 001244 MOV R3,@#STMP5
4215 023412 010437 001250 MOV R4,@#STMP7
4216 023416 016137 000026 001252 MOV 26(R1),@#STMP10
4217 023424 010100 MOV R1,R0
4218 023426 062700 000004 ADD #4,R0
4219 023432 012703 000002 MOV #2,R3
4220 023436 022022 2$: CMP (R0)+,(R2)+
4221 023440 001006 BNE 10$ ;BR IF INCORRECT.
4222 023442 077303 SOB R3,2$
4223
4224 023444 026104 000026 CMP 26(R1),R4 ;IS THE FPS CORRECT?
4225 023450 001020 BNE 15$ ;BR IF INCORRECT.
4226 023452 000161 000040 3$: JMP 40(R1) ;RETURN.
4227
4228 ;THE RESULT WAS INCORRECT SO SEE IF THE ERROR WAS ANTICIPATED.
4229 023456 012702 022512 10$: MOV #LDCT,R2
4230 023462 010100 MOV R1,R0
4231 023464 062700 000014 ADD #14,R0
4232 023470 012703 000002 MOV #2,R3
4233 023474 022022 11$: CMP (R0)+,(R2)+
4234 023476 001003 BNE 13$
4235 023500 077303 SOB R3,11$
4236 023502 000161 000032 JMP 32(R1)
4237 023506 13$:
4238 ;ERROR NOT ANTICIPATED SO REPORT RESULT INCORRECT HERE.
4239 023506 104001 14$: ERROR 1 ;BAD RES
4240 023510 000760 BR 3$
4241
4242 ;THE FPS WAS INCORRECT. SEE IF FAILURE WAS ANTICIPATED.
4243 023512 026104 000030 15$: CMP 30(R1),R4
4244 023516 001002 BNE 16$
4245 023520 000161 000036 JMP 36(R1)
4246 ;FPS ERROR WAS NOT ANTICIPATED SO REPORT FAILURE HERE.
4247 023524 16$:
4248
4249 023524 104001 17$: ERROR 1 ;BAD FPS
4250 023526 000751 BR 3$
4251
4252 023530 LLCDONE:
(1) 023530 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
4253
4262
4263
(3) ;*****
(4) ;*TEST 60 LDEXP TEST
(4) ;*
(4) ;* THIS IS A TEST OF THE LDEXP INST
```

```
(4)          ;* A SUBROUTINE IS USED TO SET UP
(4)          ;* OPERANDS, EXECUTE THE LDEXP INST AND
(4)          ;* CHECK THE RESULTS.
(4)          ;*
(3)          ;*****
(2) 023532 000004 TST60: SCOPE
4264
4265          ; NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
4266 023534          MMC1:
(1) 023534 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4267 023536 004767 001334 JSR          PC,LDXSUB          ;GO EXECUTE THE INSTRUCTION.
4268 023542 012345 067012 034567 1$: .WORD 12345,67012,34567,012345          ;ACO OPERAND.
         023550 012345
4269 023552 000010          2$: .WORD 10          ;EXPONENT OPERAND.
4270 023554 042145 067012 034567 3$: .WORD 42145,67012,34567,012345          ;EXPECTED RESULT.
         023562 012345
4271 023564 002145 067012 034567 4$: .WORD 2145,67012,34567,012345          ;ANTICIPATED ERRONEOUS RESULT.
         023572 012345
4272 023574 047217          5$: 47217          ;FPS BEFORE EXECUTION.
4273 023576 047200          47200          ;FPS AFTER EXECUTION.
4274 023600 147200          147200          ;ANTICIPATED ERRONEOUS FPS.
4275 023602 177777          -1          ;EXPECTED FEC.
4276 023604 104001          6$: ERROR 1          ;E12+E12+200 BAD
4277 023606 000400          BR 7$          ;ST 624
4278 023610 104001          7$: ERROR 1          ;REPORT FPS INCORRECT.
4279          ;ST 625 INTO 304
4280
4281          ;NON-ZERO RES NEG.
4282 023612          MMC2:
(1) 023612 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4283 023614 004737 025076 JSR          PC,@#LDXSUB          ;EXPON=377
4284 023620 123456 070123 045670 1$: .WORD 123456,70123,45670,123456          ;ACO OPERAND.
         023626 123456
4285 023630 000177          2$: .WORD 177          ;EXPONENT OPERAND.
4286 023632 177656 070123 045670 3$: .WORD 177656,70123,45670,123456          ;EXPECTED RESULT.
         023640 123456
4287 023642 137656 070123 045670 4$: .WORD 137656,70123,45670,123456          ;ANTICIPATED ERRONEOUS RESULT.
         023650 123456
4288 023652 047207          5$: 47207          ;FPS BEFORE EXECUTION.
4289 023654 047210          47210          ;FPS AFTER EXECUTION.
4290 023656 147210          147210          ;ANTICIPATED ERRONEOUS FPS.
4291 023660 177777          -1          ;EXPECTED FEC.
4292 023662 104001          6$: ERROR 1          ;REPORT RESULT INCORRECT.
4293 023664 000401          BR 7$
4294 023666 104001          7$: ERROR 1          ;REPORT FPS INCORRECT.
4295 023670
4296
4297          ;NON-ZERO RES, EXP=256=(56)REAL
4298 023670          MMC3:
(1) 023670 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
4299 023672 004737 025076 JSR          PC,@#LDXSUB          ;GO EXECUTE THE INSTRUCTION.
4300 023676 073261 057645 043323 1$: .WORD 73261,057645,43323,101760          ;ACO OPERAND.
         023704 101760
4301 023706 000056          2$: .WORD 56          ;EXPONENT OPERAND.
4302 023710 053461 057645 043323 3$: .WORD 53461,057645,43323,101760          ;EXPECTED RESULT.
         023716 101760
```



```

4303 023720 177777 177777 177777 4$: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
      023726 177777
4304 023730 047200 5$: 47200 ;FPS BEFORE EXECUTION.
4305 023732 047200 47200 ;FPS AFTER EXECUTION.
4306 023734 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
4307 023736 177777 -1 ;EXPECTED FEC.
4308 023740 104001 6$: ERROR 1 ;REPORT RESULT INCORRECT.
4309 023742 000401 BR 7$
4310 023744 104001 ERROR 1 ;REPORT FPS INCORRECT.
4311 023746 7$:
4312
4313 ;EXP=27 (EXCESS 200)=-151 (OCT)
4314 023746 MMC4:
      (1) 023746 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4315 023750 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4316 023754 012223 024252 062720 1$: .WORD 12223,24252,62720,21222 ;ACO OPERAND.
      023762 021222
4317 023764 177627 2$: .WORD -151 ;EXPONENT OPERAND.
4318 023766 005623 024252 062720 3$: .WORD 5623,24252,62720,21222 ;EXPECTED RESULT.
      023774 021222
4319 023776 177777 177777 177777 4$: .WORD -1,-1,-1,-1 ;ANTICIPATED ERRONEOUS RESULT.
      024004 177777
4320 024006 047200 5$: 47200 ;FPS BEFORE EXECUTION.
4321 024010 047200 47200 ;FPS AFTER EXECUTION.
4322 024012 147200 147200 ;ANTICIPATED ERRONEOUS FPS.
4323 024014 177777 -1 ;EXPECTED FEC.
4324 024016 104001 6$: ERROR 1 ;REPORT RESULT INCORRECT.
4325 024020 000401 BR 7$
4326 024022 104001 ERROR 1 ;(BUT EZBT) ST 544 TO 504 INTO 704 0 (BUT EXBT) ST 704 I
4327 024024 7$:
4328
4329 ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
4330 ; FIV=1
4331 024024 MMC5:
      (1) 024024 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4332 024026 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4333 024032 030131 032334 035363 1$: .WORD 30131,32334,35363,73031 ;ACO OPERAND.
      024040 073031
4334 024042 177600 2$: .WORD -200 ;EXPONENT OPERAND.
4335 024044 000131 032334 035363 3$: .WORD 00131,32334,35363,73031 ;EXPECTED RESULT.
      024052 073031
4336 024054 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
      024062 000000
4337 024064 042200 5$: 42200 ;FPS BEFORE EXECUTION.
4338 024066 142204 142204 ;FPS AFTER EXECUTION.
4339 024070 042202 42202 ;ANTICIPATED ERRONEOUS FPS.
4340 024072 000012 12 ;EXPECTED FEC.
4341 024074 104001 6$: ERROR 1 ;(BUT EXBT) ST 704 TO 64 INST 264
4342 024076 000401 BR 7$
4343 024100 104001 ERROR 1 ;(BUT FIU) ST 264 X
4344 024102 7$:
4345
4346 ;EXP=0 (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
4347 024102 MMC6:
      (1) 024102 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4348 024104 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
    
```

J 9
PAGE 1-96

CJKDDB KEF11-A DIAG PART 2 MACY11 30A(1052) 12-MAR-80 07:25
 CJKDDB.P11 12-MAR-80 07:22 T60 LDEXP TEST SEQ 0113

```

4349 024110 140414 024344 045464 1$: .WORD 140414,24344,45464,74045 ;ACO OPERAND.
      024116 074045
4350 024120 177600 2$: .WORD -200 ;EXPONENT OPERAND.
4351 024122 100014 024344 045464 3$: .WORD 100014,24344,45464,74045 ;-0 ;EXPECTED RESULT.
      024130 074045
4352 024132 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
      024140 000000
4353 024142 042200 5$: 42200 ;FPS BEFORE EXECUTION.
4354 024144 142214 142214 ;FPS AFTER EXECUTION.
4355 024146 042214 42214 ;ANTICIPATED ERRONEOUS FPS.
4356 024150 000012 12 ;EXPECTED FEC.
4357 024152 104001 6$: ERROR 1 ;REPORT RESULT INCORRECT.
4358 024154 000401 BR 7$
4359 024156 104001 ERROR 1 ;REPORT FPS INCORRECT.
4360 024160 7$:
4361
4362 ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
4363
4364 024160 MMC7:
      (1) 024160 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4365 024162 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4366 024166 051525 035455 005675 1$: .WORD 51525,35455,5675,05152 ;ACO OPERAND.
      024174 005152
4367 024176 177600 2$: .WORD -200 ;EXPONENT OPERAND.
4368 024200 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
      024206 000000
4369 024210 000125 035455 005675 4$: .WORD 00125,35455,5675,05152 ;ANTICIPATED ERRONEOUS RESULT.
      024216 005152
4370 024220 045200 45200 ;FPS BEFORE EXECUTION.
4371 024222 045204 45204 ;FPS AFTER EXECUTION.
4372 024224 145204 145204 ;ANTICIPATED ERRONEOUS FPS.
4373 024226 177777 -1 ;EXPECTED FEC.
4374 024230 104001 6$: ERROR 1 ;(BUT FIU) ST 264 X ;REPORT RESULT INCORRECT
4375 024232 000401 BR 7$
4376 024234 104001 ERROR 1 ;REPORT FPS INCORRECT.
4377 024236 7$:
4378
4379 ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
4380 024236 MMC8:
      (1) 024236 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4381 024240 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4382 024244 061626 062636 046566 1$: .WORD 61626,62636,46566,67606 ;ACO OPERAND.
      024252 067606
4383 024254 176173 2$: .WORD -1605 ;EXPONENT OPERAND.
4384 024256 076626 062636 046566 3$: .WORD 76626,62636,46566,67606 ;EXPECTED RESULT.
      024264 067606
4385 024266 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
      024274 000000
4386
4387 024276 042200 5$: 42200 ;FPS BEFORE EXECUTION.
4388 024300 142200 142200 ;FPS AFTER EXECUTION.
4389 024302 042204 42204 ;ANTICIPATED ERRONEOUS FPS.
4390 024304 000012 12 ;EXPECTED FEC.
4391 024306 104001 6$: ERROR 1 ;(BUT EZBT) ST 544 TO 704 INTO 504
4392 024310 000401 BR 7$
4393 024312 104001 ERROR 1 ;REPORT FPS INCORRECT.

```



```
4394 024314 7$:  
4395 ;EXP=-17416 (EXCESS 200)=-17616 (OCT), FIU=0  
4396 024314 MMC9:  
  (1) 024314 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
4397 024316 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.  
4398 024322 071727 037475 076777 1$: .WORD 71727,37475,76777,17273 ;ACO OPERAND.  
  024330 017273  
4399 024332 160162 2$: .WORD -17616 ;EXPONENT OPERAND.  
4400 024334 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.  
  024342 000000  
4401 024344 074527 037475 076777 4$: .WORD 74527,37475,76777,17273 ;ANTICIPATED ERRONEOUS RESULT.  
  024352 017273  
4402 024354 045200 5$: 45200 ;FPS BEFORE EXECUTION.  
4403 024356 045204 45204 ;FPS AFTER EXECUTION.  
4404 024360 145200 145200 ;ANTICIPATED ERRONEOUS FPS.  
4405 024362 177777 -1 ;EXPECTED FEC.  
4406 024364 104001 6$: ERROR 1 ;(BUT FIU) ST 504  
4407 024366 000401 BR 7$  
4408 024370 104001 ERROR 1 ;REPORT FPS INCORRECT.  
4409 024372 7$:  
4410  
4411 ;EXP=-1601 (EXCESS 200)=-2001 (OCT), FIU=1  
4412 024372 MMC10:  
  (1) 024372 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
4413 024374 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.  
4414 024400 001020 030405 006070 1$: .WORD 01020,30405,06070,00102 ;ACO OPERAND.  
  024406 000102  
4415 024410 175777 2$: .WORD -2001 ;EXPONENT OPERAND.  
4416 024412 037620 030405 006070 3$: .WORD 37620,30405,06070,00102 ;EXPECTED RESULT.  
  024420 000102  
4417 024422 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.  
  024430 000000  
4418 024432 042200 5$: 42200 ;FPS BEFORE EXECUTION.  
4419 024434 142200 142200 ;FPS AFTER EXECUTION.  
4420 024436 042204 42204 ;ANTICIPATED ERRONEOUS FPS.  
4421 024440 000012 12 ;EXPECTED FEC.  
4422 024442 104001 6$: ERROR 1 ;(BUT FIU) ST 504  
4423 024444 000401 BR 7$  
4424 024446 104001 ERROR 1 ;REPORT FPS INCORRECT.  
4425 024450 7$:  
4426  
4427 ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1  
4428 024450 MMC11:  
  (1) 024450 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
4429 024452 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.  
4430 024456 012131 014151 016171 1$: .WORD 12131,14151,16171,10111 ;ACO OPERAND.  
  024464 010111  
4431 024466 001006 2$: .WORD 1006 ;EXPONENT OPERAND.  
4432 024470 041531 014151 016171 3$: .WORD 41531,14151,16171,10111 ;EXPECTED RESULT.  
  024476 010111  
4433 024500 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.  
  024506 000000  
4434 024510 041200 5$: 41200 ;FPS BEFORE EXECUTION.  
4435 024512 141202 141202 ;FPS AFTER EXECUTION.  
4436 024514 041204 41204 ;ANTICIPATED ERRONEOUS FPS.  
4437 024516 000010 10 ;EXPECTED FEC.
```

```
4438 024520 104001 6$: ERROR 1 ;(BUT FIV) ST 104
4439 024522 000401 BR 7$
4440 024524 104001 ERROR 1 ;REPORT FPS INCORRECT.
4441 024526 7$:
4442 ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
4443 MMC12:
4444 024526 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 024526 104414 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4445 024530 004737 025076 .WORD 27262,25242,23222,21202 ;ACO OPERAND.
4446 024534 027262 025242 023222 1$:
024542 021202
4447 024544 016115 2$: .WORD 16115 ;EXPONENT OPERAND.
4448 024546 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
024554 000000
4449 024556 063262 025242 023222 4$: .WORD 63262,25242,23222,21202 ;ANTICIPATED ERRONEOUS RESULT.
024564 021202
4450 024566 046200 5$: 46200 ;FPS BEFORE EXECUTION.
4451 024570 046206 46206 ;FPS AFTER EXECUTION.
4452 024572 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
4453 024574 177777 -1 ;EXPECTED FEC.
4454 024576 104001 6$: ERROR 1 ;(BUT FIV) ST 104
4455 024600 000401 BR 7$
4456 024602 104001 ERROR 1 ;REPORT FPS INCORRECT.
4457 024604 7$:
4458 ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1
4459 MMC13:
4460
4461 024604 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 024604 104414 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4462 024606 004737 025076 .WORD 30313,32333,34353,36373 ;ACO OPERAND.
4463 024612 030313 032333 034353 1$:
024620 036373
4464 024622 010611 2$: .WORD 10611 ;EXPONENT OPERAND.
4465 024624 002313 032333 034353 3$: .WORD 2313,32333,34353,36373 ;EXPECTED RESULT.
024632 036373
4466 024634 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
024642 000000
4467 024644 041200 5$: 41200 ;FPS BEFORE EXECUTION.
4468 024646 141202 141202 ;FPS AFTER EXECUTION.
4469 024650 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
4470 024652 000010 10 ;EXPECTED FEC.
4471 024654 104001 6$: ERROR 1 ;(BUT FIV) ST 144
4472 024656 000401 BR 7$
4473 024660 104001 ERROR 1 ;REPORT FPS INCORRECT.
4474 024662 7$:
4475 ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
4476 MMC14:
4477
4478 024662 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 024662 104414 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4479 024664 004737 025076 .WORD 40414,42434,44454,46474 ;ACO OPERAND.
4480 024670 040414 042434 044454 1$:
024676 046474
4481 024700 016723 2$: .WORD 16723 ;EXPONENT OPERAND.
4482 024702 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
024710 000000
```



```

4483 024712 024614 042434 044454 4$: .WORD 24614,42434,44454,46474 ;ANTICIPATED ERRONEOUS RESULT.
      024720 046474
4484 024722 046200 5$: 46200 ;FPS BEFORE EXECUTION.
4485 024724 046206 46206 ;FPS AFTER EXECUTION.
4486 024726 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
4487 024730 177777 -1 ;EXPECTED FEC.
4488 024732 104001 6$: ERROR 1 ;(BUT FIV) ST 144
4489 024734 000401 BR 7$
4490 024736 104001 ERROR 1 ;REPORT FPS INCORRECT.
4491 024740 7$:
4492
4493 ;EXP= 254 (OCT)= 454 (EXCESS 200) FIV=1
4494
4495 024740 MMC15:
      (1) 024740 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4496 024742 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4497 024746 050515 052535 054555 1$: .WORD 50515,52535,54555,56575 ;ACO OPERAND.
      024754 056575
4498 024756 000254 2$: .WORD 254 ;EXPONENT OPERAND.
4499 024760 013115 052535 054555 3$: .WORD 13115,52535,54555,56575 ;EXPECTED RESULT.
      024766 056575
4500 024770 000000 000000 000000 4$: .WORD 0,0,0,0 ;ANTICIPATED ERRONEOUS RESULT.
      024776 000000
4501 025000 041200 5$: 41200 ;FPS BEFORE EXECUTION.
4502 025002 141202 141202 ;FPS AFTER EXECUTION.
4503 025004 041204 41204 ;ANTICIPATED ERRONEOUS FPS.
4504 025006 000010 10 ;EXPECTED FEC.
4505 025010 104001 6$: ERROR 1 ;(BUT FIV) ST344
4506 025012 000401 BR 7$
4507 025014 104001 ERROR 1 ;REPORT FPS INCORRECT.
4508 025016 7$:
4509
4510 ;EXP= 313 (OCT)= 513(EXCESS 200) FIV=0
4511
4512 025016 MMC16:
      (1) 025016 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
4513 025020 004737 025076 JSR PC,@#LDXSUB ;GO EXECUTE THE INSTRUCTION.
4514 025024 060616 062636 064656 1$: .WORD 60616,62636,64656,66676 ;ACO OPERAND.
      025032 066676
4515 025034 000313 2$: .WORD 313 ;EXPONENT OPERAND.
4516 025036 000000 000000 000000 3$: .WORD 0,0,0,0 ;EXPECTED RESULT.
      025044 000000
4517 025046 022616 062636 064656 4$: .WORD 22616,62636,64656,66676 ;ANTICIPATED ERRONEOUS RESULT.
      025054 066676
4518 025056 046200 5$: 46200 ;FPS BEFORE EXECUTION.
4519 025060 046206 46206 ;FPS AFTER EXECUTION.
4520 025062 146202 146202 ;ANTICIPATED ERRONEOUS FPS.
4521 025064 177777 -1 ;EXPECTED FEC.
4522 025066 104001 6$: ERROR 1 ;(BUT FIV) ST 344
4523 025070 000401 BR 7$
4524 025072 104001 ERROR 1 ;REPORT FPS INCORRECT.
4525 025074 7$:
4526 025074 000540 BR MMCDONE
4527
4528 ;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
4529 ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
    
```

4530
 4531
 4532
 4533
 4534
 4535
 4536
 4537
 4538
 4539
 4540
 4541
 4542
 4543
 4544
 4545
 4546
 4547
 4548
 4549
 4550
 4551
 4552
 4553
 4554
 4555
 4556
 4557
 4558
 4559
 4560
 4561
 4562
 4563
 4564
 4565
 4566
 4567
 4568
 4569
 4570
 4571
 4572
 4573
 4574
 4575
 4576
 4577
 4578
 4579
 4580
 4581
 4582
 4583
 4584
 4585

:TO IT IS MADE THUS:

```

      JSR      PC,@#LDXSUB
      ACARG:  .WORD  X,X,X,X      ;AC OPERAND
      EXP:    .WORD  X            ;EXPONENT
      RES:    .WORD  X,X,X,X      ;EXPECTED RESULT
      ERRES:  .WORD  X,X,X,X      ;ERROR RESULT
      FPSB:   .WORD  X            ;FPS BEFORE EXECUTION
      FPSA:   .WORD  X            ;FPS AFTER EXECUTION
      ERFPS:  .WORD  X            ;ERROR FPS.
      FEC:    .WORD  X            ;EXPECTED FEC
      ERR1:   ERROR  1;DATA ERROR.
      BR      CONT
      ERR2:   ERROR  1;FPS ERROR.
      CONT:   ;RETURN ADDRESS
  
```

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

```

LDXSUB: MOV      (SP)+,R1      ;GET A POINTER TO THE ARGUMENTS.
        MOV      #200,R0      ;LOAD THE ACO OPERAND.
        LDFPS   R0
        MOV      R1,R0
        LDD     (R0),ACO
        MOV      #1$,@#STMP2
        MOV      32(R1),R0    ;SET UP THE FPS.
        LDFPS   R0
        MOV      R1,R0
        ADD     #10,R0

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

        STFPS   R4            ;GET THE FPS.
        STST    R5            ;GET THE FEC.
        MOV     #200,R0      ;GET THE RESULT.
        LDFPS   R0
        MOV     #LDXT,R0
        STD     ACO,(R0)
        MOV     R4,@#STMP7
        MOV     34(R1),@#STMP10
        MOV     R5,@#STMP11
        MOV     40(R1),@#STMP12
        MOV     R1,R2
        MOV     R2,@#STMP3
  
```

```

025076 012601
025100 012700 000200
025104 170100
025106 010100
025110 172410
025112 012737 025134 001236
025120 016100 000032
025124 170100
025126 010100
025130 062700 000010
025134 176410
025136 170204
025140 170305
025142 012700 000200
025146 170100
025150 012700 025366
025154 174010
025156 010437 001250
025162 016137 000034 001252
025170 010537 001254
025174 016137 000040 001256
025202 010102
025204 010237 001240
  
```



```

4586 025210 062702 000010      ADD    #10,R2
4587 025214 011237 001242      MOV    (R2),@#STMP4
4588 025220 062702 000002      ADD    #2,R2
4589 025224 010237 001244      MOV    R2,@#STMP5
4590 025230 012737 025366      MOV    #LDXT,@#STMP6
4591 025236 012702 025366      MOV    #LDXT,R2      ;SEE IF THE RESULT WAS CORRECT.
4592 025242 010103      MOV    R1,R3
4593 025244 062703 000012      ADD    #12,R3
4594 025250 012700 000004      MOV    #4,R0
4595 025254 022223      2$:   CMP    (R2)+,(R3)+
4596 025256 001014      BNE    10$      ;BRANCH IF NOT CORRECT.
4597 025260 077003      SOB    R0,2$
4598 025262 020461 000034      CMP    R4,34(R1)      ;SEE IF THE FPS WAS CORRECT.
4599 025266 001026      BNE    15$      ;BRANCH IF NOT CORRECT.
4600 025270 005761 000034      TST    34(R1)
4601 025274 100003      BPL    3$
4602 025276 020561 000040      CMP    R5,40(R1)      ;SEE IF THE FEC WAS CORRECT.
4603 025302 001027      BNE    20$      ;BRANCH IF NOT CORRECT.
4604
4605 025304 000161 000050      3$:   JMP    50(R1)      ;RETURN.
4606
4607      ;THE RESULT WAS INCORRECT SO SEE IF THE FAILURE WAS ANTICIPATED.
4608 025310 012702 025366      10$:  MOV    #LDXT,R2
4609 025314 010103      MOV    R1,R3
4610 025316 062703 000022      ADD    #22,R3
4611 025322 012700 000004      MOV    #4,R0
4612 025326 022223      11$:  CMP    (R2)+,(R3)+
4613 025330 001003      BNE    12$
4614 025332 077003      SOB    R0,11$
4615 025334 000161 000042      JMP    42(R1)
4616
4617      ;THE ERROR WAS NOT ANTICIPATED SO REPORT IT HERE.
4618 025340      12$:
4619 025340 104001      13$:  ERROR 1      ;BAD RES
4620 025342 000760      BR    3$
4621
4622      ;SEE IF THE FPS ERROR WAS ANTICIPATED.
4623 025344 026104 000036      15$:  CMP    36(R1),R4
4624 025350 001002      BNE    16$
4625 025352 000161 000046      JMP    46(R1)
4626 025356
4627      ;THE FPS WAS NOT ANTICIPATED SO REPORT IT HERE.
4628 025356 104001      17$:  ERROR 1      ;BAD FPS
4629 025360 000751      BR    3$      ;BUT EZBTY8
4630      ;ST 063
4631
4632 025362      20$:
4633      ;REPORT FEC INCORRECT.
4634 025362 104001      21$:  ERROR 1      ;BAD FEC
4635 025364 000747      BR    3$
4636
4637      ;DATA BUFFER:
4638 025366 000000 000000 000000 LDXT: .WORD 0,0,0,0
4639 025374 000000
4640 025376      MMCDONE:

```

(1) 025376 104413

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?).

(1)
(1)
(1)
(1)
(1)
4641
4642
4649
4650

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

(3)
(4)
(4)
(4)
(4)
(3)
(2) 025400 000004

4651

4652

4653

(1)

4654

4655

4656

4657

4658

4659

4660

4661

4662

4663

4664

4665

4666

4667

4668

4669

4670

4671

4672

4673

4674

4675

4676

4677

4678

4679

4680

4681

4682

4683

4684

4685

4686

4687

4688

025402 104414
025402 012700 025502
025404 012701 000006
025410 012720 177777
025414 012700 102345
025420 012737 025450 001236
025426 012737 025574 000004
025434 170100
025442 012700 025506
025444 170210
025450 020027 025506
025452 001017
025456 023727 025506 102345
025460 001022
025466 023727 025510 177777
025470 001026
025476 000447
025502 177777 177777
025506 177777 177777
025514 177777
025516 010037 001242
025522 012737 025506 001240
025530 104001
025532 000432
025534 012737 102345 001240
025542 013737 025506 001242
025550 104001
025552 000422

NNC1: LPERR :SET UP THE LOOP ON ERROR ADDRESS.
MOV #NNCTB0,R0 :SET UP THE DATA BUFFER.
MOV #6,R1
1\$: MOV #-1,(R0)+
SOB R1,1\$
MOV #102345,R0
MOV #NNC2,@\$STMP2
MOV #NNC25,@\$ERRVECT :SET UP FOR TRAPS TO 4.
LDFPS R0 :SET UP FPS.
MOV #NNCTB1,R0
NNC2: STFPS (R0) :TEST INSTRUCTION.
CMP R0,#NNCTB1 :IS R0 CORRECT?
BNE NNC10 :BRANCH IF NOT CORRECT.
CMP @NNCTB1,#102345 :IS RESULT CORRECT?
BNE NNC15 :BRANCH IF NOT CORRECT.
CMP @NNCTB1+2,#-1 :IS THE RESULT CORRECT?
BNE NNC20 :BRANCH IF NOT CORRECT.
BR NNCDONE
:TEST DATA BUFFER:
NNCTB0: .WORD -1,-1
NNCTB1: .WORD -1,-1,-1,-1
:REPORT R0 INCORRECT.
NNC10: MOV R0,@\$STMP4
MOV #NNCTB1,@\$STMP3
1\$: ERROR 1 :R0 BAD (BUT
BR NNCDONE ; FDST)X
:REPORT RESULT INCORRECT.
NNC15: MOV #102345,@\$STMP3 : ST 634
MOV @NNCTB1,@\$STMP4
1\$: ERROR 1 :BAD DATA
BR NNCDONE


```

4689
4690
4691 025554 012737 177777 001240
4692 025562 013737 025510 001242
4693 025570 104001
4694 025572 000412
4695
4696
4697
4698
4699
4700 025574 011604
4701 025576 020427 025452
4702 025602 001402
4703 025604 000137 036166
4704
4705 025610 011637 001236
4706 025614 022626
4707 025616 104001
4708
4709 025620
(1) 025620 104413
(1)
(1)
(1)
(1)
(1)
4710
4711
4712
(3)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 025622 000004
4713
4714
4715 025624
(1) 025624 104414
4716 025626 012700 025724
4717 025632 012701 000006
4718 025636 012720 177777
4719 025642 077103
4720 025644 012700 105412
4721 025650 012737 025672 001236
4722 025656 012737 026016 000004
4723 025664 170100
4724 025666 012700 025730
4725
4726 025672 170220
4727 025674 020027 025732
4728 025700 001017
4729 025702 023727 025730 105412
4730 025710 001022
4731 025712 023727 025732 177777
    
```

```

;REPORT RESULT INCORRECT.
NNC20: MOV #-1,@#STMP3
MOV @#NNCTB1+2,@#STMP4
1$: ERROR 1 ;(BUT GR7,FL)
BR NNCDONE ;ST 357 TO 416
;INTO 417

;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
;TO THE SPURIOUS TRAP TO 4 HANDLER.
NNC25: MOV (SP),R4
CMP R4,#NNC2+2
BEQ 1$
JMP @#CPSPUR

1$: MOV (SP),@#STMP2
CMP (SP)+,(SP)+
2$: ERROR 1 ;(BUT FDST)+ ST634

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

```

TST62: SCOPE

OOC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #OOC2,R0 ;SET UP THE DATA BUFFER.
MOV #6,R1
1$: MOV #-1,(R0)+
SOB R1,1$
MOV #105412,R0
MOV #OOC2,@#STMP2
MOV #OOC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS R0 ;SET UP FPS.
MOV #OOC2B1,R0

OOC2: STFPS (R0)+ ;TEST INSTRUCTION.
CMP R0,#OOC2B1+2 ;IS R0 CORRECT?
BNE OOC10 ;BRANCH IF NOT CORRECT.
CMP @#OOC2B1,#105412 ;IS THE RESULT CORRECT?
BNE OOC15 ;BRANCH IF NOT CORRECT.
CMP @#OOC2B1+2,#-1 ;IS THE RESULT CORRECT?
    
```

```

4732 025720 001026          BNE      OOC20          ;BRANCH IF NOT CORRECT.
4733 025722 000447          BR       OOCDONE
4734
4735          ;TEST DATA BUFFER:
4736 025724 177777 177777  OOC20:  .WORD  -1,-1
4737 025730 177777 177777 177777 OOC21:  .WORD  -1,-1,-1,-1
4738 025736 177777
4739          ;REPORT R0 INCORRECT.
4740 025740 010037 001242  OOC10:  MOV      R0,@#STMP4
4741 025744 012737 025732 001240  MOV      #OOC21+2,@#STMP3
4742 025752 104001          1$:      ERROR 1          ;R0 BAD (BUT
4743 025754 000432          BR       OOCDONE          ; FDST)X
4744
4745          ;REPORT RESULT INCORRECT.
4746 025756 012737 105412 001240  OOC15:  MOV      #105412,@#STMP3          ; ST 634
4747 025764 013737 025730 001242  MOV      @#OOC21,@#STMP4
4748 025772 104001          1$:      ERROR 1          ;BAD DATA
4749 025774 000422          BR       OOCDONE
4750
4751          ;REPORT RESULT INCORRECT.
4752 025776 012737 177777 001240  OOC20:  MOV      #-1,@#STMP3
4753 026004 013737 025732 001242  MOV      @#OOC21+2,@#STMP4
4754 026012 104001          1$:      ERROR 1          ;(BUT GR7,FL)
4755 026014 000412          BR       OOCDONE          ;ST 357 TO 416
4756
4757          ;INTO 417
4758
4759          ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
4760          ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
4761          ;TO THE SPURIOUS TRAP TO 4 HANDLER.
4762 026016 011604          OOC25:  MOV      (SP),R4
4763 026020 020427 025674          CMP      R4,#OOC2+2
4764 026024 001402          BEQ     1$
4765 026026 000137 036166          JMP     @#CPSPUR
4766
4767 026032 011637 001236          1$:      MOV      (SP),@#STMP2
4768 026036 022626          CMP     (SP)+,(SP)+
4769 026040 104001          2$:      ERROR 1          ;(BUT FDST)+ ST634
4770
4771 026042          OOCDONE:
(1) 026042 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)          ;SEE IF THE USER HAS EXPRESSED
(1)          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)          ;THE USER TYPED CONTROL G?).
4772
4773
4774
4775          ;*****
(3)          ;*TEST 63 DESTINATION MODES, MODE 4 (FL=0), TEST
(4)          ;*
(4)          ;* THIS IS A TEST OF DESTINATION MODE 4 USING
(4)          ;* THE STFPS INSTRUCTION
(4)          ;*
(3)          ;*****
    
```



```
(2) 026044 000004
4776
4777 026046
(1) 026046 104414
4778 026050 012700 026146
4779 026054 012701 000006
4780 026060 012720 177777
4781 026064 077103
4782 026066 012700 105555
4783 026072 012737 026114 001236
4784 026100 012737 026240 000004
4785 026106 170100
4786 026110 012700 026154
4787
4788 026114 170240
4789 026116 020027 026152
4790 026122 001017
4791 026124 023727 026152 105555
4792 026132 001022
4793 026134 023727 026154 177777
4794 026142 001026
4795 026144 000447
4796
4797
4798 026146 177777 177777
4799 026152 177777 177777 177777
026160 177777

4800
4801
4802 026162 010037 001242
4803 026166 012737 026152 001240
4804 026174 104001
4805 026176 000432
4806
4807
4808 026200 012737 105555 001240
4809 026206 013737 026152 001242
4810 026214 104001
4811 026216 000422
4812
4813
4814
4815 026220 012737 177777 001240
4816 026226 013737 026154 001242
4817 026234 104001
4818 026236 000412
4819
4820
4821
4822
4823
4824 026240 011604
4825 026242 020427 026116
4826 026246 001402
4827 026250 000137 036166
4828

TST63: SCOPE
PPC1:
      LPERR                ;SET UP THE LOOP ON ERROR ADDRESS.
      MOV #PPCTB0,R0        ;SET UP THE DATA BUFFER.
      MOV #6,R1
1$:   MOV #-1,(R0)+
      SOB R1,1$
      MOV #105555,R0
      MOV #PPC2,@#STMP2
      MOV #PPC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
      LDFPS R0              ;SET UP FPS.
      MOV #PPCTB1+2,R0
PPC2: STFPS -(R0)          ;TEST INSTRUCTION.
      CMP R0,#PPCTB1        ;IS R0 CORRECT?
      BNE PPC10             ;BRANCH IF NOT CORRECT.
      CMP @#PPCTB1,#105555 ;IS THE RESULT CORRECT?
      BNE PPC15             ;BRANCH IF NOT CORRECT.
      CMP @#PPCTB1+2,#-1    ;IS THE RESULT CORRECT?
      BNE PPC20             ;BRANCH IF NOT CORRECT.
      BR PPCDONE
;TEST DATA BUFFER:
PPCTB0: .WORD -1,-1
PPCTB1: .WORD -1,-1,-1,-1

;REPORT R0 INCORRECT.
PPC10: MOV R0,@#STMP4
1$:   MOV #PPCTB1,@#STMP3
      ERROR 1
      BR PPCDONE ; R0 BAD (BUT
; FDST)X

;REPORT RESULT INCORRECT.
PPC15: MOV #105555,@#STMP3
1$:   MOV @#PPCTB1,@#STMP4
      ERROR 1
      BR PPCDONE ; BAD DATA

;REPORT RESULT INCORRECT.
PPC20: MOV #-1,@#STMP3
1$:   MOV @#PPCTB1+2,@#STMP4
      ERROR 1
      BR PPCDONE ;(BUT GR7,FL)
;ST 357 TO 416
;INTO 417

;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
;TO THE SPURIOUS TRAP TO 4 HANDLER.
PPC25: MOV (SP),R4
      CMP R4,#PPC2+2
      BEQ 1$
      JMP @#CPSPUR
```

4829 026254 011637 001236
4830 026260 022626
4831 026262 104001
4832
4833 026264
(1) 026264 104413
(1)
(1)
(1)
(1)
4834
4835
4836
4837
(3)
(4)
(4)
(4)
(4)
(3)
(2) 026266 000004
4838
4839 026270
(1) 026270 104414
4840 026272 012700 026374
4841 026276 012701 000010
4842 026302 012720 177777
4843 026306 077103
4844 026310 012700 106653
4845 026314 012737 026342 001236
4846 026322 012737 026472 000004
4847 026330 170100
4848 026332 012700 026410
4849 026336 012710 026400
4850
4851 026342 170230
4852 026344 020027 026412
4853 026350 001021
4854 026352 023727 026400 106653
4855 026360 001024
4856 026362 023727 026410 026400
4857 026370 001030
4858 026372 000451
4859
4860
4861 026374 177777 177777
4862 026400 177777 177777 177777
026406 177777
4863 026410 177777 177777
4864
4865
4866 026414 010037 001242
4867 026420 012737 026412 001240
4868 026426 104001
4869 026430 000432
4870

```
1$: MOV (SP),@#STMP2  
CMP (SP)+,(SP)+  
2$: ERROR 1 ;(BUT FDST)+ ST634  
PPCDONE:  
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 64 DESTINATION MODES, MODE 3 (FL=0), TEST  
:*  
:* THIS IS A TEST OF DESTINATION MODE 3 USING  
:* THE STFPS INSTRUCTION  
:*  
:*****
```

```
TST64: SCOPE  
QQC1:  
LPERR ;SET UP THE LOOP ON ERROR ADDRESS.  
MOV #QQCTB0,R0 ;SET UP THE DATA BUFFER.  
MOV #10,R1  
1$: MOV #-1,(R0)+  
SOB R1,1$  
MOV #106653,R0  
MOV #QQC2,@#STMP2  
MOV #QQC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.  
LDFPS R0 ;SET UP FPS.  
MOV #QQCTB2,R0  
MOV #QQCTB1,(R0)  
QQC2: STFPS @(R0)+ ;TEST INSTRUCTION.  
CMP R0,#QQCTB2+2 ;IS R0 CORRECT?  
BNE QQC10 ;BRANCH IF NOT CORRECT.  
CMP @#QQCTB1,#106653 ;IS THE RESULT CORRECT?  
BNE QQC15 ;BRANCH IF NOT CORRECT.  
CMP @#QQCTB2,#QQCTB1 ;IS THE RESULT CORRECT?  
BNE QQC20 ;BRANCH IF NOT CORRECT.  
BR QQCDONE
```

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

```
:REPORT R0 INCORRECT.  
QQC10: MOV R0,@#STMP4  
MOV #QQCTB2+2,@#STMP3  
1$: ERROR 1 ;R0 BAD (BUT  
BR QQCDONE ; FDST)X
```



```

4871 ;REPORT RESULT INCORRECT.
4872 026432 012737 106653 001240 QQC15: MOV #106653,@#STMP3 ; ST 634
4873 026440 013737 026400 001242 MOV @#QQCTB1,@#STMP4
4874 026446 104001 1$: ERROR 1 ;BAD DATA
4875 026450 000422 BR QQCDONE
4876
4877
4878 ;REPORT RESULT INCORRECT.
4879 026452 012737 026410 001240 QQC20: MOV #QQCTB2,@#STMP3 ;(BUT FDST)
4880 026460 013737 026402 001242 MOV @#QQCTB1+2,@#STMP4
4881 026466 104001 1$: ERROR 1
4882 026470 000412 BR QQCDONE
4883
4884
4885 ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
4886 ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
4887 ;TO THE SPURIOUS TRAP TO 4 HANDLER.
4888 026472 011604 QQC25: MOV (SP),R4
4889 026474 020427 026344 CMP R4,#QQC2+2
4890 026500 001402 BEQ 1$
4891 026502 000137 036166 JMP @#CPSPUR
4892
4893 026506 011637 001236 1$: MOV (SP),@#STMP2
4894 026512 022626 CMP (SP)+,(SP)+
4895 026514 104001 2$: ERROR 1 ;(BUT FDST)+ ST634
4896
4897 026516 QQCDONE: RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) 026516 104413 ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
4898
4899
4900
4901 ;*****
(3) ;*TEST 65 DESTINATION MODES, MODE 5 (FL=0), TEST
(4) ;*
(4) ;* THIS IS A TEST OF DESTINATION MODE 5 USING
(4) ;* THE STFPS INSTRUCTION
(4) ;*
(3) ;*****
(2) 026520 000004 TST65: SCOPE
4902
4903
4904 026522 RRC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 026522 104414 MOV #RRC2,R0 ;SET UP THE DATA BUFFER.
4905 026524 012700 026630 MOV #6,R1
4906 026530 012701 000006 1$: MOV #-1,(R0)+
4907 026534 012720 177777 SOB R1,1$
4908 026540 077103 MOV #004301,R0
4909 026542 012700 004301 MOV #RRC2,@#STMP2
4910 026546 012737 026576 001236 MOV #RRC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
4911 026554 012737 026726 000004 LDFPS R0 ;SET UP FPS.
4912 026562 170100
4913 026564 012700 026646 MOV #RRC2B2+2,R0
    
```

```

4914 026570 012760 026634 177776      MOV      #RRCTB1,-2(R0)
4915
4916 026576 170250      RRC2:   STFPS @-(R0)      ;TEST INSTRUCTION.
4917 026600 020027 026644      CMP      R0,#RRCTB2      ;IS R0 CORRECT?
4918 026604 001021      BNE      RRC10           ;BRANCH IF NOT CORRECT.
4919 026606 023727 026634 004301      CMP      @#RRCTB1,#004301 ;IS THE RESULT CORRECT?
4920 026614 001024      BNE      RRC15           ;BRANCH IF NOT CORRECT.
4921 026616 023727 026644 026634      CMP      @#RRCTB2,#RRCTB1 ;IS THE RESULT CORRECT?
4922 026624 001030      BNE      RRC20           ;BRANCH IF NOT CORRECT.
4923 026626 000451      BR       RRCDONE
4924
4925      ;TEST DATA BUFFER:
4926 026630 177777 177777      RRC20:  .WORD  -1,-1
4927 026634 177777 177777 177777      RRC21:  .WORD  -1,-1,-1,-1
4928 026644 177777 177777      RRC22:  .WORD  -1,-1
4929
4930      ;REPORT R0 INCORRECT.
4931 026650 010037 001242      RRC10:  MOV      R0,@#STMP4
4932 026654 012737 026644 001240      MOV      #RRCTB2,@#STMP3
4933 026662 104001      1$:     ERROR  1          ;R0 BAD (BUT
4934 026664 000432      BR       RRCDONE          ;FDST)X
4935
4936      ;REPORT RESULT INCORRECT.
4937 026666 012737 004301 001240      RRC15:  MOV      #004301,@#STMP3          ;ST 634
4938 026674 013737 026634 001242      MOV      @#RRCTB1,@#STMP4
4939 026702 104001      1$:     ERROR  1          ;BAD DATA
4940 026704 000422      BR       RRCDONE
4941
4942      ;REPORT RESULT INCORRECT.
4943
4944 026706 012737 026644 001240      RRC20:  MOV      #RRCTB2,@#STMP3          ;BUT FDST)
4945 026714 013737 026636 001242      MOV      @#RRCTB1+2,@#STMP4
4946 026722 104001      1$:     ERROR  1          ;(BUT GR7,FL)
4947 026724 000412      BR       RRCDONE          ;ST 357 TO 416
4948      ;INTO 417
4949
4950      ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
4951      ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
4952      ;TO THE SPURIOUS TRAP TO 4 HANDLER.
4953 026726 011604      RRC25:  MOV      (SP),R4
4954 026730 020427 026600      CMP      R4,#RRC2+2
4955 026734 001402      BEQ     1$
4956 026736 000137 036166      JMP     @#CPSPUR
4957
4958 026742 011637 001236      1$:     MOV      (SP),@#STMP2
4959 026746 022626      CMP     (SP)+,(SP)+
4960 026750 104001      2$:     ERROR  1          ;(BUT FDST)+ ST634
4961
4962 026752      RRCDONE:
4963 (1) 026752 104413      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?).
    
```



```
4964
4965 (3)
4966 (4)
4967 (4)
4968 (4)
4969 (4)
4970 (3)
4971 (2) 026754 000004
4972
4973 026756 104414
4974 (1) 026756 012700 027070
4975 026760 012701 000006
4976 026764 012701 000006
4977 026770 012720 177777
4978 026774 077103
4979 026776 012700 102514
4980 027002 012737 027026 001236
4981 027010 012737 027162 000004
4982 027016 170100
4983 027020 005001
4984 027022 012700 021673
4985
4986 027026 170260 005201
4987 027032 020127 000000
4988 027036 001064
4989 027040 020027 021673
4990 027044 001017
4991 027046 023727 027074 102514
4992 027054 001022
4993 027056 023727 027076 177777
4994 027064 001026
4995 027066 000451
4996
4997 027070 177777 177777
4998 027074 177777 177777 177777
4999 027102 177777
5000
5001
5002
5003 027104 010037 001242
5004 027110 012737 021673 001240
5005 027116 104001
5006 027120 000434
5007
5008
5009
5010 027142 012737 177777 001240
```

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

SSC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #6,R1 ;SET UP THE DATA BUFFER.
1$: MOV #-1,(R0)+
SOB R1,1$
MOV #102514,R0
MOV #SSC2,@#TMP2
MOV #SSC25,@#ERRVECT ;SET UP FOR TRAPS TO VECTOR 4.
LDFPS R0 ;SET UP FPS.
CLR R1
MOV #SSCTB1-5201,R0

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

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

;REPORT R0 INCORRECT.
SSC10: MOV R0,@#TMP4
MOV #SSCTB1-5201,@#TMP3
1$: ERROR 1 ;R0 BAD
BR SSCDONE

;REPORT RESULT INCORRECT.
SSC15: MOV #102534,@#TMP3
MOV @#SSCTB1,@#TMP4
1$: ERROR 1 ;BAD DATA
BR SSCDONE

;REPORT RESULT INCORRECT.
SSC20: MOV #-1,@#TMP3
```

```
5011 027150 013737 027076 001242      MOV    @#SSCTB1+2,@#STMP4
5012 027156 104001                    1$:   ERROR 1                      ;(BUT GR7,FL)
5013 027160 000414                    BR    SSCDONE                    ;ST 357 TO 416
5014                                         ;INTO 417
5015
5016                                     ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5017                                     ;DURING EXECUTION OF THE FP> INSTRUCTION BEING TESTED, IF NOT GO
5018                                     ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5019 027162 011604                    SSC25: MOV    (SP),R4
5020 027164 020427 027030              CMP    R4,#SSC2+2
5021 027170 001402                    BEQ    1$
5022 027172 000137 036166              JMP    @#CPSPUR
5023
5024 027176 011637 001236              1$:   MOV    (SP),@#STMP2
5025 027202 022626                    CMP    (SP)+,(SP)+
5026 027204 104001                    2$:   ERROR 1                      ;(BUT FDST)+ ST634
5027 027206 000401                    BR    SSCDONE
5028
5029                                     ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
5030 027210                                SSC30:
5031 027210 104001                    1$:   ERROR 1                      ;PC NOT
5032                                         ;INCREMENTED
5033                                         ;BY 2
5034
5035 027212                                SSCDONE:
(1) 027212 104413                    RSETUP                            ;GO INITIALIZE THE FPS AND STACK; AND
(1)                                         ;SEE IF THE USER HAS EXPRESSED
(1)                                         ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                                         ;THE USER TYPED CONTROL G?).
5036
5037
5038                                     ;*****
(3)                                     ;*TEST 67      DESTINATION MODES, MODE 7 (FL=0), TEST
(4)                                     ;*
(4)                                     ;* THIS IS A TEST OF DESTINATION MODE 7 USING
(4)                                     ;* THE STFPS INSTRUCTION
(4)                                     ;*
(3)                                     ;*****
(2) 027214 000004                    TST67: SCOPE
5039
5040 027216                                TTC1:
(1) 027216 104414                    LPERR                                ;SET UP THE LOOP ON ERROR ADDRESS.
5041 027220 012700 027336              MOV    #TTCTB0,R0                    ;SET UP THE DATA BUFFER.
5042 027224 012701 000010              MOV    #10,R1
5043 027230 012720 177777              1$:   MOV    #-1,(R0)+
5044 027234 077103                    SOB    R1,1$
5045 027236 012700 103747              MOV    #103747,R0
5046 027242 012737 027274 001236      MOV    #TTC2,@#STMP2
5047 027250 012737 027434 000004      MOV    #TTC25,@#ERRVECT            ;SET UP FOR TRAPS TO VECTOR 4.
5048 027256 170100                    LDFPS  R0                            ;SET UP FPS.
5049 027260 005001                    CLR    R1
5050 027262 012700 022151              MOV    #TTCTB2-5201,R0
5051 027266 012760 027342 005201      MOV    #TTCTB1,5201(R0)
5052
5053 027274 170270 005201              TTC2:  STFPS @5201(R0)                ;TEST INSTRUCTION.
```



```

CJKDDB KEF11-A DIAG PART 2          MACY11 30A(1052) 12-MAR-80 07:25 L 10 PAGE 1-111
CJKDDB.P11 12-MAR-80 07:22          T67      DESTINATION MODES, MODE 7 (FL=0), TEST          SEQ 0128

5054 027300 022701 000000          CMP      #0,R1          ;WAS PC CORRECT AFTER EXECUTION?
5055 027304 001066          BNE      TTC30          ;BRANCH IF NOT CORRECT.
5056 027306 020027 022151          CMP      R0,#TTC2B2-5201 ;IS R0 CORRECT?
5057 027312 001021          BNE      TTC10          ;BRANCH IF NOT CORRECT.
5058 027314 023727 027342 103747  CMP      @#TTC2B1,#103747 ;IS THE RESULT CORRECT?
5059 027322 001024          BNE      TTC15          ;BRANCH IF NOT CORRECT.
5060 027324 023727 027344 177777  CMP      @#TTC2B1+2,#-1  ;IS THE RESULT CORRECT?
5061 027332 001030          BNE      TTC20          ;BRANCH IF NOT CORRECT.
5062 027334 000453          BR       TTCDONE
5063
5064          ;TEST DATA BUFFER:
5065 027336 177777 177777          TTCTB0: .WORD  -1,-1
5066 027342 177777 177777 177777  TTCTB1: .WORD  -1,-1,-1,-1
5067 027352 177777 177777          TTCTB2: .WORD  -1,-1
5068
5069          ;REPORT R0 INCORRECT.
5070 027356 010037 001242          TTC10:  MOV      R0,@#STMP4
5071 027362 012737 022151 001240  MOV      #TTC2B2-5201,@#STMP3
5072 027370 104001          1$:      ERROR   1          ;R0 BAD
5073 027372 000434          BR       TTCDONE
5074
5075
5076          ;REPORT RESULT INCORRECT.
5077 027374 012737 103747 001240  TTC15:  MOV      #103747,@#STMP3
5078 027402 013737 027342 001242  MOV      @#TTC2B1,@#STMP4
5079 027410 104001          1$:      ERROR   1          ;BAD DATA
5080 027412 000424          BR       TTCDONE
5081
5082
5083          ;REPORT RESULT INCORRECT.
5084 027414 012737 177777 001240  TTC20:  MOV      #-1,@#STMP3
5085 027422 013737 027344 001242  MOV      @#TTC2B1+2,@#STMP4
5086 027430 104001          1$:      ERROR   1          ;(BUT GR7,FL)
5087 027432 000414          BR       TTCDONE          ;ST 357 TO 416
5088          ;INTO 417
5089
5090          ;IF A TRAP TO VECTOR 4 OCCURS COME HERE TO SEE IF THE TRAP OCCURRED
5091          ;DURING EXECUTION OF THE FPP INSTRUCTION BEING TESTED, IF NOT GO
5092          ;TO THE SPURIOUS TRAP TO 4 HANDLER.
5093 027434 011604          TTC25:  MOV      (SP),R4
5094 027436 020427 027276          CMP      R4,#TTC2+2
5095 027442 001402          BEQ     1$
5096 027444 000137 036166          JMP     @#CPSPUR
5097 027450 011637 001236          1$:      MOV      (SP),@#STMP2
5098 027454 022626          CMP      (SP)+,(SP)+
5099 027456 104001          2$:      ERROR   1          ;(BUT FSDT)+ ST634
5100 027460 000401          BR       TTCDONE
5101
5102          ;REPORT PC NOT INCREMENTED BY 2 DURING EXECUTION.
5103 027462          TTC30:
5104 027462 104001          1$:      ERROR   1          ;PC NOT
5105          ;INCREMENTED
5106 027464          TTCDONE:
(1) 027464 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)          ;SEE IF THE USER HAS EXPRESSED

```

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

(1)
(1)
(1)
5107
5114
(3)
(4)
(4)
(4)
(4)
(3)
(2) 027466 000004
5115 027470
(1) 027470 104414
5116 027472 012700 000300
5117 027476 170100
5118 027500 012700 027546
5119 027504 172410
5120 027506 012737 027520 001236
5121 027514 012700 027560
5122
5123 027520 175420
5124
5125 027522 020027 027564
5126 027526 001417
5127
5128
5129 027530 010037 001242
5130 027534 012737 027564 001240
5131 027542 104001
5132 027544 000410
5133
5134 027546 000000 000000 000000
027554 000000
5135 027556 177777
5136 027560 177777 177777 177777
5137
5138 027566
(1) 027566 104413
(1)
(1)
(1)
(1)
(1)
5139
5146
(3)
(4)
(4)
(4)
(4)
(3)
(2) 027570 000004
5147
5148 027572
(1) 027572 104414
5149 027574 012700 000300

:TEST 70 DESTINATION MODES, MODE 2 (FL=1), TEST
:
:* THIS IS A TEST OF DESTINATION MODE
:* 2 USING STCOL WITH REGISTER 0
:

TST70: SCOPE
UUC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #300,R0 ;SET UP FPS.
LDFPS R0
MOV #UUCTP1,R0 ;SET UP THE ACO OPERAND.
LDD (R0),AC0
MOV #UUC2,@#STMP2
MOV #UUCBF0,R0

UUC2: STCDL ACO,(R0)+ ;TEST INSTRUCTION.
CMP R0,#UUCBF0+4 ;IS R0 CORRECT?
BEQ UUCDONE ;BRANCH IF CORRECT.

;REPORT R0 INCORRECT.
UUC3: MOV R0,@#STMP4
MOV #UUCBF0+4,@#STMP3
1\$: ERROR 1 ;R0 NOT INCR BY 4
BR UUCDONE

;TEST DATA BUFFER:
UUCTP1: .WORD 0,0,0,0
-1
UUCBF0: .WORD -1,-1,-1

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

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

TST71: SCOPE
VVC1: LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
MOV #300,R0 ;SET UP FPS.


```

5150 027600 170100          LDFPS  RO
5151 027602 012700 027650  MOV    #VVCTP1,RO      ;SET UP THE ACO OPERAND.
5152 027606 172410          LDD    (RO),AC0
5153 027610 012737 027622 001236  MOV    #VVC2,@#STMP2
5154 027616 012700 027666  MOV    #VVCBF0+4,RO
5155
5156 027622 175440          VVC2:  STCDL  ACO,-(RO)      ;TEST INSTRUCTION.
5157
5158 027624 020027 027662  CMP    RO,#VVCBF0      ;IS RO CORRECT?
5159 027630 001417          BEQ    VVCDONE
5160
5161          ;REPORT RO INCORRECT.
5162 027632 010037 001242  VVC3:  MOV    RO,@#STMP4
5163 027636 012737 027662 001240  MOV    #VVCBF0,@#STMP3
5164 027644 104001          1$:    ERROR  1          ;RO NOT DECR BY 4
5165 027646 000410          BR     VVCDONE
5166          ;TEST DATA BUFFER:
5167 027650 000000 000000 000000  VVCTP1: .WORD  0,0,0,0
5168 027656 000000
5168 027660 177777          -1
5169 027662 177777 177777 177777  VVCBF0: .WORD  -1,-1,-1
5170
5171 027670          VVCDONE:
(1) 027670 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)          ;SEE IF THE USER HAS EXPRESSED
(1)          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)          ;THE USER TYPED CONTROL G?).
5172
5182          ;*****
(3)          ;*TEST 72      STCDI AND STCDL TEST
(4)          ;*
(4)          ;* THIS IS A TEST OF THE STCDI AND
(4)          ;* STCDL INSTRUCTIONS. NOTE THAT A
(4)          ;* SUBROUTINE, STCSUB, IS USED TO
(4)          ;* SET UP THE OPERANDS, EXECUTE THE STC
(4)          ;* INSTRUCTION AND CHECK THE RESULT.
(4)          ;*
(3)          ;*****
(2) 027672 000004          TST72:  SCOPE
5183          ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
5184          WWC1:
5185 027674          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
(1) 027674 104414          JSR    PC,@#STCSUB      ;GO EXECUTE THE INSTRUCTION.
5186 027676 004737 031042          .WORD  2000,0,0,0      ;ACO OPERAND.
5187 027702 020000 000000 000000  1$:    .WORD
5188 027710 000000          2$:    .WORD  0,0          ;EXPECTED RESULT.
5189 027716 177777 177777          3$:    .WORD  -1,-1      ;ERROR RES.
5190 027722 040300          4$:    40300          ;FPS BEFORE EXECUTION.
5191 027724 040304          40304          ;FPS AFTER EXECUTION.
5192 027726 140304          140304          ;ANTICIPATED ERRONEOUS FPS.
5193 027730 177777          -1          ;REPORT RESULT INCORRECT.
5194 027732 104001          5$:    ERROR  1          ;RESULT INCORP.
5195 027734 000401          BR     6$
5196 027736 104001          ERROR  1          ;EITHER (BUT FLAG)
    
```

```

5197 027740          6$:                               ;ST 662
5198                                     ;OR CLEAR FLAG
5199                                     ;ST 774
5200
5201          ;EXP=0 (OCT)  FL=1    FIC=0
5202 027740          WWC2:
(1) 027740 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5203 027742 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5204 027746 040000 000000 000000 1$: .WORD          40000,0,0,0 ;AC ;ACO OPERAND.
027754 000000
5205 027756 000000 000000          2$: .WORD          0,0 ;EXPECTED RESULT.
5206 027762 177777 177777          3$: .WORD          -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5207 027766 040313          4$: 40313 ;FPS BEFORE EXECUTION.
5208 027770 040304          40304 ;FPS AFTER EXECUTION.
5209 027772 140304          140304 ;ANTICIPATED ERRONEOUS FPS.
5210 027774 177777          -1 ;EXPECTED FEC.
5211 027776 104001          5$: ERROR          1 ;REPORT RESULT INCORRECT.
5212 030000 000401          BR          6$
5213 030002 104001          ERROR          1 ;REPORT FPS INCORRECT.
5214 030004          6$:
5215
5216          ;EXP=37 (OCT)  FL=1    FIC=1
5217 030004          WWC4:
(1) 030004 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5218 030006 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5219 030012 047667 075757 157737 1$: .WORD          47667,75757,157737,167773 ;ACO OPERAND.
030020 167773
5220 030022 055675 173757          2$: .WORD          55675,173757 ;EXPECTED RESULT.
5221 030026 122102 004021          3$: .WORD          122102,004021 ;ANTICIPATED ERRONEOUS RESULT.
5222 030032 040717          4$: 40717 ;FPS BEFORE EXECUTION.
5223 030034 040700          40700 ;FPS AFTER EXECUTION.
5224 030036 140705          140705 ;ANTICIPATED ERRONEOUS FPS.
5225 030040 177777          -1 ;EXPECTED FEC.
5226 030042 104001          5$: ERROR          1 ;(BUT ENBT) ST 632
5227 030044 000401          BR          6$
5228 030046 104001          ERROR          1 ;REPORT FPS INCORRECT.
5229 030050          6$:
5230
5231          ;EXP=40 (OCT)  FL=1    FIC=1
5232 030050          WWC5:
(1) 030050 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5233 030052 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5234 030056 050000 000000 000000 1$: .WORD          50000,0,0,0 ;ACO OPERAND.
030064 000000
5235 030066 000000 000000          2$: .WORD          0,0 ;EXPECTED RESULT.
5236 030072 177777 177777          3$: .WORD          -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5237 030076 040700          4$: 40700 ;FPS BEFORE EXECUTION.
5238 030100 140705          140705 ;FPS AFTER EXECUTION.
5239 030102 040705          40705 ;ANTICIPATED ERRONEOUS FPS.
5240 030104 000006          6 ;EXPECTED FEC.
5241 030106 104001          5$: ERROR          1 ;REPORT RESULT INCORRECT.
5242 030110 000401          BR          6$
5243 030112 104001          ERROR          1 ;(BUT FIC) ST 004 ;REPORT FPS INCORRECT.
5244                                     ;TO 305 INTO
5245 030114          6$:                                     ;315
5246
    
```



```

5247      ;EXP=40 (OCT)   FL=1   FIC=0
5248      WWC6:
(1) 030114 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5249 030116 004737 031042 JSR      PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5250 030122 050000 000000 000000 1$: .WORD 50000,0,0,0 ;ACO OPERAND.
030130 000000
5251 030132 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
5252 030136 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5253 030142 040312 4$: 40312 ;FPS BEFORE EXECUTION.
5254 030144 040305 40305 ;FPS AFTER EXECUTION.
5255 030146 140305 140305 ;ANTICIPATED ERRONEOUS FPS.
5256 030150 177777 -1 ;EXPECTED FEC.
5257 030152 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5258 030154 000401 BR 6$
5259 030156 104001 ERROR 1 ;(BUT FIC) ST 004 TO
5260 030160 6$: ;315 INTO 305
5261
5262      ;EXP=30 (OCT)   FL=1   FIC=1
5263      WWC7:
(1) 030160 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5264 030162 004737 031042 JSR      PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5265 030166 046000 000001 000000 1$: .WORD 46000,1,0,0 ;ACO OPERAND.
030174 000000
5266 030176 000200 000001 2$: .WORD 200,1 ;EXPECTED RESULT.
5267 030202 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5268 030206 040700 4$: 40700 ;FPS BEFORE EXECUTION.
5269 030210 040700 40700 ;FPS AFTER EXECUTION.
5270 030212 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
5271 030214 177777 -1 ;EXPECTED FEC.
5272 030216 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5273 030220 000401 BR 6$
5274 030222 104001 ERROR 1 ;REPORT FPS INCORRECT.
5275 030224 6$:
5276
5277      ;EXP=27 (OCT)   FL=1   FIC=1
5278      WWC8:
(1) 030224 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5279 030226 004737 031042 JSR      PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5280 030232 045600 000001 000000 1$: .WORD 45600,1,0,0 ;ACO OPERAND.
030240 000000
5281
5282 030242 000100 000000 2$: .WORD 100,0 ;EXPECTED RESULT.
5283 030246 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5284 030252 040707 4$: 40707 ;FPS BEFORE EXECUTION.
5285 030254 040700 40700 ;FPS AFTER EXECUTION.
5286 030256 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
5287 030260 177777 -1 ;EXPECTED FEC.
5288 030262 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5289 030264 000401 BR 6$
5290 030266 104001 ERROR 1 ;REPORT FPS INCORRECT.
5291 030270 6$:
5292
5293      ;EXP=17 (OCT)   FL=0   FIC=1
5294      WWC9:
(1) 030270 104414      LPERR      ;SET UP THE LOOP ON ERROR ADDRESS.
5295 030272 004737 031042 JSR      PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
    
```

```

5296 030276 043600 000000 000000 1$: .WORD 43600,0,0,0 ;ACO OPERAND.
      030304 000000
5297 030306 040000 177777 2$: .WORD 40000,-1 ;EXPECTED RESULT.
5298 030312 000000 177777 3$: .WORD 0,-1 ;ANTICIPATED ERRONEOUS RESULT.
5299 030316 040600 4$: 40600 ;FPS BEFORE EXECUTION.
5300 030320 040600 40600 ;FPS AFTER EXECUTION.
5301 030322 140604 140604 ;ANTICIPATED ERRONEOUS FPS.
5302 030324 177777 -1 ;EXPECTED FEC.
5303 030326 104001 5$: ERROR 1 ;BAD CONSTANT ST 066
5304 030330 000401 BR 6$
5305 030332 104001 ERROR 1 ;REPORT FPS INCORRECT.
5306 030334 6$:
5307
5308 ;EXP=20 (OCT) FL=0 FIC=1
5309 030334 WWC10:
      (1) 030334 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5310 030336 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5311 030342 044000 000000 000000 1$: .WORD 44000,0,0,0 ;ACO OPERAND.
      030350 000000
5312 030352 000000 177777 2$: .WORD 0,-1 ;EXPECTED RESULT.
5313 030356 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5314 030362 040600 4$: 40600 ;FPS BEFORE EXECUTION.
5315 030364 140605 140605 ;FPS AFTER EXECUTION.
5316 030366 040600 40600 ;ANTICIPATED ERRONEOUS FPS.
5317 030370 000006 6 ;EXPECTED FEC.
5318 030372 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5319 030374 000401 BR 6$
5320 030376 104001 ERROR 1 ;BAD CONSTANT ST 066
5321 030400 6$:
5322
5323 ;EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1
5324 030400 WWC11:
      (1) 030400 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5325 030402 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5326 030406 142000 000000 000000 1$: .WORD 142000,0,0,0 ;ACO OPERAND.
      030414 000000
5327 030416 177600 177777 2$: .WORD 177600,-1 ;EXPECTED RESULT.
5328 030422 000200 000000 3$: .WORD 200,0 ;ANTICIPATED ERRONEOUS RESULT.
5329 030426 040600 4$: 40600 ;FPS BEFORE EXECUTION.
5330 030430 040610 40610 ;FPS AFTER EXECUTION.
5331 030432 040600 40600 ;ANTICIPATED ERRONEOUS FPS.
5332 030434 177777 -1 ;EXPECTED FEC.
5333 030436 104001 5$: ERROR 1 ;(BUT ENBT) ST 632
5334 030440 000401 BR 6$
5335 030442 104001 ERROR 1 ;(SET FN) ST 473
5336 030444 6$:
5337
5338 ;EXP=37 (OCT), FL=1, FIC=1, AC NEG.
5339 030444 WWC12:
      (1) 030444 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5340 030446 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5341 030452 147600 000000 000000 1$: .WORD 147600,0,0,0 ;ACO OPERAND.
      030460 000000
5342 030462 140000 000000 2$: .WORD 140000,0 ;EXPECTED RESULT.
5343 030466 137777 000000 3$: .WORD 137777,0 ;ANTICIPATED ERRONEOUS RESULT.
5344 030472 040700 4$: 40700 ;FPS BEFORE EXECUTION.

```



```

5345 030474 040710 40710 ;FPS AFTER EXECUTION.
5346 030476 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
5347 030500 177777 -1 ;EXPECTED FEC.
5348 030502 104001 5$: ERROR 1 ;(BUT COUT) ST 375
5349 030504 000401 BR 6$ ;ST 275 TO 074
5350 030506 104001 ERROR 1 ;INTO 274
5351 030510
5352
5353 ;EXP=37 (OCT), FL=1, FIC=1, AC NEG
5354 030510 WWC13:
(1) 030510 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5355 030512 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5356 030516 147600 000000 001000 1$: .WORD 147600,0,1000,0 ;ACO OPERAND.
030524 000000
5357 030526 137777 177777 2$: .WORD 137777,177777 ;EXPECTED RESULT.
5358 030532 140000 177777 3$: .WORD 140000,177777 ;ANTICIPATED ERRONEOUS RESULT.
5359 030536 040707 4$: 40707 ;FPS BEFORE EXECUTION.
5360 030540 040710 40710 ;FPS AFTER EXECUTION.
5361 030542 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
5362 030544 177777 -1 ;EXPECTED FEC.
5363 030546 104001 5$: ERROR 1 ;(BUT COUT) ST 375
5364 030550 000401 BR 6$ ;TO 274 INTO 074
5365 030552 104001 ERROR 1 ;REPORT FPS INCORRECT.
5366 030554
5367
5368 ;EXP=41 (OCT), AC NEG, FL=1, FIC=1
5369 030554 WWC14:
(1) 030554 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5370 030556 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5371 030562 150200 000000 000000 1$: .WORD 150200,0,0,0 ;ACO OPERAND.
030570 000000
5372 030572 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
5373 030576 177777 177777 3$: .WORD -1,-1 ;ANTICIPATED ERRONEOUS RESULT.
5374 030602 040700 4$: 40700 ;FPS BEFORE EXECUTION.
5375 030604 140705 140705 ;FPS AFTER EXECUTION.
5376 030606 177777 -1 ;ANTICIPATED ERRONEOUS FPS.
5377 030610 000006 6 ;EXPECTED FEC.
5378 030612 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5379 030614 000401 BR 6$
5380 030616 104001 ERROR 1 ;(BUT EZBT) ST 377
5381 030620
5382 ;EXP=40 (OCT), AC NEG, FL=1, FIC=1
5383 030620 WWC15:
(1) 030620 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5384 030622 004737 031042 JSR PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5385 030626 150000 000001 000000 1$: .WORD 150000,1,0,0 ;ACO OPERAND.
030634 000000
5386 030636 000000 000000 2$: .WORD 0,0 ;EXPECTED RESULT.
5387 030642 100000 177600 3$: .WORD 100000,-200 ;ANTICIPATED ERRONEOUS RESULT.
5388
5389 030646 040700 4$: 40700 ;FPS BEFORE EXECUTION.
5390 030650 140705 140705 ;FPS AFTER EXECUTION.
5391 030652 040700 40700 ;ANTICIPATED ERRONEOUS FPS.
5392 030654 000006 6 ;EXPECTED FEC.
5393 030656 104001 5$: ERROR 1 ;(BUT COUT) ST 360
5394 030660 000401 BR 6$ ;TO 654 INTO 454
    
```

```
5395 030662 104001          ERROR 1          ;REPORT FPS INCORRECT.
5396 030664          6$:
5397
5398          ;EXP=40, AC NEGATIVE, FL=1, FIC=1
5399 030664          WWC16:
(1) 030664 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5400 030666 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5401 030672 150001 000000 000000 1$: .WORD 150001,0,0,0 ;ACO OPERAND.
5402 030700 000000          2$: .WORD 0,0          ;EXPECTED RESULT.
5403 030706 077400 000000          3$: .WORD 77400,0       ;ANTICIPATED ERRONEOUS RESULT.
5404 030712 040700          4$: 40700          ;FPS BEFORE EXECUTION.
5405 030714 140705          140705          ;FPS AFTER EXECUTION.
5406 030716 177777          -1          ;ANTICIPATED ERRONEOUS FPS.
5407 030720 000006          6          ;EXPECTED FEC.
5408 030722 104001          5$: ERROR 1          ;REPORT RESULT INCORRECT.
5409 030724 000401          BR 6$
5410 030726 104001          ERROR 1          ;REPORT FPS INCORRECT.
5411 030730          6$:
5412
5413
5414          ;EXP 40 (OCT), AC MOST NEG LONG INT, FL=1
5415          ;FIC=1
5416 030730          WWC17:
(1) 030730 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5417 030732 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5418 030736 150000 000000 000000 1$: .WORD 150000,0,0,0 ;ACO OPERAND.
5419 030744 000000          2$: .WORD 100000,0       ;EXPECTED RESULT.
5420 030752 000000 000000          3$: .WORD 0,0          ;ANTICIPATED ERRONEOUS RESULT.
5421 030756 040700          4$: 40700          ;FPS BEFORE EXECUTION.
5422 030760 040710          40710          ;FPS AFTER EXECUTION.
5423 030762 140705          140705          ;ANTICIPATED ERRONEOUS FPS.
5424 030764 177777          -1          ;EXPECTED FEC.
5425 030766 104001          5$: ERROR 1          ;(BUT NBIT) ST 654
5426 030770 000401          BR 6$          ;OR (BUT COUT) ST 454
5427 030772 104001          ERROR 1          ;REPORT FPS INCORRECT.
5428 030774          6$:
5429
5430          ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
5431
5432 030774          WWC18:
(1) 030774 104414          LPERR          ;SET UP THE LOOP ON ERROR ADDRESS.
5433 030776 004737 031042          JSR          PC,@#STCSUB ;GO EXECUTE THE INSTRUCTION.
5434 031002 144000 000001 000000 1$: .WORD 144000,1,0,0 ;ACO OPERAND.
5435 031010 000000          2$: .WORD 100000,-1       ;EXPECTED RESULT.
5436 031016 100000 177400          3$: .WORD 100000,177400 ;ANTICIPATED ERRONEOUS RESULT.
5437 031022 040600          4$: 40600          ;FPS BEFORE EXECUTION.
5438 031024 040610          40610          ;FPS AFTER EXECUTION.
5439 031026 140605          140605          ;ANTICIPATED ERRONEOUS FPS.
5440 031030 177777          -1          ;EXPECTED FEC.
5441 031032 104001          5$: ERROR 1          ;(BUT FL) ST 633
5442 031034 000401          BR 6$          ;TO 655 INTO 654
5443 031036 104001          ERROR 1          ;REPORT FPS INCORRECT.
5444
```


5445 031040 000534

6\$: BR WWCONE

5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500

031042 012601
 031044 012700 000200
 031050 170100
 031052 010100
 031054 172410
 031056 012702 031322
 031062 012700 000004
 031066 012722 177777
 031072 077003
 031074 016100 000020
 031100 170100
 031102 012737 031114 001236
 031110 012700 031322
 031114 175410
 031116 170204
 031120 170305
 031122 010102
 031124 010237 001240
 031130 062702 000010
 031134 010237 001244
 031140 012737 031322 001242

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

```

:
: JSR PC,@#STCSUB
: ACARG: .WORD X,X,X,X ;AC OPERAND
: RES: .WORD X,X ;EXPECTED RESULT
: ERRES: .WORD X,X ;ERROR RESULT
: FPSB: .WORD X ;FPS BEFORE EXECUTION
: FPSA: .WORD X ;FPS AFTER EXECUTION
: ERFPS: .WORD X ;ERROR FPS.
: FEC: .WORD X ;EXPECTED FEC
: ERR1: ERROR 1;DATA ERROR.
: BR CONT
: ERR2: ERROR 1;FPS ERROR.
: CONT: ;RETURN ADDRESS
  
```

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

```

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

    STFPS R4 ;GET THE FPS.
    STST R5 ;GET THE FEC.
    MOV R1,R2
    MOV R2,@#STMP3
    ADD #10,R2
    MOV R2,@#STMP5
    MOV #STCIBF,@#STMP4
  
```

```

5501 031146 010437 001250      MOV      R4,@#STMP7
5502 031152 016137 000022 001252  MOV      22(R1),@#STMP10
5503 031160 010102      MOV      R1,R2
5504 031162 062702 000010      ADD      #10,R2
5505 031166 012700 031322      MOV      #STCIBF,R0      ;SEE IF THE RESULT IS CORRECT.
5506 031172 012703 000002      MOV      #2,R3
5507 031176 022022      3$:      CMP      (R0)+,(R2)+
5508 031200 001014      BNE      15$
5509 031202 077303      SOB      R3,3$
5510 031204 016102 000022      MOV      22(R1),R2
5511 031210 020204      CMP      R2,R4      ;SEE IF THE FPS IS CORRECT.
5512 031212 001025      BNE      20$      ;BRANCH IF INCORRECT.
5513 031214 005702      TST      R2
5514 031216 100003      BPL      4$
5515 031220 026105 000026      CMP      26(R1),R5      ;SEE IF THE FEC IS CORRECT.
5516 031224 001027      BNE      25$      ;BRANCH IF INCORRECT.
5517
5518 031226 000161 000036      4$:      JMP      36(R1)      ;RETURN.
5519      ;DATA ERROR:
5520      ;SEE IF THE FAILURE WAS ANTICIPATED.
5521 031232 010102      15$:      MOV      R1,R2
5522 031234 062702 000014      ADD      #14,R2
5523 031240 012700 031322      MOV      #STCIBF,R0
5524 031244 012703 000002      MOV      #2,R3
5525 031250 022022      16$:      CMP      (R0)+,(R2)+
5526 031252 001003      BNE      17$
5527 031254 077303      SOB      R3,16$
5528 031256 000161 000030      JMP      30(R1)
5529 031262
5530      ;FAILURE WAS NOT ANTICIPATED SO REPORT INCORRECT RESULT HERE.
5531 031262 104001      18$:      ERROR   1      ;DATA BAD
5532 031264 000760      BR      4$
5533
5534      ;FPS INCORRECT, SO SEE IF FAILURE WAS ANTICIPATED.
5535 031266 020461 000024      20$:      CMP      R4,24(R1)
5536 031272 001002      BNE      21$
5537 031274 000161 000034      JMP      34(R1)
5538 031300
5539      ;NOT ANTICIPATED SO REPORT BAD FPS HERE.
5540 031300 104001      21$:
5541 031302 000751      22$:      ERROR   1      ;FPS BAD
5542      BR      4$
5543
5544 031304 016137 000026 001256      ;REPORT INCORRECT FEC.
5545 031312 010537 001254      25$:      MOV      26(R1),@#STMP12
5546 031316 104001      MOV      R5,@#STMP11
5547 031320 000742      26$:      ERROR   1
5548      BR      4$
5549
5550 031322 177777 177777 177777      ;DATA BUFFER:
5551 031330 177777      STCIBF: .WORD  -1,-1,-1,-1
5552
5553      WWC DONE:
5554      RSETUP
5555      ;GO INITIALIZE THE FPS AND STACK; AND
5556      ;SEE IF THE USER HAS EXPRESSED
5557      ;THE DESIRE TO CHANGE THE SOFTWARE
    
```


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

5553
5554
5562 :*****

(3) :*TEST 73 STCFL AND STCFI TEST
(4) :*
(4) :* THIS IS A TEST OF STCFL AND STCFI. IT
(4) :* MAKES USE OF THE SAME SUBROUTINE, STCSUB,
(4) :* WHICH WAS USED TO TEST STCDL AND STCDI.
(4) :*
(3) :*****

(2) 031334 000004 TST73: SCOPE

5563 :EXPONENT=37, FL=1
5564 :XXC1:

5566 031336 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
(1) 031336 104414 JSR PC,@#STCSUB :GO EXECUTE THE INSTRUCTION.
5567 031340 004737 031042 :AC0 OPERAND.
5568 031344 047777 177777 177777 1\$: .WORD 47777,-1,-1,-1
031352 177777
5569 031354 077777 177600 2\$: .WORD 77777,177600 :EXPECTED RESULT.
5570 031360 077777 177777 3\$: .WORD 77777,177777 :ANTICIPATED ERRONEOUS RESULT.
5571 031364 040100 4\$: 40100 :FPS BEFORE EXECUTION.
5572 031366 040100 :FPS AFTER EXECUTION.
5573 031370 177777 -1 :ANTICIPATED ERRONEOUS FPS.
5574 031372 177777 -1 :EXPECTED FEC.
5575 031374 104001 5\$: ERROR 1 ;X11(1,0)+0 ST 773X
5576 031376 000401 BR 6\$
5577 031400 104001 ERROR 1 :REPORT FPS INCORRECT.
5578 031402
5579

5580 031402 XXCDONE:
(1) 031402 104413 RSETUP

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

5581
5582 :*****
5589 :*TEST 74 STEXP TEST

(3) :*
(4) :* THIS IS A TEST OF THE STEXP
(4) :* INSTRUCTION
(4) :*
(3) :*****

(2) 031404 000004 TST74: SCOPE

5590 : EXP = 100 (EXCESS 200)
5591 :YYC1:

5592 031406 LPERR :SET UP THE LOOP ON ERROR ADDRESS.
(1) 031406 104414 JSR PC,@#STXSUB
5593 031410 004737 031674 :AC
5594 031414 020000 000000 000000 1\$: .WORD 20000,0,0,0
031422 000000
5595 031424 177700 2\$: -100 :EXP RES

```
5596 031426 052525 3$: 52525 ;ERROR EXP.
5597 031430 040000 4$: 40000 ;FPSB
5598 031432 040010 ;40010 ;FPSA
5599 031434 040000 ;40000 ;ERROR FPS
5600 031436 104001 5$: ERROR 1 ;BAD EXP
5601 031440 000401 BR 6$
5602 031442 104001 ERROR 1 ;+(BUT ENBT) ST 376
5603 031444 6$:
5604
5605 ; EXP = 200 (EXCESS 200)
5606 031444 YYC2:
(1) 031444 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5607 031446 004737 031674 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
5608 031452 040000 000000 000000 1$: .WORD 40000,0,0,0 ;ACO OPERAND.
5609 031462 000000 2$: 0 ;EXPECTED EXPONENT RESULT.
5610 031464 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
5611 031466 040000 4$: 40000 ;FPS BEFORE EXECUTION.
5612 031470 040004 ;40004 ;FPS AFTER EXECUTION.
5613 031472 040000 ;40000 ;ANTICIPATED ERRONEOUS FPS.
5614 031474 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5615 031476 000401 BR 6$
5616 031500 104001 ; ERROR 1 ;(BUT EZBT) ST 071
5617 ; ;TO 072 INT 272
5618 031502 6$:
5619
5620 ; EXP = 201 (EXCESS 200)
5621
5622 031502 YYC3:
(1) 031502 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5623 031504 004737 031674 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
5624 031510 040200 000000 000000 1$: .WORD 40200,0,0,0 ;ACO OPERAND.
5625 031516 000000 2$: 1 ;EXPECTED EXPONENT RESULT.
5626 031522 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
5627 031524 040000 4$: 40000 ;FPS BEFORE EXECUTION.
5628 031526 040000 ;40000 ;FPS AFTER EXECUTION.
5629 031530 040004 ;40004 ;ANTICIPATED ERRONEOUS FPS.
5630 031532 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
5631 031534 000401 BR 6$
5632 031536 104001 ; ERROR 1 ;(BUT EZBT) ST 071
5633 031540 6$: ;TO 272 INTO 072
5634
5635 ; EXP = 375 (EXCESS 200)
5636
5637 031540 YYC4:
(1) 031540 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5638 031542 004737 031674 JSR PC,@#STXSUB ;GO EXECUTE THE INSTRUCTION.
5639 031546 077200 000000 000000 1$: .WORD 77200,0,0,0 ;ACO OPERAND.
5640 031556 000175 2$: 175 ;EXPECTED EXPONENT RESULT.
5641 031560 052525 3$: 52525 ;ANTICIPATED ERRONEOUS RESULT.
5642 031562 040000 4$: 40000 ;FPS BEFORE EXECUTION.
5643 031564 040000 ;40000 ;FPS AFTER EXECUTION.
5644 031566 040010 ;40010 ;ANTICIPATED ERRONEOUS FPS.
5645 031570 104001 5$: ERROR 1 ;REPORT RESULT INCORRECT.
```



```

5646 031572 000401          BR      6$
5647 031574 104001          ERROR   1          ;(BUT ENBT) ST 376
5648 031576                6$:                ;TO 471 INTO 071
5649
5650                ; EXP = 1 (EXCESS 200)
5651
5652 031576                YYC5:
  (1) 031576 104414          LPERR
5653 031600 004737 031674 000000 JSR      PC,@#STXSUB ;SET UP THE LOOP ON ERROR ADDRESS.
5654 031604 000200 000000 1$:  .WORD  200,0,0,0 ;GO EXECUTE THE INSTRUCTION.
                                       ;ACO OPERAND.
5655 031614 177601          2$:      -177          ;EXPECTED EXPONENT RESULT.
5656 031616 052525          3$:      52525          ;ANTICIPATED ERRONEOUS RESULT.
5657 031620 040000          4$:      40000          ;FPS BEFORE EXECUTION.
5658 031622 040010          ;FPS AFTER EXECUTION.
5659 031624 040000          ;ANTICIPATED ERRONEOUS FPS.
5660 031626 104001          5$:      ERROR      1          ;REPORT RESULT INCORRECT.
5661 031630 000401          BR      6$
5662 031632 104001          ERROR   1          ;REPORT FPS INCORRECT.
5663 031634
5664
5665                ; EXP = 156 (EXCESS 200)
5666
5667 031634                YYC6:
  (1) 031634 104414          LPERR
5668 031636 004737 031674 000000 JSR      PC,@#STXSUB ;SET UP THE LOOP ON ERROR ADDRESS.
5669 031642 033400 000000 1$:  .WORD  33400,0,0,0 ;GO EXECUTE THE INSTRUCTION.
                                       ;ACO OPERAND.
5670 031652 177756          2$:      -22          ;EXPECTED EXPONENT RESULT.
5671 031654 052525          3$:      52525          ;ANTICIPATED ERRONEOUS RESULT.
5672 031656 047707          4$:      47707          ;FPS BEFORE EXECUTION.
5673 031660 047710          ;FPS AFTER EXECUTION.
5674 031662 177777          ;ANTICIPATED ERRONEOUS FPS.
5675 031664 104001          5$:      ERROR      1          ;REPORT RESULT INCORRECT.
5676 031666 000401          BR      6$
5677 031670 104001          ERROR   1          ;REPORT FPS INCORRECT.
5678
5679 031672 000510          6$:      BR      YYCDONE
5680
5681                ;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
5682                ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
5683                ;TO IT IS MADE THUS:
5684                :
5685                :
5686                :
5687                :
5688                :
5689                :
5690                :
5691                :
5692                :
5693                :
5694                :
5695                :
5696                :
5697                :

```

```

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  1;DATA ERROR.
        BR      CONT
ERR2:   ERROR  1;FPS ERROR.
CONT:   ;RETURN ADDRESS

```

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

5698
 5699
 5700
 5701
 5702
 5703
 5704
 5705
 5706
 5707
 5708
 5709
 5710
 5711
 5712 031674 012601
 5713 031676 010102
 5714 031700 010237 001240
 5715 031704 062702 000010
 5716 031710 012237 001244
 5717 031714 012737 031762 001236
 5718 031722 012737 123456 032102
 5719 031730 012737 076543 032104
 5720 031736 012700 000200
 5721 031742 170100
 5722 031744 010100
 5723 031746 172410
 5724 031750 016100 000016
 5725 031754 170100
 5726 031756 012700 032102
 5727 031762 175010
 5728 031764 170204
 5729 031766 010437 001250
 5730 031772 016137 000016 001252
 5731 032000 013737 032102 001242
 5732 032006 026137 000010 032102
 5733 032014 001411
 5734 032016 026137 000012 032102
 5735 032024 001002
 5736 032026 000161 000022
 5737
 5738
 5739 032032
 5740 032032 104001
 5741 032034 000161 000030
 5742
 5743 032040 020461 000016
 5744 032044 001407
 5745 032046 020461 000020
 5746 032052 001002
 5747 032054 000161 000026
 5748
 5749
 5750 032060
 5751 032060 104001
 5752 032062 000764
 5753

```

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

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

;IF NOT ANTICIPATED REPORT ERROR HERE.
2$:
3$:     ERROR   1
4$:     JMP     30(R1)      ;EXP BAD

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

;FPS ERROR WAS NOT ANTICIPATED SO REPORT ERROR HERE.
6$:
7$:     ERROR   1
        BR     4$         ;FPS BAD
  
```



```

5754 ;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
5755 032064 022737 076543 032104 10$: CMP #76543,@#STXBF+2
5756 032072 001760 BEQ 4$
5757 032074 104001 11$: ERROR 1 ;FDFL+0 ST 347X
5758 032076 000756 BR 4$
5759
5760 032100 177777 STXBF: -1
5761 032102 177777 177777 177777 STXBF: .WORD -1,-1,-1,-1,-1
5762 032110 177777 177777
5763 032114 YYCDONE:
(1) 032114 104413 RSETUP ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
5764
5775 :*****
(3) ;*TEST 75 STST TEST
(4) ;*
(4) ;* THIS IS A TEST OF THE STST
(4) ;* INSTRUCTION. FIRST AN ILLEGAL FPS OP CODE
(4) ;* (INSTRUCTION) IS USED TO ENTER AN
(4) ;* ERROR CONDITION IN THE FEC AND
(4) ;* FEA. THE STST IS EXECUTED AND
(4) ;* THE FEC AND FEA ARE CHECKED
(4) ;*
(3) :*****
(2) 032116 000004 TST75: SCOPE
5776
5777 032120 ZC1:
(1) 032120 104414 LPERR ;SET UP THE LOOP ON ERROR ADDRESS.
5778 032122 012700 040000 MOV #40000,R0 ;SET FPS. FID=1.
5779 032126 170100 LDFPS R0
5780
5781 032130 170003 ZC2: .WORD 170003 ;ILLEGAL FPP
5782 ;OP CODE
5783 032132 012700 032306 MOV #ZC1,R0 ;SET UP THE OUTPUT BUFFER.
5784 032136 012710 177777 MOV #-1,(R0)
5785 032142 012760 177777 000002 MOV #-1,2(R0)
5786 032150 012737 032156 001236 ZC3: STST (R0) ;GET FEC AND
5787 032156 170310 ;FEA
5788 ;GET FPS.
5789 032160 170204 STFPS R4
5790 032162 012700 032306 MOV #ZC1,R0
5791 032166 011037 001240 MOV (R0),@#STMP3
5792 032172 016037 000002 001242 MOV 2(R0),@#STMP4
5793 032200 012737 000002 001244 MOV #2,@#STMP5
5794 032206 012737 032130 001246 MOV #ZC2,@#STMP6
5795 032214 010437 001250 MOV R4,@#STMP7
5796 032220 012737 140000 001252 MOV #140000,@#STMP10
5797
5798 032226 022710 000002 CMP #2,(R0) ;SEE IF FEC IS CORRECT.
5799 032232 001010 BNE ZC5 ;BRANCH IF INCORRECT.
5800 032234 022760 032130 000002 CMP #ZC2,2(R0) ;SEE IF FEA, ADDRESS, IS CORRECT.
5801 032242 001006 BNE ZC10 ;BRANCH IF INCORRECT.
    
```

```

5802 032244 022704 140000      CMP      #140000,R4      ;SEE IF FPS IS CORRECT.
5803 032250 001013      BNE      ZZC15          ;BRANCH IF INCORRECT.
5804 032252 000422      BR       ZZCDONE
5805
5806      ;REPORT FEC INCORRECT
5807 032254      ZZC5:
5808 032254 104001      1$:      ERROR      1          ;STST BAD
5809 032256 000420      BR       ZZCDONE          ;FECX
5810
5811      ;REPORT FEA INCORRECT
5812 032260 022760 177777 000002 ZZC10:  CMP      #-1,2(R0)
5813 032266 001402      BEQ      ZZC12
5814 032270 104001      1$:      ERROR      1          ;STST BAD FEA
5815 032272 000412      BR       ZZCDONE
5816 032274      ZZC12:
5817 032274 104001      1$:      ERROR      1          ;SET FD FL ST 636
5818 032276 000410      BR       ZZCDONE
5819
5820      ;REPORT FPS INCORRECT
5821 032300      ZZC15:
5822 032300 104001      1$:      ERROR      1          ;FPS X AFTER ST ST
5823 032302 000406      BR       ZZCDONE
5824
5825      ;DATA BUFFER:
5826 032304 177777      -1
5827 032306 177777 177777 177777 ZZCBF: .WORD  -1,-1,-1,-1
5828 032314 177777      -1
5829 032316 177777
5830 032320 012706 001100      ZZCDONE:MOV  #STACK,SP      ;SET UP STACK POINTER
5831 032324 104413      RSETUP      ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
5832
5842
5843      ;*****
(3) ;*TEST 76      SPECIAL CASE TEST
(4) ;*THIS TEST IS DERIVED FROM THE FORTRAN 4 BENCH MARKS
(4) ;*IT WAS FOUND THAT THIS CODE FAILED EVEN THOUGH THE
(4) ;*FP UNIT HAD PASSED THE DIAGNOSTICS. THE HARDWARE WAS
(4) ;*MODIFIED TO CURRECT THE ERROR BUT SINCE A TEST DEFICIENCY
(4) ;*WAS INDICATED THIS TEST WAS ADDED.
(4) ;*ALL THE TEST DOES IS PUT A MIXED NUMBER IN THE FPAC
(4) ;*AND SUBTRACTS A WHOLE NUMBER FROM IT.
(3) ;*****
(2) 032326 000004      TST76: SCOPE
5844
5845 032330 012746 144724      AAD1:  MOV      #144724, -(SP) ;PUT FRACTION ON STACK
5846 032334 012746 040600      MOV      #40600, -(SP)      ;PUT EXPONENT ON STACK
5847 032340 005046      CLR      -(SP)              ;PUT SUBTRAHEND FRACTION ON STACK
5848 032342 012746 040600      MOV      #40600, -(SP)      ;PUT SUBTRAHEND EXPONENT ON STACK
5849 032346 172466 000004      LDF      4(SP), ACO          ;LOAD FP ACCUMULATORS
5850 032352 173026      SUBF     (SP)+, ACO          ;DO SUBTRACTION
5851 032354 174037 032404      STF      ACO, @#AADBF      ;GET AND STORE ANSWER
    
```



```

5852 032360 022737 036711 032404      CMP      #36711, @#AADBFB :IS EXPONENT CORRECT
5853 032366 001401                      BEQ      1$              :IF YES GO CHECK FRACTION
5854 032370 104002                      ERROR   2              :BAD EXPONENT FROM SUBTRACTION
5855 032372 022737 152000 032406 1$:  CMP      #152000, @#AADBFB+2 :IS FRACTION CORRECT
5856 032400 001403                      BEQ     AADDONE         :IF YES GO TO END OF TEST
5857 032402 104002                      ERROR   2              :FRACTION INCORRECT
5858
5859 032404 000000                      AADBFB: .WORD 0
5860 032406 000000                      .WORD 0
5861
5862 032410 012706 001100                      AADDONE: MOV #STACK, SP :RESTORE STACK POINTER
5863 032414 104413                      RSETUP :GO INITIALIZE THE FPS AND STACK; AND
(1) :SEE IF THE USER HAS EXPRESSED
(1) :THE DESIRE TO CHANGE THE SOFTWARE
(1) :VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) :THE USER TYPED CONTROL G?).
5864
5865
5872
5873
(3)
(4)
(4)
(4)
(4)
(4)
(3)
(2) 032416 000004
5874
5875 032420 032737 000001 001336 ZD1: BIT #1, @#SENV :ARE WE ON APT
5876 032426 001403                      BEQ     ZD2            :IF NO DO THIS TEST ALWAYS
5877 032430 005737 001324                      TST    @#SPASS        :IF YES THEN CHECK PASS COUNTER
5878 032434 001122                      BNE    $EOP           :AND ONLY DO IT ON FIRST PASS
5879 032436 005001                      ZD2: CLR R1 :INITIALIZE A COUPLE OF COUNTERS
5880 032440 005000                      CLR   R0
5881 032442 172627 040400                      LDF   #2, AC2 :MAKE SURE FP ACCUMALATOR NON-ZERO
5882 032446 016767 145412 146556                      MOV   TPVEC, $TMP0 :SAVE INTERRUPT VECTOR
5883 032454 016767 145406 146552                      MOV   TPVEC+2, $TMP1 :SAVE INTERRUPT PRIORITY
5884 032462 012767 032534 145374                      MOV   #3$, TPVEC :SET UP VECTOR FOR THIS TEST
5885 032470 005067 145372                      CLR   TPVEC+2 :SET NEW PRIORITY TO 0
5886 032474 005067 145276                      CLR   PS :SET PROCESSOR PRIORITY TO 0
5887 032500 105077 146446                      CLRB @STPB :SEND A CHARACTER
5888 032504 105777 146440 1$: TSTB @STPS :WAIT FOR DONE ON THIS CHARACTER
5889 032510 100375                      BPL   1$
5890 032512 105077 146434                      CLRB @STPB :SEND A SECON CHARACTER
5891 032516 052777 000100 146424                      BIS   #BIT6, @STPS :SET INTERRUPT ENABLE
5892 032524 005200 2$: INC R0 :INCREMENT COUNT
5893 032526 001376                      BNE   2$ :IF NO INTERRUPT BEFORE 0 ERROR
5894 032530 000005                      RESET :CLEAR ALL BITS IN SLU
5895 032532 104003                      ERROR 3 :NO INTERRUPT OCCURRED
5896 032534 166700 000112 3$: SUB Y, R0 :SUBTRACT TIME FOR FP INSTRUCTION
5897 032540 010067 000110                      MOV   R0, Z :SAVE PRE LOOP COUNT
5898 032544 012767 032624 145312                      MOV   #7$, TPVEC :SET UP VECTOR FOR INTERRUPT FROM FP
5899 032552 005100 4$: COM R0 :MAKE COUNT NEGATIVE
5900 032554 005077 146370                      CLR   @STPS :DON'T ALLOW INTERRUPTS ON FIRST CHARACTER
5901 032560 105077 146366                      CLRB @STPB :SEND FIRST CHARACTER
5902 032564 105777 146360 5$: TSTB @STPS :WAIT FOR READY

```

```

:*****
:*TEST 77 INTERRUPTABILITY TEST
:* F11 INTERRUPTABILITY TEST
:* THIS TEST WILL VERIFY THE ABILITY OF THE KEF11-A FLOATING
:* POINT PROCESSOR TO BE INTERRUPTED DURING EXECUTION
:* OF MICRO-CODE MULTIPLY LOOP.
:*****
TST77: SCOPE

```

```

5903 032570 100375          BPL      5$
5904 032572 105077 146354  CLRB    @$TPB      ;SEND SECOND CHARACTER
5905 032576 052777 000100 146344  BIS     #BIT6, @$TPS ;SET INTERRUPT ENABLE
5906 032604 005200          6$:    INC     R0      ;DO PRE LOOP
5907 032606 001376          BNE     6$
5908 032610 171227 040400          10$:   MULF   #2, AC2     ;DO FP INSTRUCTION
5909 032614 171227 040400          MULF   #2, AC2     ;JUST INCASE INTERRUPT HAPPENS TOO LATE
5910 032620 000005          RESET          ;CLEAR ALL STATUS BITS IN SLU
5911 032622 104003          ERROR        3     ;NO INTERRUPT OCCURRED SO ERROR
5912 032624 022716 032610          7$:    CMP    #10$, (SP) ;IS PC ON STACK POINTING TO FP INSTRUCTION
5913 032630 001412          BEQ    ZZDDONE    ;IF YES WE ARE DONE
5914 032632 062706 000004          ADD    #4, SP     ;UPDATE STACK POINTER
5915 032636 062767 000002 000010          ADD    #2, Z      ;ADD A LITTLE TIME TO PRE LOOP
5916 032644 016700 000004          MOV    Z, R0     ;PUT NEW COUNT IN LOOP COUNTER
5917 032650 000740          BR     4$       ;DO IT ALL AGAIN
5918
5919 032652 000027          Y:     .WORD   27
5920 032654 000000          Z:     .WORD   0
5921
5922 032656          ZZDDONE:
5923 032656 042777 000100 146264          BIC    #BIT6, @$TPS ;CLEAR INTERRUPT ENABLE BIT
5924 032664 016767 146342 145172          MOV    $TMP0, TPVEC ;RESTORE INTERRUPT VECTOR
5925 032672 016767 146336 145166          MOV    $TMP1, TPVEC+2 ;RESTORE INTERRUPT PRIORITY
5926 032700 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1) ;SEE IF THE USER HAS EXPRESSED
(1) ;THE DESIRE TO CHANGE THE SOFTWARE
(1) ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1) ;THE USER TYPED CONTROL G?).
5927

```


5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939 032702 000004
5940 032704 005067 146172
5941 032710 005067 146366
5942 032714 005267 146404
5943 032720 042767 100000 146376
5944 032726 005327
5945 032730 000001
5946 032732 003045
5947 032734 012737
5948 032736 000001
5949 032740 032730
5950 032742 104401 033115
5951 032746 016746 146352
5952 032752 104405
5953 032754 104401 033112
5954 032760 013700 000042
5955 032764 001411
5956 032766 005046
5957 032770 012746 032776
5958 032774 000442
5959
5960 032776 000005
5961 033000 004710
5962 033002 000240
5963 033004 000240
5964 033006 000240
5965 033010 013737 000004 001232
5966 033016 012737 033034 000004
5967 033024 012737 000001 164000
5968 033032 000402
5969 033034 062706 000004 1\$:
5970 033040 013737 001232 000004 2\$:
5971 033046 104400
5972 033050 042716 000020
5973 033054 032777 010000 146056
5974 033062 001005
5975 033064 005167 000020
5976 033070 100402
5977 033072 052716 000020
5978 033076 012746 033104
5979 033102 000002
5980
5981 033104 000137
5982 033106 002056
5983 033110 000000
5984 033112 377 377 000

.SBTTL END OF PASS ROUTINE
:*****
:*INCREMENT THE PASS NUMBER (\$PASS)
:*INDICATE END-OF-PROGRAM AFTER 1 PASS THRU THE PROGRAM
:*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
:*IF SW12=1 INHIBIT TRACE TRAP
:*IF THERE IS A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO LOOP
:*****
\$EOP: SCOPE
CLR \$STNM ;ZERO THE TEST NUMBER
CLR \$TIMES ;ZERO THE NUMBER OF ITERATIONS
INC \$PASS ;INCREMENT THE PASS NUMBER
BIC #100000,\$PASS ;DON'T ALLOW A NEGATIVE PASS NUMBER
DEC (PC)+ ;LOOP?
\$EOPCT: .WORD 1 ;1 PASS FIRST TIME (QV)!!!
BGT \$DOAGN ;YES
MOV (PC)+,@(PC)+ ;RESTORE COUNTER
\$ENDCT: .WORD 1
\$SENDMG TYPE ;TYPE 'END PASS #'
MOV \$PASS,-(SP) ;SAVE PASS COUNT FOR TYPEOUT
TYPDS ;TYPE PASS COUNT IN DECIMAL
TYPE \$ENULL ;TYPE A NULL CHARACTER STRING
\$GET42: MOV @#42,R0 ;GET MONITOR ADDRESS
BEQ DOAGN ;BRANCH IF NO MONITOR
CLR -(SP) ;INSURE THE 'T' BIT IS CLEAR
MOV #\$CLR.T,-(SP) ;SETUP FOR AN RTI OR RTT
BR \$RTRN ;GO DO AN RTI OR RTT TO LOAD THE PSW
;WITH A CLEARED 'T' BIT
\$CLR.T: RESET ;CLEAR THE WORLD
\$ENDAD: JSR PC,(R0) ;GO TO THE MONITOR
NOP ;SAVE ROOM
NOP ;FOR
NOP ;ACT11
DOAGN: MOV @#4,@#\$TMP0 ;SAVE CONTENTS OF LOCATION 4
MOV #1\$,@#4 ;SET UP INCASE OF TRAP
MOV #1,@#164000 ;NOTIFY MULTI-TESTER
BR 2\$;NO TRAP SO DON'T RESET STACK
1\$: ADD #4,SP ;RESET STACK AFTER TRAP
2\$: MOV @#\$TMP0,@#4 ;RESTORE CONTENTS OF LOCATION 4
\$DOAGN: TRAP ;PUSH OLD PSW AND PC ON STACK
BIC #20,(SP) ;CLEAR THE 'T' BIT
BIT #BIT12,@SWR ;RUN WITH TRACE TRAP?
BNE 1\$;BRANCH IF NO
COM \$TBIT ;IS IT TIME FOR TRACE TRAP
BMI 1\$;BRANCH IF NO
BIS #20,(SP) ;SET TRACE TRAP
1\$: MOV #\$LOOP,-(SP) ;RETURN HERE FROM RTI
\$RTRN: RTI ;RETURN--THIS IS CHANGED TO AN 'RTT'
;IF IT IS A LEGAL INSTRUCTION
;RETURN TO TESTING
\$LOOP: JMP @(PC)+
\$RTNAD: .WORD LOOP
\$TBIT: .WORD 0 ;'T' BIT STATE INDICATOR
\$ENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING


```
(1) 033324 012767 000001 145552 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
(1) 033332 016767 000052 145742 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
(1) 033340 105267 145536 $SVLAD: INCB $STNM ;;COUNT TEST NUMBERS
(1) 033344 116767 145532 145750 MOV $STNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
(1) 033352 011667 145530 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
(1) 033356 011667 145526 MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
(1) 033362 005067 145716 CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 033366 112767 000001 145521 MOV $1,$ERMAX ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 033374 016777 145502 145540 $OVER: MOV $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1) 033402 016716 145500 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
(1) 033406 000002 RTI ;;FIXES PS
(1) 033410 000001 $MXCNT: 1 ;;MAX. NUMBER OF ITERATIONS
```

5988

5989

```
(1) .SBTTL ERROR HANDLER ROUTINE
(2) ;:*****
(1) ;:THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1) ;:SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1) ;:AND GO TO ERTYPE ON ERROR
(1) ;:THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;:*SW15=1 HALT ON ERROR
(1) ;:*SW13=1 INHIBIT ERROR TYPEOUTS
(1) ;:*SW10=1 BELL ON ERROR
(1) ;:*SW09=1 LOOP ON ERROR
(1) ;:*CALL
(1) ;:* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1) $ERROR:
(1) 033412
(1) 033412 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
(1) 033414 105267 145463 7$: INCB $ERFLG ;;SET THE ERROR FLAG
(1) 033420 001775 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
(1) 033422 016777 145454 145512 MOV $STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1) 033430 032777 002000 145502 BIT #BIT10,@SWR ;;BELL ON ERROR?
(1) 033436 001402 BEQ 1$ ;;NO - SKIP
(1) 033440 104401 001306 TYPE $BELL ;;RING BELL
(1) 033444 005267 145442 1$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
(1) 033450 011667 145442 MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
(1) 033454 162767 000002 145434 SUB #2,$ERRPC
(1) 033462 117767 145430 145424 MOV $ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
(1) 033470 032777 020000 145442 BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET
(1) 033476 001004 BNE 20$ ;;SKIP TYPEOUTS
(1) 033500 004767 002314 JSR PC,ERTYPE ;;GO TO USER ERROR ROUTINE
(1) 033504 104401 001313 TYPE $CRLF
(1) 033510 20$: CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
(1) 033516 001007 BNE 2$ ;;NO,SKIP APT ERROR REPORT
(1) 033520 116767 145370 000004 MOV $ITEMB,21$ ;;SET ITEM NUMBER AS ERROR NUMBER
(1) 033526 004767 001126 JSR PC,$ATY4 ;;REPORT FATAL ERROR TO APT
(1) 033532 000 .BYTE 0
(1) 033533 000 .BYTE 0
(1) 033534 000777 21$: BR 22$ ;;APT ERROR LOOP
(1) 033536 005777 145376 22$: BR 2$ ;;HALT ON ERROR
(1) 033542 100002 2$: TST @SWR ;;SKIP IF CONTINUE
(1) 033544 000000 BPL 3$ ;;HALT ON ERROR!
(1) 033546 104407 HALT
(1) 033550 032777 001000 145362 3$: CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
```


(1) 033704 000002
5992
5993

RTI

.SBTTL TYPE ROUTINE

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

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

(1)	033706	105767	145245	\$TYPE:	TSTB	\$TPFLG	:: IS THERE A TERMINAL?	
(1)	033712	100002			BPL	1\$:: BR IF YES	
(1)	033714	000000			HALT		:: HALT HERE IF NO TERMINAL	
(1)	033716	000430			BR	3\$:: LEAVE	
(1)	033720	010046		1\$:	MOV	R0,-(SP)	:: SAVE R0	
(1)	033722	017600	000002		MOV	@2(SP),R0	:: GET ADDRESS OF ASCIZ STRING	
(1)	033726	122767	000001	145402	CMPB	#APTENV,\$ENV	:: RUNNING IN APT MODE	
(1)	033734	001011			BNE	62\$:: NO,GO CHECK FOR APT CONSOLE	
(1)	033736	132767	000100	145373	BITB	#APTPOOL,\$ENVM	:: SPOOL MESSAGE TO APT	
(1)	033744	001405			BEQ	62\$:: NO,GO CHECK FOR CONSOLE	
(1)	033746	010067	000004		MOV	R0,61\$:: SETUP MESSAGE ADDRESS FOR APT	
(1)	033752	004767	000672		JSR	PC,\$ATY3	:: SPOOL MESSAGE TO APT	
(1)	033756	000000		61\$:	.WORD	0	:: MESSAGE ADDRESS	
(1)	033760	132767	000040	145351	62\$:	BITB	#APTCSUP,\$ENVM	:: APT CONSOLE SUPPRESSED
(1)	033766	001003			BNE	60\$:: YES,SKIP TYPE OUT	
(1)	033770	112046		2\$:	MOVB	(R0)+,-(SP)	:: PUSH CHARACTER TO BE TYPED ONTO STACK	
(1)	033772	001005			BNE	4\$:: BR IF IT ISN'T THE TERMINATOR	
(1)	033774	005726			TST	(SP)+	:: IF TERMINATOR POP IT OFF THE STACK	
(1)	033776	012600		60\$:	MOV	(SP)+,R0	:: RESTORE R0	
(1)	034000	062716	000002	3\$:	ADD	#2,(SP)	:: ADJUST RETURN PC	
(1)	034004	000002			RTI		:: RETURN	
(1)	034006	122716	000011	4\$:	CMPB	#HT,(SP)	:: BRANCH IF <HT>	
(1)	034012	001430			BEQ	8\$		
(1)	034014	122716	000200		CMPB	#CRLF,(SP)	:: BRANCH IF NOT <CRLF>	
(1)	034020	001006			BNE	5\$		
(1)	034022	005726			TST	(SP)+	:: POP <CR><LF> EQUIV	
(1)	034024	104401			TYPE		:: TYPE A CR AND LF	
(1)	034026	001313			\$CRLF			
(1)	034030	105067	000130		CLRB	\$CHARCNT	:: CLEAR CHARACTER COUNT	
(1)	034034	000755			BR	2\$:: GET NEXT CHARACTER	
(1)	034036	004767	000056	5\$:	JSR	PC,\$TYPEC	:: GO TYPE THIS CHARACTER	
(1)	034042	126726	145110	6\$:	CMPB	\$FILLC,(SP)+	:: IS IT TIME FOR FILLER CHARS.?	
(1)	034046	001350			BNE	2\$:: IF NO GO GET NEXT CHAR.	
(1)	034050	016746	145100		MOV	\$NULL,-(SP)	:: GET # OF FILLER CHARS. NEEDED	
(1)							:: AND THE NULL CHAR.	
(1)	034054	105366	000001	7\$:	DECB	1(SP)	:: DOES A NULL NEED TO BE TYPED?	
(1)	034060	002770			BLT	6\$:: BR IF NO--GO POP THE NULL OFF OF STACK	

```
(1) 034062 004767 000032 JSR PC,$TYPEC ;;GO TYPE A NULL
(1) 034066 105367 000072 DECB $CHARCNT ;;DO NOT COUNT AS A COUNT
(1) 034072 000770 BR 7$ ;;LOOP
(1)
(1) ;HORIZONTAL TAB PROCESSOR
(1)
(1) 034074 112716 000040 8$: MOVB #' ,(SP) ;;REPLACE TAB WITH SPACE
(1) 034100 004767 000014 9$: JSR PC,$TYPEC ;;TYPE A SPACE
(1) 034104 132767 000007 000052 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
(1) 034112 001372 BNE 9$ ;;TAB STOP
(1) 034114 005726 TST (SP)+ ;;POP SPACE OFF STACK
(1) 034116 000724 BR 2$ ;;GET NEXT CHARACTER
(1) 034120 105777 145024 $TYPEC: TSTB @ $TPS ;;WAIT UNTIL PRINTER IS READY
(1) 034124 100375 BPL $TYPEC
(1) 034126 116677 000002 145016 MOVB 2(SP),@$TPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 034134 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
(1) 034142 001003 BNE 1$ ;;BRANCH IF NO
(1) 034144 105067 000014 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
(1) 034150 000406 BR $TYPEX ;;EXIT
(1) 034152 122766 000012 000002 1$: CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?
(1) 034160 001402 BEQ $TYPEX ;;BRANCH IF YES
(1) 034162 105227 INCB (PC)+ ;;COUNT THE CHARACTER
(1) 034164 000000 $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
(1) 034166 000207 $TYPEX: RTS PC
```

5994
5995

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```
(1)
(2)
(1) *****
(1) *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
(1) *OCTAL (ASCII) NUMBER AND TYPE IT.
(1) *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1) *CALL:
(1) * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
(1) * TYPOS ;;CALL FOR TYPEOUT
(1) * .BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1) * .BYTE M ;;M=1 OR 0
(1) * ;;1=TYPE LEADING ZEROS
(1) * ;;0=SUPPRESS LEADING ZEROS
(1) *
(1) *$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1) *$TYPOS OR $TYPOC
(1) *CALL:
(1) * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
(1) * TYPON ;;CALL FOR TYPEOUT
(1) *
(1) *$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1) *CALL:
(1) * MOV NUM,-(SP) ;;NUMBER TO BE TYPED
(1) * TYPOC ;;CALL FOR TYPEOUT
(1) *
(1) 034170 017646 000000 $TYPOS: MOV @(SP),-(SP) ;;PICKUP THE MODE
(1) 034174 116667 000001 000211 MOVB 1(SP),$OFILL ;;LOAD ZERO FILL SWITCH
(1) 034202 112667 000207 MOVB (SP)+,$OMODE+1 ;;NUMBER OF DIGITS TO TYPE
(1) 034206 062716 000002 ADD #2,(SP) ;;ADJUST RETURN ADDRESS
(1) 034212 000406 BR $TYPON
```



```

(1) 034214 112767 000001 000171 $TYPOC: MOVB #1,$OFILL      ;;SET THE ZERO FILL SWITCH
(1) 034222 112767 000006 000165      MOVB #6,$SOMODE+1  ;;SET FOR SIX(6) DIGITS
(1) 034230 112767 000005 000154 $TYPON: MOVB #5,$OCNT  ;;SET THE ITERATION COUNT
(1) 034236 010346      MOV R3,-(SP)      ;;SAVE R3
(1) 034240 010446      MOV R4,-(SP)      ;;SAVE R4
(1) 034242 010546      MOV R5,-(SP)      ;;SAVE R5
(1) 034244 116704 000145      MOVB $SOMODE+1,R4  ;;GET THE NUMBER OF DIGITS TO TYPE
(1) 034250 005404      NEG R4            ;;SUBTRACT IT FOR MAX. ALLOWED
(1) 034252 062704 000006      ADD #6,R4        ;;SAVE IT FOR USE
(1) 034256 110467 000132      MOVB R4,$SOMODE  ;;GET THE ZERO FILL SWITCH
(1) 034262 116704 000125      MOVB $OFILL,R4   ;;PICKUP THE INPUT NUMBER
(1) 034266 016605 000012      MOV 12(SP),R5   ;;CLEAR THE OUTPUT WORD
(1) 034272 005003      CLR R3          ;;ROTATE MSB INTO 'C'
(1) 034274 006105      1$: ROL R5     ;;GO DO MSB
(1) 034276 000404      BR 3$          ;;FORM THIS DIGIT
(1) 034300 006105      2$: ROL R5
(1) 034302 006105      ROL R5
(1) 034304 006105      ROL R5
(1) 034306 010503      MOV R5,R3
(1) 034310 006103      3$: ROL R3     ;;GET LSB OF THIS DIGIT
(1) 034312 105367 000076      DECB $SOMODE    ;;TYPE THIS DIGIT?
(1) 034316 100016      BPL 7$         ;;BR IF NO
(1) 034320 042703 177770      BIC #177770,R3  ;;GET RID OF JUNK
(1) 034324 001002      BNE 4$         ;;TEST FOR 0
(1) 034326 005704      TST R4        ;;SUPPRESS THIS 0?
(1) 034330 001403      BEQ 5$        ;;BR IF YES
(1) 034332 005204      4$: INC R4     ;;DON'T SUPPRESS ANYMORE 0'S
(1) 034334 052703 000060      BIS #'0,R3     ;;MAKE THIS DIGIT ASCII
(1) 034340 052703 000040      5$: BIS #' ,R3  ;;MAKE ASCII IF NOT ALREADY
(1) 034344 110367 000040      MOVB R3,8$     ;;SAVE FOR TYPING
(1) 034350 104401 034410      TYPE ,8$      ;;GO TYPE THIS DIGIT
(1) 034354 105367 000032      7$: DECB $OCNT  ;;COUNT BY 1
(1) 034360 003347      BGT 2$        ;;BR IF MORE TO DO
(1) 034362 002402      BLT 6$        ;;BR IF DONE
(1) 034364 005204      INC R4        ;;INSURE LAST DIGIT ISN'T A BLANK
(1) 034366 000744      BR 2$        ;;GO DO THE LAST DIGIT
(1) 034370 012605      6$: MOV (SP)+,R5  ;;RESTORE R5
(1) 034372 012604      MOV (SP)+,R4  ;;RESTORE R4
(1) 034374 012603      MOV (SP)+,R3  ;;RESTORE R3
(1) 034376 016666 000002 000004      MOV 2(SP),4(SP) ;;SET THE STACK FOR RETURNING
(1) 034404 012616      MOV (SP)+,(SP)
(1) 034406 000002      RTI          ;;RETURN
(1) 034410 000      8$: .BYTE 0    ;;STORAGE FOR ASCII DIGIT
(1) 034411 000      .BYTE 0     ;;TERMINATOR FOR TYPE ROUTINE
(1) 034412 000      $OCNT: .BYTE 0 ;;OCTAL DIGIT COUNTER
(1) 034413 000      $OFILL: .BYTE 0 ;;ZERO FILL SWITCH
(1) 034414 000000      $SOMODE: .WORD 0 ;;NUMBER OF DIGITS TO TYPE
  
```

5996
5997

```

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.
  
```

```

(1)          ;*CALL:
(1)          ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
(1)          ;*      TYPDS      ;;GO TO THE ROUTINE
(1)          $TYPDS:
(1) 034416   MOV      R0,-(SP)      ;;PUSH R0 ON STACK
(3) 034416   010046   MOV      R1,-(SP)      ;;PUSH R1 ON STACK
(3) 034420   010146   MOV      R2,-(SP)      ;;PUSH R2 ON STACK
(3) 034422   010246   MOV      R3,-(SP)      ;;PUSH R3 ON STACK
(3) 034424   010346   MOV      R5,-(SP)      ;;PUSH R5 ON STACK
(3) 034426   010546   MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
(1) 034430   012746   020200  MOV      20(SP),R5    ;;GET THE INPUT NUMBER
(1) 034434   016605   000020  BPL      1$          ;;BR IF INPUT IS POS.
(1) 034440   100004          NEG      R5          ;;MAKE THE BINARY NUMBER POS.
(1) 034442   005405          MOVB    #'-,1(SP)   ;;MAKE THE ASCII NUMBER NEG.
(1) 034444   112766   000055  000001  1$:      CLR      R0          ;;ZERO THE CONSTANTS INDEX
(1) 034452   005000          MOV     #$DBLK,R3   ;;SETUP THE OUTPUT POINTER
(1) 034454   012703   034632  MOVB    #' ,(R3)+   ;;SET THE FIRST CHARACTER TO A BLANK
(1) 034460   112723   000040  2$:      CLR      R2          ;;CLEAR THE BCD NUMBER
(1) 034464   005002          MOV     $DTBL(R0),R1 ;;GET THE CONSTANT
(1) 034466   016001   034622  3$:      SUB     R1,R5        ;;FORM THIS BCD DIGIT
(1) 034472   160105          BLT     4$          ;;BR IF DONE
(1) 034474   002402          INC     R2          ;;INCREASE THE BCD DIGIT BY 1
(1) 034476   005202          BR     3$
(1) 034500   000774          ADD     R1,R5        ;;ADD BACK THE CONSTANT
(1) 034502   060105          TST     R2          ;;CHECK IF BCD DIGIT=0
(1) 034504   005702          BNE     5$          ;;FALL THROUGH IF 0
(1) 034506   001002          TSTB   (SP)         ;;STILL DOING LEADING 0'S?
(1) 034510   105716          BMI     7$          ;;BR IF YES
(1) 034512   100407          ASLB   (SP)         ;;MSD?
(1) 034514   106316          BCC     6$          ;;BR IF NO
(1) 034516   103003          MOVB   1(SP),-1(R3)  ;;YES--SET THE SIGN
(1) 034520   116663   000001  177777  6$:      BIS     #'0,R2       ;;MAKE THE BCD DIGIT ASCII
(1) 034526   052702   000060  7$:      BIS     #' ,R2       ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 034532   052702   000040          MOVB   R2,(R3)+     ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 034536   110223          TST    (R0)+        ;;JUST INCREMENTING
(1) 034540   005720          CMP    R0,#10       ;;CHECK THE TABLE INDEX
(1) 034542   020027   000010  2$:      BLT     8$          ;;GO DO THE NEXT DIGIT
(1) 034546   002746          BGT     8$          ;;GO TO EXIT
(1) 034550   003002          MOV    R5,R2        ;;GET THE LSD
(1) 034552   010502          BR     6$          ;;GO CHANGE TO ASCII
(1) 034554   000764          TSTB   (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
(1) 034556   105726          BPL     9$          ;;BR IF NO
(1) 034560   100003          MOVB   -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
(1) 034562   116663   177777  177776  9$:      CLRB   (R3)        ;;SET THE TERMINATOR
(3) 034572   012605          MOV    (SP)+,R5     ;;POP STACK INTO R5
(3) 034574   012603          MOV    (SP)+,R3     ;;POP STACK INTO R3
(3) 034576   012602          MOV    (SP)+,R2     ;;POP STACK INTO R2
(3) 034600   012601          MOV    (SP)+,R1     ;;POP STACK INTO R1
(3) 034602   012600          MOV    (SP)+,R0     ;;POP STACK INTO R0
(1) 034604   104401   034632  TYPE    $DBLK       ;;NOW TYPE THE NUMBER
(1) 034610   016666   000002  000004  MOV     2(SP),4(SP)  ;;ADJUST THE STACK
(1) 034616   012616          MOV    (SP)+,(SP)
(1) 034620   000002          RTI
(1) 034622   023420          $DTBL: 10000.
(1) 034624   001750          1000.

```



```

(1) 034626 000144          100.
(1) 034630 000012          10.
(1) 034632 000004          $DBLK: .BLKW 4
5998
5999
(1) .SBTTL APT COMMUNICATIONS ROUTINE
(2)
(1) 034642 112767 000001 000236 $ATY1: MOVB #1,$FFLG      ;;TO REPORT FATAL ERROR
(1) 034650 112767 000001 000226 $ATY3: MOVB #1,$MFLG      ;;TO TYPE A MESSAGE
(1) 034656 000403          BR $ATYC
(1) 034660 112767 000001 000220 $ATY4: MOVB #1,$FFLG      ;;TO ONLY REPORT FATAL ERROR
(1) 034666          $ATYC:
(3) 034666 010046          MOV R0,-(SP)      ;;PUSH R0 ON STACK
(3) 034670 010146          MOV R1,-(SP)      ;;PUSH R1 ON STACK
(1) 034672 105767 000206          TSTB $MFLG      ;;SHOULD TYPE A MESSAGE?
(1) 034676 001450          BEQ 5$           ;;IF NOT: BR
(1) 034700 122767 000001 144430  CMPB #APTENV,$ENV  ;;OPERATING UNDER APT?
(1) 034706 001031          BNE 3$           ;;IF NOT: BR
(1) 034710 132767 000100 144421  BITB #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(1) 034716 001425          BEQ 3$           ;;IF NOT: BR
(1) 034720 017600 000004          MOV @4(SP),R0     ;;GET MESSAGE ADDR.
(1) 034724 062766 000002 000004  ADD #2,4(SP)      ;;BUMP RETURN ADDR.
(1) 034732 005767 144360          1$: TST $MSGTYPE    ;;SEE IF DONE W/ LAST XMISSION?
(1) 034736 001375          BNE 1$           ;;IF NOT: WAIT
(1) 034740 010067 144366          MOV R0,$MSGAD     ;;PUT ADDR IN MAILBOX
(1) 034744 105720          2$: TSTB (R0)+      ;;FIND END OF MESSAGE
(1) 034746 001376          BNE 2$
(1) 034750 166700 144356          SUB $MSGAD,R0     ;;SUB START OF MESSAGE
(1) 034754 006200          ASR R0            ;;GET MESSAGE LNTH IN WORDS
(1) 034756 010067 144352          MOV R0,$MSGGLT    ;;PUT LENGTH IN MAILBOX
(1) 034762 012767 000004 144326  MOV #4,$MSGTYPE   ;;TELL APT TO TAKE MSG.
(1) 034770 000413          BR 5$
(1) 034772 017667 000004 000016 3$: MOV @4(SP),4$    ;;PUT MSG ADDR IN JSR LINKAGE
(1) 035000 062766 000002 000004  ADD #2,4(SP)      ;;BUMP RETURN ADDRESS
(3) 035006 016746 142764          MOV 177776,-(SP) ;;PUSH 177776 ON STACK
(1) 035012 004767 176670          JSR PC,$TYPE     ;;CALL TYPE MACRO
(1) 035016 000000          4$: .WORD 0
(1) 035020          5$:
(1) 035020 105767 000062          10$: TSTB $FFLG      ;;SHOULD REPORT FATAL ERROR?
(1) 035024 001416          BEQ 12$          ;;IF NOT: BR
(1) 035026 005767 144304          TST $ENV         ;;RUNNING UNDER APT?
(1) 035032 001413          BEQ 12$          ;;IF NOT: BR
(1) 035034 005767 144256          11$: TST $MSGTYPE    ;;FINISHED LAST MESSAGE?
(1) 035040 001375          BNE 11$          ;;IF NOT: WAIT
(1) 035042 017667 000004 144250  MOV @4(SP),$FATAL ;;GET ERROR #
(1) 035050 062766 000002 000004  ADD #2,4(SP)      ;;BUMP RETURN ADDR.
(1) 035056 005267 144234          INC $MSGTYPE     ;;TELL APT TO TAKE ERROR
(1) 035062 105067 000020          12$: CLRB $FFLG     ;;CLEAR FATAL FLAG
(1) 035066 105067 000013          CLRB $LFLG      ;;CLEAR LOG FLAG
(1) 035072 105067 000006          CLRB $MFLG      ;;CLEAR MESSAGE FLAG
(3) 035076 012601          MOV (SP)+,R1     ;;POP STACK INTO R1
(3) 035100 012600          MOV (SP)+,R0     ;;POP STACK INTO R0
(1) 035102 000207          RTS PC          ;;RETURN
(1) 035104 000          $MFLG: .BYTE 0   ;;MESSG. FLAG
(1) 035105 000          $LFLG: .BYTE 0   ;;LOG FLAG
(1) 035106 000          $FFLG: .BYTE 0   ;;FATAL FLAG

```

(1) 035110
(1) 000200
(1) 000001
(1) 000100
(1) 000040

.EVEN
APTSIZE=200
APTENV=001
APTSPOOL=100
APTCSUP=040

6000
6001
6002

.SBTTL TTY INPUT ROUTINE

(1)
(2)
(1)
(1)
(2)
(1)

::*****
.ENABL LSB

(1)
(1)
(1)
(1)

::*****
*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
*WHEN OPERATING IN TTY FLAG MODE.

(1) 035110 022767 000176 144022
(1) 035116 001074
(1) 035120 105777 144020
(1) 035124 100071
(1) 035126 117746 144014
(1) 035132 042716 177600
(1) 035136 022726 000007
(1) 035142 001062
(1) 035144 126727 143764 000001
(1) 035152 001456

\$CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
BNE 15\$;;BRANCH IF NO
TSTB @TKS ;;CHAR THERE?
BPL 15\$;;IF NO, DON'T WAIT AROUND
MOVB @TKB, -(SP) ;;SAVE THE CHAR
BIC #^C177, (SP) ;;STRIP-OFF THE ASCII
CMP #7, (SP)+ ;;IS IT A CONTROL G?
BNE 15\$;;NO, RETURN TO USER
CMPB \$AUTOB, #1 ;;ARE WE RUNNING IN AUTO-MODE?
BEQ 15\$;;BRANCH IF YES

(1) 035154 104401 035517
(1) 035160 104401 035524
(2) 035164 016746 143006
(2) 035170 104402
(1) 035172 104401 035535
(1) 035176 005046
(1) 035200 005046
(1) 035202 105777 143736
(1) 035206 100375
(1)
(1) 035210 117746 143732
(1) 035214 042716 177600

\$GTSWR: TYPE , \$CNTLG ;;ECHO THE CONTROL-G (^G)
TYPE , \$MSWR ;;TYPE CURRENT CONTENTS
MOV SWREG, -(SP) ;;SAVE SWREG FOR TYPEOUT
TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
TYPE , \$MNEW ;;PROMPT FOR NEW SWR
19\$: CLR -(SP) ;;CLEAR COUNTER
CLR -(SP) ;;THE NEW SWR
7\$: TSTB @TKS ;;CHAR THERE?
BPL 7\$;;IF NOT TRY AGAIN

MOVB @TKB, -(SP) ;;PICK UP CHAR
BIC #^C177, (SP) ;;MAKE IT 7-BIT ASCII

(1)
(1) 035220 021627 000025
(1) 035224 001005
(1) 035226 104401 035512
(1) 035232 062706 000006
(1) 035236 000757

9\$: CMP (SP), #25 ;;IS IT A CONTROL-U?
BNE 10\$;;BRANCH IF NOT
TYPE , \$CNTLU ;;YES, ECHO CONTROL-U (^U)
20\$: ADD #6, SP ;;IGNORE PREVIOUS INPUT
BR 19\$;;LET'S TRY IT AGAIN

(1)
(1) 035240 021627 000015
(1) 035244 001022
(1) 035246 005766 000004
(1) 035252 001403
(1) 035254 016677 000002 143656
(1) 035262 062706 000006

10\$: CMP (SP), #15 ;;IS IT A <CR>?
BNE 16\$;;BRANCH IF NO
TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
BEQ 11\$;;BRANCH IF YES
11\$: MOV 2(SP), @SWR ;;SAVE NEW SWR
ADD #6, SP ;;CLEAR UP STACK


```
(1) 035266 104401 001313 14$: TYPE $SCRLF ::ECHO <CR> AND <LF>
(1) 035272 126727 143637 000001 CMPB $INTAG,#1 ::RE-ENABLE TTY KBD INTERRUPTS?
(1) 035300 001003 BNE 15$ ::BRANCH IF NOT
(1) 035302 012777 000100 143634 MOV #100,@$TKS ::RE-ENABLE TTY KBD INTERRUPTS
(1) 035310 000002 15$: RTI ::RETURN
(1) 035312 004767 176602 16$: JSR PC,$TYPEC ::ECHO CHAR
(1) 035316 021627 000060 CMP (SP),#60 ::CHAR < 0?
(1) 035322 002420 BLT 18$ ::BRANCH IF YES
(1) 035324 021627 000067 CMP (SP),#67 ::CHAR > 7?
(1) 035330 003015 BGT 18$ ::BRANCH IF YES
(1) 035332 042726 000060 BIC #60,(SP)+ ::STRIP-OFF ASCII
(1) 035336 005766 000002 TST 2(SP) ::IS THIS THE FIRST CHAR
(1) 035342 001403 BEQ 17$ ::BRANCH IF YES
(1) 035344 006316 ASL (SP) ::NO, SHIFT PRESENT
(1) 035346 006316 ASL (SP) :: CHAR OVER TO MAKE
(1) 035350 006316 ASL (SP) :: ROOM FOR NEW ONE.
(1) 035352 005266 000002 17$: INC 2(SP) ::KEEP COUNT OF CHAR
(1) 035356 056616 177776 BIS -2(SP),(SP) ::SET IN NEW CHAR
(1) 035362 000707 BR 7$ ::GET THE NEXT ONE
(1) 035364 104401 001312 18$: TYPE $QUES ::TYPE ?<CR><LF>
(1) 035370 000720 BR 20$ ::SIMULATE CONTROL-U
(1) .DSABL LSB
```

```
(1) ::*****
(1) ::*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1) ::*CALL:
(1) ::* RDCHR ::INPUT A SINGLE CHARACTER FROM THE TTY
(1) ::* RETURN HERE ::CHARACTER IS ON THE STACK
(1) ::* ::WITH PARITY BIT STRIPPED OFF
(1) ::*
```

```
(1) 035372 011646 $RDCHR: MOV (SP),-(SP) ::PUSH DOWN THE PC
(1) 035374 016666 000004 000002 MOV 4(SP),2(SP) ::SAVE THE PS
(1) 035402 105777 143536 1$: TSTB @$TKS ::WAIT FOR
(1) 035406 100375 BPL 1$ ::A CHARACTER
(1) 035410 117766 143532 000004 MOVB @$TKB,4(SP) ::READ THE TTY
(1) 035416 042766 177600 000004 BIC #^C<177>,4(SP) ::GET RID OF JUNK IF ANY
(1) 035424 026627 000004 000023 CMP 4(SP),#23 ::IS IT A CONTROL-S?
(1) 035432 001013 BNE 3$ ::BRANCH IF NO
(1) 035434 105777 143504 2$: TSTB @$TKS ::WAIT FOR A CHARACTER
(1) 035440 100375 BPL 2$ ::LOOP UNTIL ITS THERE
(1) 035442 117746 143500 MOVB @$TKB,-(SP) ::GET CHARACTER
(1) 035446 042716 177600 BIC #^C177,(SP) ::MAKE IT 7-BIT ASCII
(1) 035452 022627 000021 CMP (SP)+,#21 ::IS IT A CONTROL-Q?
(1) 035456 001366 BNE 2$ ::IF NOT DISCARD IT
(1) 035460 000750 BR 1$ ::YES, RESUME
(1) 035462 026627 000004 000140 3$: CMP 4(SP),#140 ::IS IT UPPER CASE?
(1) 035470 002407 BLT 4$ ::BRANCH IF YES
(1) 035472 026627 000004 000175 CMP 4(SP),#175 ::IS IT A SPECIAL CHAR?
(1) 035500 003003 BGT 4$ ::BRANCH IF YES
(1) 035502 042766 000040 000004 BIC #40,4(SP) ::MAKE IT UPPER CASE
(1) 035510 000002 4$: RTI ::GO BACK TO USER
(1) 035512 052536 005015 000 $CNTLU: .ASCIZ /^U/<15><12> ::CONTROL 'U'
(1) 035517 136 006507 000012 $CNTLG: .ASCIZ /^G/<15><12> ::CONTROL 'G'
(1) 035524 005015 053523 020122 $MSWR: .ASCIZ <15><12>/SWR = /
```

(1) 035532 020075 000
(1) 035535 040 047040
(1) 035542 036440 000040

053505 \$MNEW: .ASCIZ / NEW = /

6003
6004

.SBTTL TRAP DECODER

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

(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1) 035546 010046
(1) 035550 016600 000002
(1) 035554 005740
(1) 035556 111000
(1) 035560 006300
(1) 035562 016000 035602
(1) 035566 000200

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

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO

(1) 035570 011646
(1) 035572 016666 000004 000002
(1) 035600 000002

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

.SBTTL TRAP TABLE

;;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;;BY THE 'TRAP' INSTRUCTION.

(3)
(3)
(3)
(3)
(3)
(3)

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)
\$TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
\$GTSWR ;;CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING
\$CKSWR ;;CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
\$RDCHR ;;CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
\$SAVREG ;;CALL=SAVREG TRAP+11(104411) SAVE R0-R5 ROUTINE
\$RESREG ;;CALL=RESREG TRAP+12(104412) RESTORE R0-R5 ROUTINE
.RSET ;;CALL=RSETUP TRAP+13(104413) ROUTINE TO INITIALIZE AT END OF EACH TES
.LPER ;;CALL=LPERR TRAP+14(104414) ROUTINE TO SET UP LOOP ON ERROR ADDRESS
\$TERM=.-\$TRPAD

6005
6006
6007
6008
6009

.SBTTL POWER DOWN AND UP ROUTINES

(1)
(2)
(1)
(1) 035634 012737 036012 000024

POWER DOWN ROUTINE
\$PWRDN: MOV # \$ILLUP, @#PWRVEC ;;SET FOR FAST UP


```
(1) 035642 012737 000340 000026      MOV      #340,@#PWRVEC+2  ;;PRIO:7
(3) 035650 010046                    MOV      R0,-(SP)         ;;PUSH R0 ON STACK
(3) 035652 010146                    MOV      R1,-(SP)         ;;PUSH R1 ON STACK
(3) 035654 010246                    MOV      R2,-(SP)         ;;PUSH R2 ON STACK
(3) 035656 010346                    MOV      R3,-(SP)         ;;PUSH R3 ON STACK
(3) 035660 010446                    MOV      R4,-(SP)         ;;PUSH R4 ON STACK
(3) 035662 010546                    MOV      R5,-(SP)         ;;PUSH R5 ON STACK
(3) 035664 017746 143250             MOV      @SWR,-(SP)       ;;PUSH @SWR ON STACK
(1) 035670 010667 000122             MOV      SP,$SAVR6        ;;SAVE SP
(1) 035674 012737 035706 000024     MOV      #PWRUP,@#PWRVEC ;;SET UP VECTOR
(1) 035702 000000                    HALT
(1) 035704 000776                    BR       -2              ;;HANG UP
(1)
(2)
(1)
(1) 035706 012737 036012 000024     $PWRUP: MOV      #ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
(1) 035714 016706 000076             MOV      $SAVR6,SP        ;;GET SP
(1) 035720 005067 000072             CLR      $SAVR6          ;;WAIT LOOP FOR THE TTY
(1) 035724 005267 000066             1$: INC      $SAVR6        ;;WAIT FOR THE INC
(1) 035730 001375                    BNE     1$              ;;OF WORD
(3) 035732 012677 143202             MOV      (SP)+,@SWR       ;;POP STACK INTO @SWR
(3) 035736 012605                    MOV      (SP)+,R5        ;;POP STACK INTO R5
(3) 035740 012604                    MOV      (SP)+,R4        ;;POP STACK INTO R4
(3) 035742 012603                    MOV      (SP)+,R3        ;;POP STACK INTO R3
(3) 035744 012602                    MOV      (SP)+,R2        ;;POP STACK INTO R2
(3) 035746 012601                    MOV      (SP)+,R1        ;;POP STACK INTO R1
(3) 035750 012600                    MOV      (SP)+,R0        ;;POP STACK INTO R0
(1) 035752 012737 035634 000024     MOV      #PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1) 035760 012737 000340 000026     MOV      #340,@#PWRVEC+2 ;;PRIO:7
(1) 035766 104401                    TYPE
(1) 035770 036300                    $PWRMG: .WORD  POWERM    ;;REPORT THE POWER FAILURE
(1) 035772 012716                    MOV      (PC)+,(SP)      ;;POWER FAIL MESSAGE POINTER
(1) 035774 001370                    $PWRAD: .WORD  START     ;;RESTART AT START
(1) 035776 042766 000020 000002     BIC      #20,2(SP)       ;;RESTART ADDRESS
(1) 036004 005067 175100             CLR      $TBIT           ;;CLEAR 'T' BIT
(1) 036010 000002                    RTI
(1) 036012 000000                    $ILLUP: HALT            ;;THE POWER UP SEQUENCE WAS STARTED
(1) 036014 000776                    BR       -2              ;; BEFORE THE POWER DOWN WAS COMPLETE
(1) 036016 000000                    $SAVR6: 0                ;;PUT THE SP HERE
```

6010
6011
6012
6013
(2)
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025

```
;;*****  
;;*****  
;THIS ROUTINE IS CALLED TO TYPE AN ERROR MESSAGE WHICH IS INCLUDED  
;IN THE ERROR MESSAGE DATA TABLE. IT IS CALLED BY THE $ERROR ROUTINE  
;OR BY FIRST SETTING $ITEMB EQUAL TO THE ERROR TABLE ITEM TO BE PRINTED  
;OUT AND THEN EXECUTING A:  
;* JSR PC,ERTYPE  
;*  
ERTYPE: TYPE TYPE ;TYPE A CRLF  
        .WORD $CRLF  
        MOVB @#$STSTM,@#$TMP0  
        BIC #177400,@#$TMP0  
        MOV @#$ERRPC,@#$TMP1 ;GET PC OF CALL  
        MOV R0,-(SP) ;SAVE R0
```

```

6026
6027 036050 113700 001114          MOVB    @#$ITEMB,RO          ;GET THE ITEM NUMBER
6028 036054 042700 177400          BIC     #177400,RO
6029 036060 001007                BNE     MULT
6030 036062 104401 036351          PCTYP: TYPE    ,EMSG
6031 036066 013746 001116          MOV     @#$ERRPC,-(SP)      ;IF ZERO THEN JUST
6032 036072 104402                TYPOC                    ;PRINT THE PC
6033 036074 000137 036130          JMP     @#ERT5
6034
6035 036100 005300                MULT:  DEC     RO          ;OTHERWISE MULT RO BY 2 TO
6036                                     ;GET ERROR MESSAGE POINTER
6037 036102 006300                ASL     RO
6038 036104 062700 001346          ADD     #$ERRTB,RO
6039 036110 011037 036120          MOV     (RO),@#2$          ;PICK UP ADDRESS OF ERROR MESSAGE
6040 036114 001405                BEQ     ERT5
6041 036116 104401                TYPE
6042 036120 000000                2$:   .WORD   0
6043 036122 104401                TYPE
6044 036124 001313                .WORD   $CRLF
6045 036126 000755                BR      PCTYP
6046 036130 012600                ERT5:  MOV     (SP)+,RO      ;RESTORE RO.
6047 036132 000207                RTS     PC                  ;AND RETURN.
6048
6049
6050
6051
6052
6053          .SBTTL  FPP SPURIOUS TRAP TO 244 HANDLER
6054          ::*****
6055          ::*****
6056          ;*THIS ROUTINE HANDLES UNEXPECTED TRAPS TO THE FPP TRAP VECTOR AT 244.
6057          ;*THE LAST FPP INSTRUCTION EXECUTED AND ITS ADDRESS HAS BEEN RECORDED
6058          ;*THESE ALONG WITH THE FEC, FPS AND PC OF TRAP ARE REPORTED.
6059          ;*
6060          FPSPUR: MOV     (SP),@#$TMP2          ;SAVE PC OF TRAP.
6061                   CMP     (SP)+,(SP)+          ;RESTORE SP.
6062                   STFPS  RO                    ;GET FPS
6063                   MOV     RO,@#$TMP3
6064                   STST  RO                    ;GET FEC
6065          1$:   ERROR  1
6066                   RSETUP                    ;GO INITIALIZE THE FPS AND STACK; AND
6067                   ;SEE IF THE USER HAS EXPRESSED
6068                   ;THE DESIRE TO CHANGE THE SOFTWARE
6069                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
6070                   ;THE USER TYPED CONTROL G?).
6071                   JMP     @#$EOP
6072
6073          .SBTTL  CPU SPURIOUS TRAP TO 4 HANDLER
6074          ::*****
6075          ::*****
6076          ;*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 4.
6077          ;*
6078          CPSPUR: MOV     (SP),@#$TMP2          ;SAVE PC OF TRAP.
6079                   CMP     (SP)+,(SP)+
6080          1$:   ERROR  0
    
```



```

6076 036176 104413          RSETUP          ;GO INITIALIZE THE FPS AND STACK; AND
(1)                          ;SEE IF THE USER HAS EXPRESSED
(1)                          ;THE DESIRE TO CHANGE THE SOFTWARE
(1)                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
(1)                          ;THE USER TYPED CONTROL G?).
6077 036200 000137 032702    JMP          @#$EOP

        .SBTTL CPU SPURIOUS TRAP TO 10 HANDLER
        :*****
        :*****
        :*THIS ROUTINE REPORTS UNEXPECTED CPU TRAPS TO VECTOR 10.
        :*
CPTWO:  MOV      (SP),@#$TMP2          ;SAVE PC OF TRAP.
        CMP      (SP)+,(SP)+
1$:     ERROR    2
        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?).
6088 036216 000137 032702    JMP          @#$EOP

        .SBTTL SET LOOP ON ERROR ADDRESS ROUTINE
        :*****
        :*****
        :*
.LPER:  MOV      (SP),@#$LPERR
        RTI

        .SBTTL FLAG RESET AND CONSOLE TEST ROUTINE
        :*****
        :*****
        :*THIS ROUTINE WILL BE CALLED AT THE END OF EACH TEST TO
        :*RESET THE STACK, CLEAR THE FPS AND SEE IF THE USER HAS TYPED
        :*CONTROL G ON THE TERMINAL. IF THE USER HAS TYPED CONTROL G AND
        :*THERE IS NO PHYSICAL CONSOLE SWITCH REGISTER THEN THE CONTENTS
        :*OF THE SOFTWARE SWITCH REGISTER WILL BE TYPED IN OCTAL ON THE
        :*TELETYPE AND THE USER CAN MODIFY IT.
        :*
RSET:   CMP      @#$SWR,#177570        ;SEE IF THERE IS A PHYSICAL
        ;CONSOLE SWITCH REGISTER.
        BNE      1$                    ;BRANCH IF NO.
        CKSWR          ;OTHERWISE TYPE THE CONTENTS
        ;OF THE PROGRAM VIRTUAL SWITCH REGISTER
        ;AND GIVE THE USER A CHANCE TO
        ;MODIFY IT.
6116 036242 012737 036134 000244 1$:  MOV      #FPSPUR,@#FPVECT
6117 036250 012737 036166 000004      MOV      #CPSPUR,@#ERRVECT
6118 036256 012737 036204 000010      MOV      #CPTWO,@#10
6119 036264 011600          MOV      (SP),R0          ;SAVE RETURN ADDRESS.
6120 036266 012706 001100          MOV      #STACK,SP      ;RESET THE STACK POINTER.
  
```

CJKDDB KEF11-A DIAG PART 2
CJKDDB.P11 12-MAR-80 07:22

MACY11 30A(1052) 12-MAR-80 07:25 PAGE 1-144
FLAG RESET AND CONSOLE TEST ROUTINE

SEQ 0161

```

6121 036272 005004          CLR      R4          ;CLEAR THE FPS.
6122 036274 170104          LDFPS  R4
6123 036276 000110          JMP      (R0)        ;RETURN.
6124
6125
6126          .NLIST  BEX
6127
6128          ;THESE ARE SPECIAL MESSAGES:
6129
6130 036300 050200 053517 051105 POWERM: .ASCIZ <CRLF>'POWER FAILURE. PROGRAM RESTARTING.'
6131 036344 020040      000          SPACE: .ASCIZ ' '
6132 036347      011      000          $TAB: .ASCIZ <TAB>
6133 036351      106 047514 052101 EMSG: .ASCIZ /FLOATING POINT ERROR, STOPPED AT PC= /
6134 036417      120 047522 040502 EM1: .ASCIZ /PROBABLY BAD FP1 CHIP./
6135 036446 051120 041117 041101 EM2: .ASCIZ /PROBABLY BAD HYBRID FP CHIP./
6136 036503      116 020117 047111 EM3: .ASCIZ /NO INTERRUPT FROM SLU IN ALLOTTED TIME./
6137
6138          000001          .END

```


AABBF0	007070	1301	1305	1313	1319	1324	1337	1342#
AABDON	007120	1321	1327	1333	1339	1356#		
AABTP1	007100	1300	1347#					
AABTP2	007110	1325	1352#					
AAB1	006712	1299#						
AAB2	007014	1315	1324#					
AAB3	007034	1318	1330#					
AAB4	007052	1320	1336#					
AACDON	017242	3196	3207	3213	3228	3230#		
AACTP1	017146	3183	3192	3200#	3204			
AAC1	017066	3181#						
AAC10	017152	3193	3204#					
AAC11	017170	3195	3210#					
AAC2	017122	3186	3188#	3220	3222			
AAC20	017206	3187	3218#					
AADBF	032404	5851*	5852	5855	5859#			
AADDON	032410	5856	5862#					
AAD1	032330	5845#						
ABASE =	000000	80						
ACDW1 =	000000	80						
ACDW2 =	000000	80						
ACPUOP=	000000	80						
ADDW0 =	000000	80						
ADDW1 =	000000	80						
ADDW10=	000000	80						
ADDW11=	000000	80						
ADDW12=	000000	80						
ADDW13=	000000	80						
ADDW14=	000000	80						
ADDW15=	000000	80						
ADDW2 =	000000	80						
ADDW3 =	000000	80						
ADDW4 =	000000	80						
ADDW5 =	000000	80						
ADDW6 =	000000	80						
ADDW7 =	000000	80						
ADDW8 =	000000	80						
ADDW9 =	000000	80						
ADEVCT=	000000	80						
ADEVN =	000000	80						
AENV =	000000	80						
AENVN =	000000	80						
AFATAL=	000000	80						
AMADR1=	000000	80						
AMADR2=	000000	80						
AMADR3=	000000	80						
AMADR4=	000000	80						
AMAMS1=	000000	80						
AMAMS2=	000000	80						
AMAMS3=	000000	80						
AMAMS4=	000000	80						
AMSGAD=	000000	80						
AMSGLG=	000000	80						
AMSGTY=	000000	80						
AMTYP1=	000000	80						
AMTYP2=	000000	80						

AMTYP3=	000000	80																		
AMTYP4=	000000	80																		
APASS =	000000	80																		
APRIOR=	000000	80																		
APTCSU=	000040	5993	5999#																	
APTENV=	000001	5989	5993	5999#																
APTSIZ=	000200	90	5999#																	
APTSPO=	000100	5993	5999#																	
ASWREG=	000000	80																		
AESTN=	000000	80																		
AUNIT =	000000	80																		
AUSWR =	000000	80																		
AVECT1=	000000	80																		
AVECT2=	000000	80																		
BBBDON	017420	3251	3264	3270	3288#															
BBCTP1	017326	3237	3247	3256#	3261															
BBC1	017246	3235#																		
BBC10	017332	3248	3261#																	
BBC11	017350	3250	3267#																	
BBC2	017302	3240	3243#	3277	3280															
BBC20	017366	3241	3275#																	
BIT0 =	000001	67#																		
BIT00 =	000001	67#																		
BIT01 =	000002	67#																		
BIT02 =	000004	67#																		
BIT03 =	000010	67#																		
BIT04 =	000020	67#																		
BIT05 =	000040	67#																		
BIT06 =	000100	67#																		
BIT07 =	000200	67#																		
BIT08 =	000400	67#	5987																	
BIT09 =	001000	67#	5987	5989																
BIT1 =	000002	67#																		
BIT10 =	002000	67#	5989																	
BIT11 =	004000	67#	5987																	
BIT12 =	010000	67#	5973																	
BIT13 =	020000	67#	5989																	
BIT14 =	040000	67#	5987																	
BIT15 =	100000	67#																		
BIT2 =	000004	67#																		
BIT3 =	000010	67#																		
BIT4 =	000020	67#																		
BIT5 =	000040	67#																		
BIT6 =	000100	67#	5891	5905	5923															
BIT7 =	000200	67#																		
BIT8 =	000400	67#																		
BIT9 =	001000	67#																		
BPTVEC=	000014	67#																		
CCBDON	007220	1376	1383	1389#																
CCB1	007124	1364#																		
CCB10	007166	1373	1379#																	
CCB15	007204	1375	1386#																	
CCB2	007142	1367	1368#																	
CKSWR =	104407	5987	5989	6004#	6112															
CPSPUR	036166	388	488	566	640	1555	1635	1714	1789	1866	1944	2021	2094	2162						
		3224	3282	3325	3373	3422	3476	3531	4703	4765	4827	4891	4956	5022						

FFBDON	010316	1609	1640	1647	1653	1660#
FFBTP1	010142	1585	1612#	1644		
FFBTP2	010152	1616#	1643			
FFB1	010032	1584#				
FFB10	010202	1594	1630#			
FFB15	010236	1602	1643#			
FFB2	010102	1593	1596#	1631	1633	
FFB20	010264	1606	1650#			
FFB25	010302	1608	1656#			
FFCDON	020226	3393	3407	3413	3426#	
FFCTP1	020120	3381	3382*	3398#	3400	
FFCTP2	020130	3380	3389	3400#	3404	
FFC1	020030	3379#	3385			
FFC10	020140	3390	3404#			
FFC11	020156	3392	3410#			
FFC2	020074	3387#	3418	3420		
FFC20	020174	3386	3417#			
FPSPUR	036134	130	706	777	6058#	6116
FPVECT=	000244	61#	110*	6116*		
GGBBF0	010452	1665	1678	1684	1699#	1724 1729
GGBBF1	010462	1671	1703#			
GGBDON	010606	1688	1719	1726	1732	1739#
GGBTP1	010432	1664	1691#	1723		
GGBTP2	010442	1695#	1722			
GGB1	010322	1663#				
GGB10	010472	1673	1709#			
GGB15	010526	1681	1722#			
GGB2	010372	1672	1675#	1710	1712	
GGB20	010554	1685	1729#			
GGB25	010572	1687	1735#			
GGCDON	020444	3443	3455	3461	3467	3480#
GGCTP1	020324	3429	3430*	3439	3448#	3452
GGC1	020232	3428#				
GGC10	020336	3440	3452#			
GGC11	020354	3442	3458#			
GGC2	020272	3432	3435#	3464	3465	3472 3474
GGC20	020412	3434	3471#			
GGC25	020372	3438	3464#			
GNS =	***** U	79	91	6004	6005	6006
GTSWR =	104406	91	6004#			
HHBBF0	010752	1744	1757	1772	1774#	1799
HHBBF1	010762	1750	1762	1778#	1804	
HHBDON	011106	1766	1794	1801	1807	1813#
HHBTP1	010722	1743	1769#	1798		
HHBTP2	010742	1773#	1797			
HHB1	010612	1742#				
HHB10	010772	1752	1784#			
HHB15	011026	1760	1797#			
HHB2	010662	1751	1754#	1785	1787	
HHB20	011054	1763	1804#			
HHB25	011072	1765	1809#			
HHCDON	020676	3498	3510	3516	3522	3535#
HHCTP1	020550	3484	3485*	3503#		
HHCTP2	020560	3483	3494	3504#	3507	
HHC1	020450	3482#				
HHC10	020570	3495	3507#			

HHC11	020606	3497	3513#					
HHC2	020516	3487	3490#	3519	3520	3527	3529	
HHC20	020644	3489	3526#					
HHC25	020624	3493	3519#					
HT	= 000011	67#	5993					
IIBBF0	011252	1818	1831	1846	1851#	1876		
IIBBF1	011262	1824	1836	1855#	1881			
IIBDON	011406	1840	1871	1878	1884	1890#		
IIBTP1	011222	1817	1843#	1875				
IIBTP2	011242	1847#	1874					
IIB1	011112	1816#						
IIB10	011272	1826	1861#					
IIB15	011326	1834	1874#					
IIB2	011162	1825	1828#	1862	1864			
IIB20	011354	1837	1881#					
IIB25	011372	1839	1886#					
IICDON	021012	3561	3572	3580#				
IIC1	020702	3547#						
IIC2	020730	3548	3554#	3565	3570			
IIC20	021002	3549	3576#					
IIC3	020750	3565#						
IOTVEC=	000020	67#	90*					
JJBBF0	011546	1896	1902	1909	1914	1929#	1954	1959
JJBBF1	011556	1933#						
JJB DON	011702	1918	1949	1956	1962	1967#		
JJBTP1	011524	1895	1921#	1953				
JJBTP2	011536	1925#	1952					
JJB1	011412	1894#						
JJB10	011566	1904	1939#					
JJB15	011622	1912	1915	1952#				
JJB2	011462	1903	1906#	1940	1942			
JJB20	011650	1917	1959#					
JJB25	011666	1964#						
KKBBF0	012050	1972	1985	2000	2006#	2031		
KKBBF1	012060	1978	1990	2010#	2036			
KKBDON	012204	1994	2026	2033	2039	2045#		
KKBTP1	012020	1971	1997#	2030				
KKBTP2	012040	2002#	2029					
KKB1	011706	1970#						
KKB10	012070	1980	2016#					
KKB15	012124	1988	2029#					
KKB2	011756	1979	1982#	2017	2019			
KKB20	012152	1991	1993	2036#				
KKB25	012170	2041#						
KKCDON	022522	3918	4015#					
KKC1	021154	3640#						
KKC10	021572	3768#						
KKC11	021630	3782#						
KKC12	021666	3796#						
KKC13	021724	3810#						
KKC14	021762	3824#						
KKC15	022020	3838#						
KKC16	022056	3852#						
KKC17	022114	3866#						
KKC18	022152	3880#						
KKC19	022210	3893#						

OOC DON	026042	4733	4743	4749	4756	4771#			
OOC TBO	025724	4716	4736#						
OOC TB1	025730	4724	4727	4729	4731	4737#	4741	4747	4754
OOC1	025624	4715#							
OOC10	025740	4728	4740#						
OOC15	025756	4730	4746#						
OOC2	025672	4721	4726#	4763					
OOC20	025776	4732	4753#						
OOC25	026016	4722	4762#						
O OODON	002214	124	147	150	154#				
O OOT	002124	110	127#						
O O01	002060	106#							
O O02	002104	117#	128						
O O03	002140	129	133#						
O O04	002206	144	149#						
PCTYP	036062	6030#	6045						
PIRQ =	177772	67#							
PIRQVE =	000240	67#							
POWERM	036300	6009	6130#						
PPBDON	013542	2330	2346	2352	2359#				
PPBTP1	013450	2306	2311	2320	2326	2333#	2343	2349	
PPBTP2	013460	2307	2321	2337#	2344				
PPB1	013336	2304#							
PPB10	013470	2324	2343#						
PPB15	013510	2327	2349#						
PPB2	013404	2313	2317#						
PPB20	013526	2329	2355#						
PPCDON	026264	4795	4805	4811	4818	4833#			
PPCTBO	026146	4778	4798#						
PPCTB1	026152	4786	4789	4791	4793	4799#	4803	4809	4816
PPC1	026046	4777#							
PPC10	026162	4790	4802#						
PPC15	026200	4792	4808#						
PPC2	026114	4783	4788#	4825					
PPC20	026220	4794	4815#						
PPC25	026240	4784	4824#						
PPPBF0	002356	169	206#						
PPPBF1	002372	179	187	194	208#	215	221		
PPPDON	002454	192	204	217	224#				
PPPTP1	002406	176	195	210#	214	220			
PPP1	002220	166#							
PPP10	002416	197	199	214#					
PPP15	002436	201	203	220#					
PPP2	002236	171#	172						
PPP3	002300	182	185#						
PPP4	002320	188	194#						
PROGNU =	000003	5#							
PRO =	000000	67#							
PR1 =	000040	67#							
PR2 =	000100	67#							
PR3 =	000140	67#							
PR4 =	000200	67#							
PR5 =	000240	67#							
PR6 =	000300	67#							
PR7 =	000340	67#							
PS =	177776	67#	5886*						

TST17	007222	1399#		
TST2	002216	164#		
TST20	007322	1437#		
TST21	007540	1503#		
TST22	010030	1582#		
TST23	010320	1661#		
TST24	010610	1740#		
TST25	011110	1814#		
TST26	011410	1892#		
TST27	011704	1968#		
TST3	002456	234#		
TST30	012206	2046#		
TST31	012450	2113#		
TST32	012722	2187#		
TST33	013124	2246#		
TST34	013334	2303#		
TST35	013544	2360#		
TST36	013766	2420#		
TST37	014206	2480#		
TST4	003040	327#		
TST40	014430	2538#		
TST41	014640	2594#		
TST42	015066	2652#		
TST43	015330	2720#		
TST44	015602	2793#		
TST45	017064	3178#		
TST46	017244	3233#		
TST47	017422	3291#		
TST5	003362	428#		
TST50	017616	3330#		
TST51	020026	3378#		
TST52	020230	3427#		
TST53	020446	3481#		
TST54	020700	3545#		
TST55	021014	3589#		
TST56	021152	3635#		
TST57	022524	4023#		
TST6	003672	515#		
TST60	023532	4263#		
TST61	025400	4650#		
TST62	025622	4712#		
TST63	026044	4775#		
TST64	026266	4837#		
TST65	026520	4901#		
TST66	026754	4965#		
TST67	027214	5038#		
TST7	004142	591#		
TST70	027466	5114#		
TST71	027570	5146#		
TST72	027672	5182#		
TST73	031334	5562#		
TST74	031404	5589#		
TST75	032116	5775#		
TST76	032326	5843#		
TST77	032416	5873#		
TBBDON	014636	2564	2580	2586 2593#

UUJTP1	004310	602	611	629#	655				
UUU1	004144	593#							
UUU10	004320	604	635#						
UUU11	004342	637	639	642#					
UUU15	004354	609	648#						
UUU2	004230	605	607#	636	638				
UUU20	004372	614	654#						
UUU3	004254	613#	615						
VVB DON	007320	1415	1421	1428#					
VVB1	007224	1401#							
VVB10	007266	1412	1418#						
VVB15	007304	1414	1424#						
VVB2	007242	1404	1406#						
VVCBF0	027662	5154	5158	5163	5169#				
VVC DON	027670	5159	5165	5171#					
VVCTP1	027650	5151	5167#						
VVC1	027572	5148#							
VVC2	027622	5153	5156#						
VVC3	027632	5162#							
VVVBFO	004534	669	678	682	684	691#	715	720	
VVVDON	004644	690	711	717	723#				
VVVT P1	004544	675	685	695#	721				
VVV1	004414	666#							
VVV10	004554	674	701#						
VVV11	004576	703	705	708#					
VVV15	004610	683	714#						
VVV2	004476	677	680#	702	704				
VVV20	004626	688	720#						
WVBDON	017062	3011	3168#						
WV B1	015604	2795#							
WV B2	015662	2819#							
WV B3	015740	2843#							
WV B4	016016	2867#							
WV B5	016074	2891#							
WV B6	016152	2915#							
WV B7	016230	2939#							
WV B8	016306	2963#							
WV B9	016364	2987#							
WVCDON	031332	5445	5552#						
WV C1	027674	5185#							
WV C10	030334	5309#							
WV C11	030400	5324#							
WV C12	030444	5339#							
WV C13	030510	5354#							
WV C14	030554	5369#							
WV C15	030620	5383#							
WV C16	030664	5399#							
WV C17	030730	5416#							
WV C18	030774	5432#							
WV C2	027740	5202#							
WV C4	030004	5217#							
WV C5	030050	5232#							
WV C6	030114	5248#							
WV C7	030160	5263#							
WV C8	030224	5278#							
WV C9	030270	5294#							

WwBF0	004776	735	746	751	758#	786	791		
WwBF1	005016	744	746*	749	766#				
WwDON	005116	757	782	788	794#				
WwTP1	005006	741	752	762#	792				
Ww1	004650	732#							
Ww10	005026	740	772#						
Ww11	005050	774	776	779#					
Ww15	005062	750	785#						
Ww2	004740	743	747#	773	775				
Ww20	005100	755	791#						
XXBDON	015326	2681	2698	2704	2710	2718#			
XXBTP1	015214	2654	2659	2660*	2671	2677	2684#	2701	2707
XXBTP2	015224	2655	2672	2688#	2702				
XXB1	015070	2653#							
XXB10	015254	2675	2701#						
XXB15	015274	2678	2707#						
XXB2	015142	2661	2666#	2695	2696				
XXB20	015312	2680	2714#						
XXB25	015234	2670	2695#						
XXCDON	031402	5580#							
XXC1	031336	5566#							
XXXDON	005752	917	1029#						
XXX1	005122	804#							
XXX2	005176	827#							
XXX3	005252	850#							
XXX4	005326	873#							
XXX5	005402	896#							
Y	032652	5896	5919#						
YYBDON	015600	2751	2768	2774	2780	2787#			
YYBTP1	015464	2723	2729	2730*	2741	2754#	2762	2771	
YYBTP2	015474	2724	2742	2758#	2772				
YYBTP3	015504	2728	2747	2762#	2777				
YYB1	015332	2722#							
YYB10	015526	2745	2771#						
YYB15	015546	2748	2777#						
YYB2	015412	2731	2736#	2765	2766				
YYB20	015564	2750	2783#						
YYB25	015506	2740	2765#						
YYCDON	032114	5679	5763#						
YYC1	031406	5592#							
YYC2	031444	5606#							
YYC3	031502	5622#							
YYC4	031540	5637#							
YYC5	031576	5652#							
YYC6	031634	5667#							
YYYDON	006606	1152	1258#						
YYY1	005756	1039#							
YYY2	006032	1062#							
YYY3	006106	1085#							
YYY4	006162	1108#							
YYY5	006236	1131#							
Z	032654	5897*	5915*	5916	5920#				
ZZCBF	032306	5783	5790	5827#					
ZZCDON	032320	5804	5809	5815	5818	5823	5830#		
ZZC1	032120	5777#							
ZZC10	032260	5801	5812#						

ZZC12	032274	5813	5816#						
ZZC15	032300	5803	5821#						
ZZC2	032130	5781#	5794	5800					
ZZC3	032156	5786	5787#						
ZZC5	032254	5799	5807#						
ZZDDON	032656	5913	5922#						
ZZD1	032420	5875#							
ZZD2	032436	5876	5879#						
ZZZDON	006706	1279	1285	1291#					
ZZZ1	006612	1266#							
ZZZ10	006654	1275	1282#						
ZZZ15	006672	1278	1288#						
ZZZ2	006630	1269	1270#						
\$APTHD	001354	87#							
\$ASTAT=	***** U	5999							
\$ATYC	034666	5999#							
\$ATY1	034642	5999#							
\$ATY3	034650	5993	5999#						
\$ATY4	034660	5989	5999#						
\$AUTOB	001134	80#	91*	6002					
\$BDADR	001122	80#							
\$BDDAT	001126	80#							
\$BELL	001306	80#	5989						
\$CHARC	034164	5993#*							
\$CKSWR	035110	6002#	6004						
\$CLR.T	032776	5957	5960#						
\$CMTAG	001100	80#	90						
\$CM1 =	000024	80#							
\$CM2 =	000050	80#							
\$CM3 =	000024	80#							
\$CM4 =	000024	80#							
\$CNTLG	035517	6002#							
\$CNTLU	035512	6002#							
\$CPUOP	001344	80#							
\$CRLF	001313	80#	5989	5993	6002	6021	6044		
\$DBLK	034632	5997#							
\$DEVCT	001326	80#							
\$DOAGN	033046	5946	5971#						
\$DTBL	034622	5997#							
\$ENDAD	033000	86	91	5961#	5989				
\$ENDCT	032736	90	5948#						
\$ENDMG	033115	5950	5985#						
\$ENULL	033112	5953	5984#						
\$ENV	001336	80#	91	5875	5989	5993	5999		
\$ENVM	001337	80#	90	5993	5999				
\$EOP	032702	5878	5939#	6066	6077	6088			
\$EOPCT	032730	90*	5945#	5949					
\$ERFLG	001103	80#	5987*	5989*					
\$ERMAX	001115	80#	90*	5987*					
\$ERROR	033412	90	5989#						
\$ERRPC	001116	80#	5989*	6024	6031				
\$ERRTB	001346	80#	6038						
\$ERTTL	001112	80#	5989*						
\$ESCAP	001304	80#	90*	5987*	5989				
\$ETABL	001336	80#							
\$ETEND	001346	80#	87						

CJKDDB KEF11-A DIAG PART 2
CJKDDB.P11 12-MAR-80 07:22

MACY11 30A(1052) 12-MAR-80 07:25 PAGE 3-2
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0184

.SCATC	52#	79
.SCMTA	52#	80
.SEOP	53#	
.SERRO	53#	5989
.SPOWE	54#	6009
.SREAD	55#	6002
.SSAVE	53#	5991
.SSCOP	53#	5987
.STRAP	54#	6004
.STYPD	54#	5997
.STYPE	53#	5993
.STYPO	53#	5995

. ABS. 036553 000

ERRORS DETECTED: 0

CJKDDB.BIN,CJKDDB.LST/CRF=CJKDDB.P11
RUN-TIME: 91 66 7 SECONDS
RUN-TIME RATIO: 248/166=1.4
CORE USED: 27K (53 PAGES)