

IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-8194C-MC
PROJECT NAME: CVKACCO LSI-11 FIS INST
DATE CREATED: AUGUST 22, 1975
MAINTAINER: DIAGNOSTIC GROUP

This software is furnished under a license for use only on a single computer system and may be copied only with the inclusion of the above copyright notice. This software, or any other copies thereof, may not be provided or otherwise made available to any other person except for use on such system and to one who agrees to these license terms. Title to and ownership of the software shall at all times remain in DEC.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

Copyright (C) 1975, 1978 by Digital Equipment Corporation
Maynard, Mass.

CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
 - 2.1 Equipment
 - 2.2 Storage
 - 2.3 Preliminary Programs
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
 - 4.1 Control Switch Settings
 - 4.2 Starting Address
 - 4.3 Program And/or Operator Action
- 5.0 OPERATING PROCEDURE
 - 5.1 Operational Switch Settings
 - 5.2 Subroutine Abstracts
- 6.0 ERRORS
 - 6.1 Error Printout
 - 6.2 Error Recovery
 - 6.3 Error Counter
- 7.0 RESTRICTIONS
- 8.0 MISCELLANEOUS
 - 8.1 Execution Time
 - 8.2 Stack Pointer
 - 8.3 Pass Counter
 - 8.4 Test Number
 - 8.5 Power Fail
- 9.0 PROGRAM DESCRIPTION

1.0 ABSTRACT

This program tests the LSI-11 floating instruction set <FADD, FSUB, FMUL, and FDIV> option with fixed number patterns, using each register at least once as the stack pointer. It also checks stack overflow and that the floating instructions can be interrupted (by the console teletype). [However, this test will not be executed when bit 5 of \$ENVM byte is high]. The program should be run for at least 2 passes with all switches low. The program is designed to run under APT, and ACT, systems. When running under APT, with bit 5 of \$ENVM low it will be required to have a SLU with TTY registers having addresses of 176560-66 and interrupt vectors of 70 for receiver and 74 for transmitter. Under such a condition it will also be required to change the run time of first pass from 5 seconds to the time given in Sec. 8.1, and the run time for the longest test from 3 seconds to 30 seconds.

2.0 REQUIREMENTS

LTC SWITCH MUST BE IN OFF POSITION TO RUN DIAGNOSTIC.

2.1 Equipment

LSI-11 standard computer with FIS option and 4K of memory.

2.2 Storage

Program Storage - The routines use memory 0 - 17500.

2.3 Preliminary Programs

None.

3.0 LOADING PROCEDURE

Use standard procedure for ABS Tapes.

4.0 STARTING PROCEDURE

4.1 Control Switch Settings

See 5.1.1 (all low for worst case testing).

4.2 Starting Address

After loading the program it should always be started at 200. If it is desired to save the pass counter then the program should be restarted at location RESTRT (i.e. 222) otherwise the program can be restarted at 200.

4.3 Program And/or Operator Action

4.3.1 Stand Alone -

1. Load program into memory using ABS loader.
2. Set switches (see Sec 5.1.1) all low except bit 7 for worst case.
3. Type 200G.
4. The program will loop and 'END PASS' will be typed after completion of every pass. However type out will be suppressed if bit 5 of location \$ENVM is high.
5. A minimum of two passes should always be run.

4.3.2 Under Apt - Load the program, set the switches (see Sec. 5.1.1) and start. When under APT, with bit 5 of \$ENVM low it will be required to have a SLU with IIV registers having addresses of 176560-66 and interrupt vectors of 70 for receiver and 74 for transmitter. Under such a condition it will also be required to change the run time of first pass from 5 seconds to the time given in Sec. 8.1, and the run time for the longest test from 3 seconds to 30 seconds. The test times and pass times are suggested with bit 7 of \$SWREG, low, if it is desired to enable the iterations then the times should be multiplied by a factor of 256.

5.0 OPERATING PROCEDURE

5.1 Operational Switch Settings

All switches low except SW<11> is worst case testing. With bit 11 of the location \$SWREG (i.e. location 422), high each subtest will be looped upon until completion of 256 passes of that subtest. 'END PASS' will be typed upon completion of a pass of the entire program. Alternate pass will run with the T-bit set.

5.1.1 Switch Settings Are - A 16 bit location called \$SWREG (i.e. location 422) has been used to give the following options by inserting a 1 in their respective positions.

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	Halt On Error
14	040000.....	Scope Loop
13	020000.....	Inhibit Printout
12	010000.....	Inhibit Trace Trapping
11	004000.....	Enable Iterations Of Subtest
10	002000.....	Bell On Error
09	001000.....	Loop On Error
08	0004XX.....	Loop On Test In Bits 7 Thru 0

An 8 bit byte \$ENVM (i.e. location 421) has been used to define the operating mode. All typeouts can be suppressed by making bit 5 of byte \$ENVM high, in other words by placing a 20000 in location 420.

5.2 Subroutine Abstracts

5.2.1 Scope - This subroutine call (via a TRAP instruction) is placed between each subtest in the instruction section. It records the starting address of each subtest as it is being entered in location 'LA \$'. If a scope loop is requested, the current subtest will be looped upon. SW<11> is a 1 inhibits iteration of subtests. The contents of 'LADS\$' may be used to determine the last subtest successfully completed.

5.2.2 HLT - This routine (called by an EMT instruction) prints out an error message (see 6.1). If SW<9> is a 1 and a HLT is executed, the subtest will be looped upon until 256 consecutive good passes are completed. To inhibit typeouts, make SW<13> a 1. To ring the bell on an error, make SW<10> a 1. A high bit 5 in location \$ENVM will inhibit any typeouts and ringing of bells.

5.2.3 T Bit Trap - If SW<12> is a 0, the T-Bit will be set on alternate passes. When the T-Bit is set, the processor traps after each instruction. The first instruction executed upon trapping is an 'RTT' which returns to the interrupted sequence of instructions. This sequence is continued until the end of the program is reached.

5.2.4 Trap Catcher - A '+2' - 'HALT' sequence is repeated from 0-776 to catch any unexpected traps. Thus any unexpected traps or interrupts will halt at the vector +2.

5.2.5 Floating Error Trap (To 244) - If a floating point error (overflow, underflow, or divide by zero) was expected, the vector will point to a unique ISR within the subtest where the error occurred which checks the data on the stack(s). If an error was not anticipated, an erroneous trap will be detected in traper.

5.2.6 NOP - A NOP is placed just before each FIS instruction. This allows the operator to patch in a halt for debugging purposes.

6.0 ERRORS

6.1 Error Printout

The format is as follows:

```
ERRNM  ADR  PS  SP  ANS1  ANS2  ANS3  ANS4  ANS5  ANS6
```

Where:

```
ERRNM ; - Error Number
ADR   ; - Address of Error HLT
PS    ; - Processor Status
SP    ; - Contents of Stack Pointer Register
ANS1-6 - Error Data Read from the Stack(s). From 0 to 6 of
        these may be typed depending on the number following
        the HLT; e.g., HLT+3 would type ANS1 thru ANS3, HLT
        (by itself) would stop after ERRNM, ADR, PS, and SP.
```

To find the failing test, loop at the listing above the address typed. In most cases the comment beside the HLT tells what was being checked and what was expected. All printouts will be suppressed when bit 5 of location \$ENVM is high. While running under APT the diagnostic will not support spooling of console outputs.

6.2 Error Recovery

Restart at 200 or 222 (see Sec. 4.2).

6.3 Error Counter

An error count is kept in location 'ERRORS'. It can only be cleared from the console or by reloading the program.

7.0 RESTRICTIONS

None.

8.0 MISCELLANEOUS

8.1 Execution Time

Due to the random characteristics of the interrupt tests, the execution time can be half a minute or more. However, normally 'END PASS' will be typed within 40 seconds with all switches down. Execution time will increase by a factor of 256 when iterations of each subtest are enabled.

8.2 Stack Pointer

Stack is initially set to 600.

8.3 Pass Count

A 16 bit location '\$PASS' (i.e. location 406) is used to keep pass count. It can be cleared by restarting the program at 200.

8.4 Test Number

A 16 bit location '\$TESTN' (i.e. location 404) is used to keep track of the test number. Upper byte of this location gives the iteration number and the lower byte the test that was being executed.

0008.5 Power fail

Each test can be power failed with no errors. To use, start the test as usual and power down then up at any time. The program should type "POWER" and continue to run from where the power fail interrupted with no other error typeouts.

0009.0 PROGRAM DESCRIPTION

This program tests all the FIS instructions of the LSI-11 (FADD, FSUB, FMUL, and FDIV). All registers are checked to see if they function properly as the stack pointer. The program has many subtests (the code between 2 scope statements) which are run once before continuing to the next subtest. SW<11> set to a 1 causes each subtest to be run 00256 times SW<9> set to a 1 enables loop on error. The location \$ICNT contains the iteration count and the location \$TESTN contains the test number. All the subtests should be run sequentially by starting at 00200 not by starting at the beginning of the subtest. To loop on a particular subtest, put the test number (see listing) in the right byte of the location \$SWREG and SW<8> set to a 1. This test will be looped upon until SW<8> is set to a 0 or the right byte is changed. If the test is non-existent, the program will be run as usual.

.ENDR

CVKACC MACY11 30A(1052) 21-AUG-78 15:28
 CVKACC.P11 16-AUG-78 08:41 TABLE OF CONTENTS

00100			SWITCH OPTIONS AND ASSIGNMENTS
00200			ACT11 HOOKS
00300			VECTOR AREA, STACKS, ANSWER AREA, AND SETUP ROUTINE
00400	32	55100	APT MAILBOX-ETABLE
00500	81		APT PARAMETER BLOCK
00600	91	60500	STARTING OF THE PROGRAM
00700	94		FADD TEST SECTION
00800	126		TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW
00900	193	38300	TEST FLOATING ADD INSTRUCTION WITH OVERFLOW
01000	226	41700	FSUB TEST SECTION
01100	879	43500	TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW
01200	1024	44000	TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW
01300	1169	44500	FMUL TEST SECTION
01400	1740	46000	TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW
01500	1813	46400	TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW
01600	1886	46800	FDIV TEST SECTION
01700	2271	48100	TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW
01800	2344	48500	TEST FLOATING DIV. INSTRUCTION WITH OVERFLOW
01900	2417	48900	TEST OF ALL FIS AT ONCE
02000	2656	49800	ADDRESS ERROR TEST
02100	2729	50200	INTERUPT ABORT TEST SECTION
02200	2802	50600	END OF PASS ROUTINE
02300	2953	51200	SCOPE ROUTINE
02400	3016	51700	PUSH AND POP SUBROUTINES
02500	3145	52500	HLT ROUTINE (ERROR TYPEOUT)
02600	3369		USER ERROR ROUTINE
02700	3408		OCTAL WORD & ADDRESS TYPER
02800	3433	54700	POWER DOWN AND UP ROUTINES
02900	3567		ASCII TYPE OUT ROUTINE
03000	3600	72800	
03100	3613		
03200	3648		
03300	3686	74400	

00100
00200
00300
00400

CVKACC MACV11 30A(1052) 21-AUG-78 15:28 PAGE 2
CVKACC.P11 16-AUG-78 08:41

K 1

SEQ 0010

31700

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 3
 CVKACC.P11 16-AUG-78 08:41

00100		
00200		
00300		
00400	2	31900
00500	3	32000
00600	4	
00700	5	
00800	6	
00900	7	
01000	8	
01100	9	
01200	10	
01300	11	
01400	12	
01500	13	
01600	14	000001
01700	15	160000
01800	16	
01900	17	32900
02000	18	33600
02100	19	34000
02200	20	35400
02300	21	35500
02400	22	36900
02500	23	40200
02600	24	40700
02700	25	41200
02800	26	42000
02900	27	45100
03000	28	49600
03100	29	54900
03200	30	55300
03300	31	
03400	32	
03500	33	
03600	34	
03700	35	
03800	36	
03900	37	
04000	38	
04100	39	
04200	40	
04300	41	
04400	42	
04500	43	
04600	44	55500
04700	45	55600
04800	46	55700
04900	47	55800
05000	48	55900
05100	49	56000
05200	50	56100
05300	51	56200
05400	52	56300
05500	53	56400
05600	54	56500
05700	55	56600
05800	56	56700
05900	57	56800

```

.ABS
.TITLE CVKACC
.*COPYRIGHT (C) AUGUST 1978
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY DIAGNOSTIC ENGINEERING
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-B), JULY 11, 1975.
.*
$TN-1
$SWR-160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
  
```

SWITCH	USE
8	LOOP ON TEST IN SW<7:0>
9	LOOP ON ERROR
10	1 - BELL ON ERROR
11	INHIBIT ITERATIONS
12	INHIBIT TRACE TRAP
13	INHIBIT ERROR TYPEOUTS
14	LOOP ON TEST
15	HALT ON ERROR

```

:ERROR MESSAGE FORMAT:
:  ERRNM  ADR  PSW  SP  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6
:
:WHERE  ERRNM-  ERROR NUMBER
:       ADR -  ADDRESS OF 'HLT' INSTRUCTION + 2
:       PSW -  PROCESSOR STATUS WORD
:       SP -  STACK POINTER
:       ANS1 THRU ANS6 -  DATA OFF THE STACK(S)
:NOTE:  ANS1 THRU ANS6 ARE NOT ALWAYS TYPED, DEPENDING ON THE
:       NUMBER ADDED TO THE 'HLT'. 'HLT' ALONE TYPES NONE,
:       'HLT+1' TYPES ANS1, 'HLT+2' TYPES ANS1 AND ANS2, ETC.
  
```

```

HLT-  EMT
RO-   %0
  
```

104000
000000

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 4
00200 CVKACC.P11 16-AUG-78 08:41

SWITCH OPTIONS AND ASSIGNMENTS

SEQ 0012

00300			
00400	58	000001	56900
00500	59	000002	57000
00600	60	000003	57100
00700	61	000004	57200
00800	62	000005	57300
00900	63	000005	57400
01000	64	000006	57500
01100	65	000007	57600
01200	66	000024	57700
01300	67	104400	57800
01400	68	100000	57900
01500	69	040000	58000
01600	70	020000	58100
01700	71	010000	58200
01800	72	004000	58300
01900	73	002000	58400
02000	74	001000	58500
02100	75	000400	58600
02200	76	000004	58700
02300	77	000001	58800
02400	78	000001	58900
02500	79		
02600	80		59100
02700	81	000000	59200
02800	82		60000
02900	83		
03000	84		
03100	85		
03200	86		
03300	87	001000	
03400	88	000046	
03500	89	015664	
03600	90	000052	
03700	91	000052	
03800	92	001000	
03900	93		60200
04000	94		60300

R1	%1
R2-	%2
R3	%3
R4	%4
R5	%5
TTY=	%5
SP	%6
PC-	%7
PWRVEC	24
SCOPE=	TRAP
SW15=	100000
SW14-	40000
SW13-	20000
SW12=	10000
SW11-	4000
SW10	2000
SW09	1000
SW08=	400
TYPE-	IOT
N=	1
\$F	1

: 0 ;TRAP CATCHER FROM 0 - 776

```
.SBTTL ACT11 HOOKS
;HOOKS REQUIRED BY ACT11
;SAVE PC
.-46
;SENDAD ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
.52
;WORD 0 ;:2)SET LOC.52 TO ZERO
.-$VPC ;: RESTORE PC
```

CVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 5
CVKACC.P11 16-AUG-78 08:41

VECTOR AREA, STACKS, ANSWER AREA, AND SETUP ROUTINE

SEQ 0013

00300	95		
00400	96	000400	
00500	97		
00600	98		
00700	99		
00800	100		
00900	101		
01000	102		
01100	103		
01200	104	000400	
01300	105	000400	000000
01400	106	000402	000000
01500	107	000404	000000
01600	108	000406	000000
01700	109	000410	000000
01800	110	000412	000000
01900	111	000414	000000
02000	112	000416	000000
02100	113	000420	
02200	114	000420	000
02300	115	000421	000
02400	116	000422	000000
02500	117	000424	000000
02600	118	000426	000000
02700	119		
02800	120		
02900	121		
03000	122		
03100	123		
03200	124		
03300	125	000430	
03400	126		
03500	127		
03600	128		
03700	129		
03800	130		
03900	131		
04000	132	000430	
04100	133	000024	000024
04200	134	000024	000200
04300	135		000044
04400	136	000044	000430
04500	137		000430
04600	138		
04700	139		
04800	140		
04900	141		
05000	142	000430	
05100	143	000430	000000
05200	144	000432	000400
05300	145	000434	000003
05400	146	000436	000005
05500	147	000440	000000
05600	148	000442	000014
05700	149		000430
05800	150		
05900	151	000430	

60800
60900

400

.SBTTL APT MAILBOX-ETABLE

.EVEN

\$MAIL:			:: APT MAILBOX
\$MSGTY:	.WORD	AMSGTY	:: MESSAGE TYPE CODE
\$FATAL:	.WORD	AFATAL	:: FATAL ERROR NUMBER
\$TESTN:	.WORD	ATESTN	:: TEST NUMBER
\$PASS:	.WORD	APASS	:: PASS COUNT
\$DEVCT:	.WORD	ADEVCT	:: DEVICE COUNT
\$UNIT:	.WORD	AUNIT	:: I/O UNIT NUMBER
\$MSGAD:	.WORD	AMSGAD	:: MESSAGE ADDRESS
\$MSGLG:	.WORD	AMSGLG	:: MESSAGE LENGTH
\$ETABLE:			:: APT ENVIRONMENT TABLE
\$ENV:	.BYTE	AENV	:: ENVIRONMENT BYTE
\$ENVM:	.BYTE	AENVM	:: ENVIRONMENT MODE BITS
\$SWREG:	.WORD	ASWREG	:: APT SWITCH REGISTER
\$USWR:	.WORD	AUSWR	:: USER SWITCHES
\$CPUOP:	.WORD	ACPUOP	:: CPU TYPE, OPTIONS
*			BITS 15-11=CPU TYPE
*			11/04=01,11/05-02,11/20-03,11/40 04,11/45=05
*			11/70=06,PDQ=07,Q-10
*			BIT 10=REAL TIME CLOCK
*			BIT 9-FLOATING POINT PROCESSOR
*			BIT 8=MEMORY MANAGEMENT

\$ETEND:

.MEXIT

.SBTTL APT PARAMETER BLOCK

;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT

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

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

\$APTHD:			
\$HIBTS:	.WORD	0	:: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
\$MBADR:	.WORD	\$MAIL	:: ADDRESS OF APT MAILBOX (BITS 0-15)
\$TSTM:	.WORD	3	:: RUN TIME OF LONGEST TEST
\$PASTM:	.WORD	5	:: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VER
\$UNITM:	.WORD		:: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDI
	.WORD	\$ETEND-\$MAIL/2	:: LENGTH MAILBOX-ETABLE (WORD)
\$APTHD			

61200
61300

HLTAD\$:

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 7
CVKACC.P'1 16-AUG-78 08:41

194					
195					
196					
197					
198					
199		000200			
200	000200	000167	017104		
201	000204	000167	000370		
202					
203					
204					
205					
206		000600			
207					
208	000600	012706	000600		
209	000604	012737	000542	000014	
210	000612	012737	017170	000020	
211	000620	012737	017030	000024	
212					
213	000626	012700	000030		
214	000632	012720	016464		
215	000636	012720	000340		
216	000642	012720	015716		
217	000646	012710	000340		
218	000652	012737	000006	000004	
219	000660	132737	000007	000420	
220	000666	001410			
221	000670	012700	000554		
222	000674	012740	176566		
223	000700	012740	176564		
224	000704	012740	000074		
225	000710	005067	177470		
226	000714	005067	177550		
227					
228					

APT PARAMETER BLOCK

SEQ

.SBTTL STARTING OF THE PROGRAM

200
JMP NOOP ;GO TO TYPE HEADING & INITIALIZATION
RESTRT: JMP BEGIN ;START THE PROGRAM

600
BEGIN: MOV #BEGIN, SP ;INITIALIZE STACK POINTER
MOV #YESRT, @#14 ;SET TRACE TRAP VECTOR
MOV #STYPE, @#20 ;SET UP VECTOR 20
MOV #SPWRDN, @#24 ;SERVICE POWER DOWN ROUTINE FOR ANY FUTURE
;POWER DOWN
MOV #30, R0 ;SET R0 TO VECTOR 30
MOV #HLTS, (0)+ ;SET EMT VECTOR
MOV #340, (0)+
MOV #SCOPES, (0)+ ;SET TRAP VECTOR
MOV #340, (0)
1\$: MOV #6, @#4 ;RESTORE TIME-OUT VECTOR
BITB #1, @#SENV ;ARE WE UNDER APT ?
BEQ 2\$;IF NOT THEN GO TO 2\$
MOV #STPB+2, R0 ;OTHERWISE SET FOR THE OTHER SLU
MOV #176566, -(R0)
MOV #176564, -(R0)
MOV #74, -(R0)
2\$: CLR \$TESTN
CLR LAD\$;CLEAR LOOP ADDRESS

41500


```

00300
00400 229
00500 230
00600 231
00700 232
00800 233
00900 234
01000 235
01100 236 000720 104400
01200 237 000722 004567 015260
01300 238 000726 000000 000000
01400 239 000732 000000 000000
01500 240 000736 000000
01600 241 000740 016456 000340
01700 242 000744 012700 000510
01800 243
01900 244 000750 000240
02000 245 000752 075000
02100 246
02200 247 000754 004767 015260
02300 248 000760 010067 177450
02400 249 000764 122767 000004 177440
02500 250 000772 001402
02600 251 000774 104000
02700 252 000776 000001
02800 253
02900 254 001000 022767 000514 177426
03000 255 001006 001402
03100 256 001010 104000
03200 257 001012 000002
03300 258
03400 259 001014 005767 177416
03500 260 001020 001402
03600 261 001022 104002
03700 262 001024 000003
03800 263
03900 264 001026 005767 177406
04000 265 001032 001402
04100 266 001034 104002
04200 267 001036 000004
04300 268
04400 269 001040 122767 000001 177336
04500 270 001046 001402
04600 271 001050 104000
04700 272 001052 000005
04800 273
04900 274

```

```

*****
:TEST 1: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 000000,000000 + 000000,000000 = 000000,000000
: PS = 004, STACK POINTER = R0
*****

```

```

TST1: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
.WORD 000000,000000 ;SECOND OPERAND ON TOP
.WORD 000000,000000 ;FIRST OPERAND ON BOTTOM
.WORD 000 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR
MOV #STACK0,R0 ;SET UP STACK POINTER

NOP
FADD R0 ;FLOATING ADD ON THE R0 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R0, $SP ;SAVE 'STACK POINTER'
CMPB #004, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 004
1 ;THE ERROR NUMBER IS 1

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R0)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
2 ;THE ERROR NUMBER IS 2

TST ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 000000
3 ;THE ERROR NUMBER IS 3

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
4 ;THE ERROR NUMBER IS 4

END1: CMPB #1, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
5 ;THE ERROR NUMBER IS 5

```

```

00300      275
00400      276
00500      277
00600      278
00700      279
00800      280
00900      281
01000      282
01100      283 001054 104400
01200      284 001056 004567 015124
01300      285 001062 152525 052524
01400      286 001066 000000 000000
01500      287 001072 000040
01600      288 001074 016456 000340
01700      289 001100 012703 000510
01800
01900      290 001104 000240
02000      291 001106 075003
02100
02200      292
02300      293 001110 004767 015124
02400      294 001114 010367 177314
02500      295 001120 122767 000010 177304
02600      296 001126 001402
02700      297 001130 104000
02800      298 001132 000006
02900      299
03000      300 001134 022767 000514 177272
03100      301 001142 001402
03200      302 001144 104000
03300      303 001146 000007
03400      304
03500      305 001150 022767 152525 177260
03600      306 001156 001402
03700      307 001160 104002
03800      308 001162 000010
03900      309
04000      310 001164 022767 052524 177246
04100      311 001172 001402
04200      312 001174 104002
04300      313 001176 000011
04400      314
04500      315 001200 122767 000002 177176
04600      316 001206 001402
04700      317 001210 104000
04800      318 001212 000012
04900      319
           320

```

```

:*****
:TEST 2:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
:      000000,000000 + 152525,052524 = 152525,052524
:      PS = 010,      STACK POINTER = R3
:*****
TST2:  SCOPE
      JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
      .WORD   152525,052524      ;SECOND OPERAND ON TOP
      .WORD   000000,000000      ;FIRST OPERAND ON BOTTOM
      .WORD   040                ;PROCESSOR PRIORITY LEVEL
      .WORD   TRAPER,340         ;FIS TRAP VECTOR
      MOV     #STACK0,R3        ;SET UP STACK POINTER

      NOP
      FADD   R3                ;FLOATING ADD ON THE R3 STACK

      JSR   PC,      POPR      ;POP THE ANSWER
      MOV   R3,      $SP      ;SAVE 'STACK POINTER'
      CMPB #010,    $PSW      ;CHECK PS (EXCEPT T BIT)
      BEQ  .+6            ;BRANCH IF OK
      HLT  6              ;PS NOT EQUAL TO 010
                          ;THE ERROR NUMBER IS 6

      CMP   #STACK4,$SP      ;CHECK THE STACK POINTER (R3)
      BEQ  .+6            ;BRANCH IF OK
      HLT  7              ;STACK POINTER (R3) NOT EQUAL TO #STACK4
                          ;THE ERROR NUMBER IS 7

      CMP   #152525,ANS1     ;CHECK FIRST HALF OF ANSWER
      BEQ  .+6            ;BRANCH IF OK
      HLT+2 10            ;ANS1 NOT EQUAL TO 152525
                          ;THE ERROR NUMBER IS 10

      CMP   #052524,ANS2     ;CHECK SECOND HALF OF ANSWER
      BEQ  .+6            ;BRANCH IF OK
      HLT+2 11            ;ANS2 NOT EQUAL TO 052524
                          ;THE ERROR NUMBER IS 11

      END2: CMPB #2,      $TESTN ;CHECK THE TEST NUMBER
            BEQ  .+6            ;BRANCH IF OK
            HLT  12            ;WRONG TEST! PC MUST HAVE FOULED UP.
                          ;THE ERROR NUMBER IS 12

```

```

00300 321
00400 322
00500 323
00600 324
00700 325
00800 326
00900 327
01000 328 001214 104400
01100 329 001216 004567 014764
01200 330 001222 040200 000000
01300 331 001226 040200 000000
01400 332 001232 000040
01500 333 001234 016456 000340
01600 334 001240 012701 000510
01700 335
01800 336 001244 000240
01900 337 001246 075001
02000 338
02100 339 001250 004767 014764
02200 340 001254 010167 177154
02300 341 001260 105767 177146
02400 342 001264 001402
02500 343 001266 104000
02600 344 001270 000013
02700 345
02800 346 001272 022767 000514 177134
02900 347 001300 001402
03000 348 001302 104000
03100 349 001304 000014
03200 350
03300 351 001306 022767 040400 177122
03400 352 001314 001402
03500 353 001316 104002
03600 354 001320 000015
03700 355
03800 356 001322 005767 177112
03900 357 001326 001402
04000 358 001330 104002
04100 359 001332 000016
04200 360
04300 361 001334 122767 000003 177042
04400 362 001342 001402
04500 363 001344 104000
04600 364 001346 000017
04700 365
04800 366
04900

```

```

*****
:TEST 3: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 040200,000000 + 040200,000000 = 040400,000000
: PS = 000, STACK POINTER R1
*****

```

```

TST3: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
:SECOND OPERAND ON TOP
:WORD 040200,000000 ;FIRST OPERAND ON BOTTOM
:WORD 040200,000000 ;PROCESSOR PRIORITY LEVEL
:WORD 040 ;FIS TRAP VECTOR
MOV TRAPER,340 ;SET UP STACK POINTER
#STACK0,R1

NOP
FADD R1 ;FLOATING ADD ON THE R1 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R1, $SP ;SAVE 'STACK POINTER'
TSTB $PSW ;CHECK PS (EXCEPT T BJJ)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
13 ;THE ERROR NUMBER IS 13

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R1)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK4
14 ;THE ERROR NUMBER IS 14

CMP #040400,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 040400
15 ;THE ERROR NUMBER IS 15

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
16 ;THE ERROR NUMBER IS 16

END3: CMPB #3, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED JP.
17 ;THE ERROR NUMBER IS 17

```

```

00100
00200
00300
00400 367
00500 368
00600 369
00700 370
00800 371
00900 372
01000 373
01100 374 001350 104400
01200 375 001352 004567 014630
01300 376 001356 177777 177777
01400 377 001362 077777 177777
01500 378 001366 000100
01600 379 001370 016456 000340
01700 380 001374 012702 000510
01800 381
01900 382 001400 000240
02000 383 001402 075002
02100 384
02200 385 001404 004767 014630
02300 386 001410 010267 177020
02400 387 001414 122767 000004 177010
02500 388 001422 001402
02600 389 001424 104000
02700 390 001426 000020
02800 391
02900 392 001430 022767 000514 176776
03000 393 001436 001402
03100 394 001440 104000
03200 395 001442 000021
03300 396
03400 397 001444 005767 176766
03500 398 001450 001402
03600 399 001452 104002
03700 400 001454 000022
03800 401
03900 402 001456 005767 176756
04000 403 001462 001402
04100 404 001464 104002
04200 405 001466 000023
04300 406
04400 407 001470 122767 000004 176706
04500 408 001476 001402
04600 409 001500 104000
04700 410 001502 000024
04800
04900

```

```

.....
:TEST 4: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 077777,177777 + 177777,177777 = 000000,000000
: PS = 004, STACK POINTER = R2
.....

```

```

TST4: SCOPE
JSR R5, PUSH4 :PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
:SECOND OPERAND ON TOP
:WORD 177777,177777 :FIRST OPERAND ON BOTTOM
:WORD 077777,177777 :PROCESSOR PRIORITY LEVEL
:WORD 100 :FIS TRAP VECTOR
:WORD TRAPER,340 :SET UP STACK POINTER
MOV #STACK0,R2

NOP
FADD R2 :FLOATING ADD ON THE R2 STACK

JSR PC, POPR :POP THE ANSWER
MOV R2, $SP :SAVE 'STACK POINTER'
CMPB #004, $PSW :CHECK PS (EXCEPT T BIT)
BEQ .+6 :BRANCH IF OK
HLT :PS NOT EQUAL TO 004
20 :THE ERROR NUMBER IS 20

CMP #STACK4,$SP :CHECK THE STACK POINTER (R2)
BEQ .+6 :BRANCH IF OK
HLT :STACK POINTER (R2) NOT EQUAL TO #STACK4
21 :THE ERROR NUMBER IS 21

TST ANS1 :CHECK FIRST HALF OF ANSWER
BEQ .+6 :BRANCH IF OK
HLT+2 :ANS1 NOT EQUAL TO 000000
22 :THE ERROR NUMBER IS 22

TST ANS2 :CHECK SECOND HALF OF ANSWER
BEQ .+6 :BRANCH IF OK
HLT+2 :ANS2 NOT EQUAL TO 000000
23 :THE ERROR NUMBER IS 23

END4: CMPB #4, $TESTN :CHECK THE TEST NUMBER
BEQ .+6 :BRANCH IF OK
HLT :WRONG TEST! PC MUST HAVE FOULED UP.
24 :THE ERROR NUMBER IS 24

```

00100 413
00200 414
00300 415
00400 416
00500 417
00600 418
00700 419
00800 420
00900 421
01000 422
01100 423
01200 424
01300 425
01400 426
01500 427
01600 428
01700 429
01800 430
01900 431
02000 432
02100 433
02200 434
02300 435
02400 436
02500 437
02600 438
02700 439
02800 440
02900 441
03000 442
03100 443
03200 444
03300 445
03400 446
03500 447
03600 448
03700 449
03800 450
03900 451
04000 452
04100 453
04200 454
04300 455
04400 456
04500 457
04600 458

001504 104400
001506 004567 014322
001512 152525 052524
001516 052525 052525
001522 000217
001524 016456 000340
001530 000240
001532 075006
001534 004767 014334
001540 022706 000600
001544 001405
001546 012706 000600
001552 104000
001554 000025
001556 000421
001560 122767 000200 176644
001566 001402
001570 104000
001572 000026
001574 022767 044600 176634
001602 001402
001604 104002
001606 000027
001610 005767 176624
001614 001402
001616 104002
001620 000030
001622 122767 000005 176554
001630 001402
001632 104000
001634 000031

TEST 5: FADD (LSI-11 FLOATING ADD INSTRUCTION)
052525,052525 + 152525,052524 = 044600,000000
PS 200, STACK POINTER - SP

SCOPE
TST5: JSR R5 PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORI
.WORD 152525,052524 ;SECOND OPERAND ON TOP
.WORD 052525,052525 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR

NOP
FADD SP ;FLOATING ADD ON THE STACK

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA5 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER
HLT ;STACK POINTER FOULED UP
25 ;THE ERROR NUMBER IS 25
BR ENDS ;SKIP REST OF TEST

TSA5: CMPB #200, SPSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
26 ;THE ERROR NUMBER IS 26

CMP #044600,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 044600
27 ;THE ERROR NUMBER IS 27

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
30 ;THE ERROR NUMBER IS 30

ENDS: CMPB #5, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
31 ;THE ERROR NUMBER IS 31

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MA 30A(1052) 21-AUG-78 15:28 PAGE 13
CVKACC.P 16-AUG-78 08:41

459
460
461
462
463
464
465
466 001636 104400
467 001640 004567 014170
468 001644 025177 177777
469 001650 125200 000000
470 001654 000307
471 001656 016456 000340
472
473 001662 000240
474 001664 075006
475
476 001666 004767 014202
477 001672 022706 000600
478 001676 001405
479 001700 012706 000600
480 001704 104000
481 001706 000032
482 001710 000421
483
484 001712 122767 000210 176512
485 001720 001402
486 001722 104000
487 001724 000033
488
489 001726 022767 117200 176502
490 001734 001402
491 001736 104002
492 001740 000034
493
494 001742 005767 176472
495 001746 001402
496 001750 104002
497 001752 000035
498
499 001754 122767 000006 176422
500 001762 001402
501 001764 104000
502 001766 000036
503
504

FADD TEST SECTION

SEQ 0021

TEST 6: FADD (LSI-11 FLOATING ADD INSTRUCTION)
125200,000000 + 025177,177777 = 117200,000000
PS 210, STACK POINTER SP

SCOPE
TST6: JSR R5 PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORI
.WORD 025177,177777 ;SECOND OPERAND ON TOP
.WORD 125200,000000 ;FIRST OPERAND ON BOTTOM
.WORD 307 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR

NOP
FADD SP ;FLOATING ADD ON THE STACK

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA6 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER
HLT ;STACK POINTER FOULED UP
32 ;THE ERROR NUMBER IS 32
BR END6 ;SKIP REST OF TEST

TSA6: CMPB #210, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
33 ;THE ERROR NUMBER IS 33

CMP #117200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 117200
34 ;THE ERROR NUMBER IS 34

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
35 ;THE ERROR NUMBER IS 35

END6: CMPB #6, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
36 ;THE ERROR NUMBER IS 36

```

00100
00200
00300
00400 505
00500 506
00600 507
00700 508
00800 509
00900 510
01000 511
01100 512 001770 104400
01200 513 001772 004567 014210
01300 514 001776 100125 052525
01400 515 002002 135753 024642
01500 516 002006 000347
01600 517 002010 016456 000340
01700 518 002014 012705 000510
01800 519
01900 520 002020 000240
02000 521 002022 075005
02100 522
02200 523 002024 004767 014210
02300 524 002030 010567 176400
02400 525 002034 122767 000210 176370
02500 526 002042 001402
02600 527 002044 104000
02700 528 002046 000037
02800 529
02900 530 002050 022767 000514 176356
03000 531 002056 001402
03100 532 002060 104000
03200 533 002062 000040
03300 534
03400 535 002064 022767 135753 176344
03500 536 002072 001402
03600 537 002074 104002
03700 538 002076 000041
03800 539
03900 540 002100 022767 024642 176332
04000 541 002106 001402
04100 542 002110 104002
04200 543 002112 000042
04300 544
04400 545 002114 122767 000007 176262
04500 546 002122 001402
04600 547 002124 104000
04700 548 002126 000043
04800 549
04900 550

```

```

.....
TEST 7: FADD (LSI-11 FLOATING ADD INSTRUCTION)
135753,024642 + 100125,052525 = 135753,024642
PS = 210. STACK POINTER R5
.....

```

```

TEST7: SCOPE
JSR R5, PUSHR ; PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; WORD 100125,052525
; WORD 135753,024642 ; FIRST OPERAND ON BOTTOM
; WORD 347 ; PROCESSOR PRIORITY LEVEL
; WORD TRAPER,340 ; FIS TRAP VECTOR
MOV #STACK0,R5 ; SET UP STACK POINTER

NOP
FADD R5 ; FLOATING ADD ON THE R5 STACK

JSR PC, POPR ; POP THE ANSWER
MOV R5, SSP ; SAVE 'STACK POINTER'
CMPB #210, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 210
37 ; THE ERROR NUMBER IS 37

CMP #STACK4,SSP ; CHECK THE STACK POINTER (R5)
BEQ .+6 ; BRANCH IF OK
HLT ; STACK POINTER (R5) NOT EQUAL TO #STACK4
40 ; THE ERROR NUMBER IS 40

CMP #135753,ANS1 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 135753
41 ; THE ERROR NUMBER IS 41

CMP #024642,ANS2 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 024642
42 ; THE ERROR NUMBER IS 42

END7: CMPB #7, $*ESTN ; CHECK THE TEST NUMBER
BEQ .+6 ; BRANCH IF OK
HLT ; WRONG TEST. PC MUST HAVE FOULED UP.
43 ; THE ERROR NUMBER IS 43

```

```

00100
00200
00300
00400 551
00500 552
00600 553
00700 554
00800 555
00900 556
01000 557
01100 558 002130 104400
01200 559 002132 004567 014050
01300 560 002136 001357 024642
01400 561 002142 000052 125252
01500 562 002146 000257
01600 563 002150 016456 000340
01700 564 002154 012701 000510
01800 565
01900 566 002160 000240
02000 567 002162 075001
02100 568
02200 569 002164 004767 014050
02300 570 002170 010167 176240
02400 571 002174 122767 000200 176230
02500 572 002202 001402
02600 573 002204 104000
02700 574 002206 000044
02800 575
02900 576 002210 022767 000514 176216
03000 577 002216 001402
03100 578 002220 104000
03200 579 002222 000045
03300 580
03400 581 002224 022767 001357 176204
03500 582 002232 001402
03600 583 002234 104002
03700 584 002236 000046
03800 585
03900 586 002240 022767 024642 176172
04000 587 002246 001402
04100 588 002250 104002
04200 589 002252 000047
04300 590
04400 591 002254 122767 000010 176122
04500 592 002262 001402
04600 593 002264 104000
04700 594 002266 000050
04800 595
04900 596

```

```

:*****
:TEST 10: FADD (LSI-11 FLOATING ADD INSTRUCTION)
:          000052,125252 + 001357,024642 = 001357,024642
:          PS - 200,          STACK POINTER - R1
:*****

```

```

TST10: SCOPE
JSR    R5,    PUSHR    ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
.WORD  001357,024642  ;SECOND OPERAND ON TOP
.WORD  000052,125252  ;FIRST OPERAND ON BOTTOM
.WORD  257           ;PROCESSOR PRIORITY LEVEL
.WORD  TRAPER,340    ;FIS TRAP VECTOR
MOV    #STACK0,R1    ;SET UP STACK POINTER

NOP
FADD   R1            ;FLOATING ADD ON THE R1 STACK

JSR    PC,    POPR    ;POP THE ANSWER
MOV    R1,    $SP    ;SAVE 'STACK POINTER'
CMPB   #200,   $PSW   ;CHECK PS (EXCEPT T BIT)
BEQ    .+6        ;BRANCH IF OK
HLT    44        ;PS NOT EQUAL TO 200
                     ;THE ERROR NUMBER IS 44

CMP    #STACK4,$SP  ;CHECK THE STACK POINTER (R1)
BEQ    .+6        ;BRANCH IF OK
HLT    45        ;STACK POINTER (R1) NOT EQUAL TO #STACK4
                     ;THE ERROR NUMBER IS 45

CMP    #001357,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ    .+6        ;BRANCH IF OK
HLT+2 46        ;ANS1 NOT EQUAL TO 001357
                     ;THE ERROR NUMBER IS 46

CMP    #024642,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ    .+6        ;BRANCH IF OK
HLT+2 47        ;ANS2 NOT EQUAL TO 024642
                     ;THE ERROR NUMBER IS 47

END10: CMPB   #10,   $TESTN ;CHECK THE TEST NUMBER
BEQ    .+6        ;BRANCH IF OK
HLT    50        ;WRONG TEST PC MUST HAVE FOULED UP.
                     ;THE ERROR NUMBER IS 50

```



```

00100
00200
00300
00400 597
00500 598
00600 599
00700 600
00800 601
00900 602
01000 603
01100 604 002270 104400
01200 605 002272 004567 013710
01300 606 002276 000200 000000
01400 607 002302 100400 000000
01500 608 002306 000140
01600 609 002310 016456 000340
01700 610 002314 012705 000510
01800 611
01900 612 002320 000240
02000 613 002322 075005
02100 614
02200 615 002324 004767 013710
02300 616 002330 010567 176100
02400 617 002334 122767 000010 176070
02500 618 002342 001402
02600 619 002344 104000
02700 620 002346 000051
02800 621
02900 622 002350 022767 000514 176056
03000 623 002356 001402
03100 624 002360 104000
03200 625 002362 000052
03300 626
03400 627 002364 022767 100200 176044
03500 628 002372 001402
03600 629 002374 104002
03700 630 002376 000053
03800 631
03900 632 002400 005767 176034
04000 633 002404 001402
04100 634 002406 104002
04200 635 002410 000054
04300 636
04400 637 002412 122767 000011 175764
04500 638 002420 001402
04600 639 002422 104000
04700 640 002424 000055
04800 641
04900 642

```

```

*****
:TEST 11: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 100400,000000 + 000200,000000 = 100200,000000
: PS = 010, STACK POINTER R5
*****

```

```

TST11: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
.WORD 000200,000000 ;SECOND OPERAND ON TOP
.WORD 100400,000000 ;FIRST OPERAND ON BOTTOM
.WORD 140 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR
MOV #STACK0,R5 ;SET UP STACK POINTER

NOP
FADD R5 ;FLOATING ADD ON THE R5 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R5, $SP ;SAVE 'STACK POINTER'
CMPB #010, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 010
51 ;THE ERROR NUMBER IS 51

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R5)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R5) NOT EQUAL TO #STACK4
52 ;THE ERROR NUMBER IS 52

CMP #100200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 100200
53 ;THE ERROR NUMBER IS 53

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
54 ;THE ERROR NUMBER IS 54

END11: CMPB #11, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
55 ;THE ERROR NUMBER IS 55

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

(VKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 17
(VKACC.P'1 16-AUG-78 08:41

643
644
645
646
647
648
649
650 002426 104400
651 002430 004567 013552
652 002434 100252 125252
653 002440 000425 052525
654 002444 000217
655 002446 016456 000340
656 002452 012704 000510
657
658 002456 000240
659 002460 075004
660
661 002462 004767 013552
662 002466 010467 175742
663 002472 122767 000200 175732
664 002500 001402
665 002502 104000
666 002504 000056
667
668 002506 022767 000514 175720
669 002514 001402
670 002516 104000
671 002520 000057
672
673 002522 022767 000200 175706
674 002530 001402
675 002532 104002
676 002534 000060
677
678 002536 005767 175676
679 002542 001402
680 002544 104002
681 002546 000061
682
683 002550 122767 000012 175626
684 002556 001402
685 002560 104000
686 002562 000062
687
688

FADD TEST SECTION

SEQ 0025

:TEST 12: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 000425,052525 + 100252,125252 = 000200,000000
: PS - 200, STACK POINTER - R4

TST12: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
.WORD 100252,125252 ;SECOND OPERAND ON TOP
.WORD 000425,052525 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR
MOV #STACK0,R4 ;SET UP STACK POINTER

NOP
FADD R4 ;FLOATING ADD ON THE R4 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R4, \$SP ;SAVE 'STACK POINTER'
CMPB #200, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
56 ;THE ERROR NUMBER IS 56

CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R4)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R4) NOT EQUAL TO #STACK4
57 ;THE ERROR NUMBER IS 57

CMP #000200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 000200
60 ;THE ERROR NUMBER IS 60

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
61 ;THE ERROR NUMBER IS 61

END12: CMPB #12, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
62 ;THE ERROR NUMBER IS 62


```

00100
00200
00300
00400 735
00500 736
00600 737
00700 738
00800 739
00900 740
01000 741
01100 742 002716 104400
01200 743 002720 004567 013110
01300 744 002724 077452 125252
01400 745 002730 077652 125252
01500 746 002734 000257
01600 747 002736 016456 000340
01700 748
01800 749 002742 000240
01900 750 002744 075006
02000 751
02100 752 002746 004767 013122
02200 753 002752 022706 000600
02300 754 002756 001405
02400 755 002760 012706 000600
02500 756 002764 104000
02600 757 002766 000070
02700 758 002770 000422
02800 759
02900 760 002772 122767 000200 175432
03000 761 003000 001402
03100 762 003002 104000
03200 763 003004 000071
03300 764
03400 765 003006 022767 077777 175422
03500 766 003014 001402
03600 767 003016 104002
03700 768 003020 000072
03800 769
03900 770 003022 022767 177777 175410
04000 771 003030 001402
04100 772 003032 104002
04200 773 003034 000073
04300 774
04400 775 003036 122767 000014 175340
04500 776 003044 001402
04600 777 003046 104000
04700 778 003050 000074
04800 779
04900 780

```

```

*****
:TEST 14: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 077652,125252 + 077452,125252 = 077777,177777
: PS 200, STACK POINTER = SP
*****

```

```

SCOPE
TST14: JSR R5 PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORI
        .WORD 077452,125252 ;SECOND OPERAND ON TOP
        .WORD 077652,125252 ;FIRST OPERAND ON BOTTOM
        .WORD 257 ;PROCESSOR PRIORITY LEVEL
        .WORD TRAPER,340 ;FIS TRAP VECTOR

NOP
FADD SP ;FLOATING ADD ON THE STACK

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA14 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER
HLT ;STACK POINTER FOULED UP
70 ;THE ERROR NUMBER IS 70
BR END14 ;SKIP REST OF TEST

TSA14: CMPB #200, SPSW ;CHECK PS (EXCEPT T BIT)
        BEQ .+6 ;BRANCH IF OK
        HLT ;PS NOT EQUAL TO 200
        71 ;THE ERROR NUMBER IS 71

CMP #077777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 077777
72 ;THE ERROR NUMBER IS 72

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177777
73 ;THE ERROR NUMBER IS 73

END14: CMPB #14, $TESTN ;CHECK THE TEST NUMBER
        BEQ .+6 ;BRANCH IF OK
        HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
        74 ;THE ERROR NUMBER IS 74

```

```

00100
00200
00300
00400 781
00500 782
00600 783
00700 784
00800 785
00900 786
01000 787
01100 788 003052 104400
01200 789 003054 004567 013126
01300 790 003060 177652 125252
01400 791 003064 177452 125252
01500 792 003070 000357
01600 793 003072 016456 000340
01700 794 003076 012704 000510
01800 795
01900 796 003102 000240
02000 797 003104 075004
02100 798
02200 799 003106 004767 013126
02300 800 003112 010467 175316
02400 801 003116 122767 000210 175306
02500 802 003124 001402
02600 803 003126 104000
02700 804 003130 000075
02800 805
02900 806 003132 022767 000514 175274
03000 807 003140 001402
03100 808 003142 104000
03200 809 003144 000076
03300 810
03400 811 003146 022767 177777 175262
03500 812 003154 001402
03600 813 003156 104002
03700 814 003160 000077
03800 815
03900 816 003162 022767 177777 175250
04000 817 003170 001402
04100 818 003172 104002
04200 819 003174 000100
04300 820
04400 821 003176 122767 000015 175200
04500 822 003204 001402
04600 823 003206 104000
04700 824 003210 000101
04800 825
04900 826

```

```

:*****
:TEST 15: FADD (LSI-11 FLOATING ADD INSTRUCTION)
:          177452,125252 + 177652,125252 = 177777,177777
:          PS = 210,          STACK POINTER R4
:*****
TST15:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
        .WORD   177652,125252      ;SECOND OPERAND ON TOP
        .WORD   177452,125252      ;FIRST OPERAND ON BOTTOM
        .WORD   357                  ;PROCESSOR PRIORITY LEVEL
        .WORD   TRAPER,340          ;FIS TRAP VECTOR
        MOV     #STACK0,R4          ;SET UP STACK POINTER

        NOP
        FADD   R4                  ;FLOATING ADD ON THE R4 STACK

        ISR    PC,      POPR        ;POP THE ANSWER
        MOV    R4,      $SP         ;SAVE 'STACK POINTER'
        CMPB  #210,    $PSW        ;CHECK PS (EXCEPT T BIT)
        BEQ   .+6                ;BRANCH IF OK
        HLT   75                  ;PS NOT EQUAL TO 210
        ;THE ERROR NUMBER IS 75

        CMP   #STACK4,$SP          ;CHECK THE STACK POINTER (R4)
        BEQ   .+6                ;BRANCH IF OK
        HLT   76                  ;STACK POINTER (R4) NOT EQUAL TO #STACK4
        ;THE ERROR NUMBER IS 76

        CMP   #177777,ANS1         ;CHECK FIRST HALF OF ANSWER
        BEQ   .+6                ;BRANCH IF OK
        HLT+2 77                  ;ANS1 NOT EQUAL TO 177777
        ;THE ERROR NUMBER IS 77

        CMP   #177777,ANS2         ;CHECK SECOND HALF OF ANSWER
        BEQ   .+6                ;BRANCH IF OK
        HLT+2 100                 ;ANS2 NOT EQUAL TO 177777
        ;THE ERROR NUMBER IS 100

END15:  CMPB  #15,      $TESTN     ;CHECK THE TEST NUMBER
        BEQ   .+6                ;BRANCH IF OK
        HLT   101                 ;WRONG TEST. PC MUST HAVE FOULED UP.
        ;THE ERROR NUMBER IS 101

```



```

00100      883
00200      884
00300      885
00400      886
00500      887
00600      888
00700      889
00800      890
00900      891
01000      892
01100      893
01200      894
01300      895
01400      896
01500      897
01600      898
01700      899
01800      900
01900      901
02000      902
02100      903
02200      904
02300      905
02400      906
02500      907
02600      908
02700      909
02800      910
02900      911
03000      912
03100      913
03200      914
03300      915
03400      916
03500      917
03600      918
03700      919
03800      920
03900      921
04000      922
04100      923
04200      924
04300      925
04400      926
04500      927
04600      928
04700      929
04800      930
04900      931
05000      932
05100      933
05200      934
05300      935
05400      936
05500      937
05600      938
05700
05800
05900

      003364 104400
      003366 004567 012614
      003372 000377 177777
      003376 100200 000000
      003402 000157
      003404 003436 000000
      003410 012703 000510

      003414 000240
      003416 075003

      003420 004767 012614
      003424 010367 175004
      003430 104002
      003432 000110
      003434 000462

      003436 004767 012626
      003442 010367 174766
      003446 105767 174760
      003452 001402
      003454 104000
      003456 000111

      003460 022767 000510 174746
      003466 001402
      003470 104000
      003472 000112

      003474 022767 003420 174734
      003502 001402
      003504 104001
      003506 000113

      003510 022767 000012 174722
      003516 001402
      003520 104002
      003522 000114

      003524 022767 000377 174710
      003532 001402
      003534 104004
      003536 000115

      003540 022767 177777 174676
      003546 001402
      003550 104004
      003552 000116

      003554 022767 100200 174664
      003562 001402

```

```

:TEST 17:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
:              100200,000000 + 000377,177777 > UNDERFLOW
:              PS(ON STACK) 012,      STACK POINTER = R3
:*****
TST17:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
        .WORD   000377,177777      ;SECOND OPERAND ON TOP
        .WORD   100200,000000      ;FIRST OPERAND ON BOTTOM
        .WORD   157                ;PROCESSOR PRIORITY LEVEL
        .WORD   ISR17, 000         ;FIS TRAP VECTOR
        MOV     #STACK0,R3        ;SET UP R3 AS STACK POINTER

        NOP
        FADD    R3                ;FLOATING ADD ON THE R3 STACK

RTA17:  JSR      PC,      POPR       ;POP THE 'ANSWER'
        MOV     R3,      $SP       ;SAVE STACK POINTER (R3)
        HLT+2   110              ;FIS TRAP DIDN'T OCCURE!
        BR      END17            ;THE ERROR NUMBER IS 110

ISR17:  JSR      PC,      POPER     ;POP ALL DATA OFF THE STACKS
        MOV     R3,      $SP       ;SAVE STACK POINTER (R3)
        TSTB   $PSW              ;CHECK PS AFTER FIS TRAP
        BEQ    .+6               ;BRANCH IF OK
        HLT    111              ;PS AFTER FIS TRAP NOT EQUAL TO 000
        HLT    111              ;THE ERROR NUMBER IS 111

        CMP     #STACK0,$SP       ;CHECK THE STACK POINTER (R3)
        BEQ    .+6               ;BRANCH IF OK
        HLT    112              ;STACK POINTER (R3) NOT EQUAL TO #STACK0
        HLT    112              ;THE ERROR NUMBER IS 112

        CMP     #RTA17,ANS1      ;CHECK FIS TRAP RETURN ADDRESS
        BEQ    .+6               ;BRANCH IF OK
        HLT+1  113              ;FIS TRAP AT WRONG ADDRESS
        HLT    113              ;THE ERROR NUMBER IS 113

        CMP     #012,ANS2        ;CHECK PS BEFORE FIS TRAP
        BEQ    .+6               ;BRANCH IF OK
        HLT+2  114              ;PS AT FIS TRAP TIME NOT 012
        HLT    114              ;THE ERROR NUMBER IS 114

        CMP     #000377,ANS3     ;CHECK DATA FROM THE STACK
        BEQ    .+6               ;BRANCH IF OK
        HLT+4  115              ;DATA ON STACK (000377) CHANGED
        HLT    115              ;THE ERROR NUMBER IS 115

        CMP     #177777,ANS4     ;CHECK DATA FROM STACK
        BEQ    .+6               ;BRANCH IF OK
        HLT+4  116              ;DATA ON STACK (177777) CHANGED
        HLT    116              ;THE ERROR NUMBER IS 116

        CMP     #100200,ANS5     ;CHECK DATA FROM STACK
        BEQ    .+6               ;BRANCH IF OK

```

```

00100 CVKACC MARV11 30A(1052) 21-AUG-78 15:28 PAGE 23
00200 CVKACC.P11 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW SEQ 0031
00300
00400 939 003564 104006 HLT+6 ;DATA ON STACK (100200) CHANGED
00500 940 003566 000117 117 ;THE ERROR NUMBER IS 117
00600 941
00700 942 003570 005767 174654 TST ANS6 ;CHECK DATA FROM STACK
00800 943 003574 001402 BEQ .+6 ;BRANCH IF OK
00900 944 003576 104006 HLT+6 ;DATA ON STACK (000000) CHANGED
01000 945 003600 000120 120 ;THE ERROR NUMBER IS 120
01100 946
01200 947 003602 122767 000017 174574 END17: CMPB #17, $TESTN ;CHECK THE TEST NUMBER
01300 948 003610 001402 BEQ .+6 ;BRANCH IF OK
01400 949 003612 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
01500 950 003614 000121 121 ;THE ERROR NUMBER IS 121
01600 951
01700 952

```


00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900
05000
05100
05200
05300
05400
05500
05600
05700
05800
05900

CVKACC MARV11 30A(1052) 21-AUG-78 15:28 PAGE 24
CVKACC.P11 16-AUG-78 08:41

953
954
955
956
957
958
959
960 003616 104400
961 003620 004567 012210
962 003624 100377 177777
963 003630 000200 000000
964 003634 000257
965 003636 003664 000340
966
967 003642 000240
968 003644 075006
969
970 003646 004767 012222
971 003652 104002
972 003654 000122
973 003656 012706 000600
974 003662 000463
975
976 003664 004767 012236
977 003670 022706 000600
978 003674 001405
979 003676 012706 000600
980 003702 104000
981 003704 000123
982 003706 000451
983
984 003710 122767 000340 174514
985 003716 001402
986 003720 104000
987 003722 000124
988
989 003724 022767 003646 174504
990 003732 001402
991 003734 104001
992 003736 000125
993
994 003740 022767 000212 174472
995 003746 001402
996 003750 104002
997 003752 000126
998
999 003754 022767 100377 174460
1000 003762 001402
1001 003764 104004
1002 003766 000127
1003
1004 003770 022767 177777 174440
1005 003776 001402
1006 004000 104004
1007 004002 000130
1008

TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW

TEST 20: FADD (LSI-11 FLOATING ADD INSTRUCTION)
000200,000000 + 100377,177777 ==> UNDERFLOW
PS(ON STACK) = 212, STACK POINTER = SP

TST20: SCOPE
JSR R5, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 100377,177777 ;SECOND OPERAND ON TOP
.WORD 000200,000000 ;FIRST OPERAND ON BOTTOM
.WORD 257 ;PROCESSOR PRIORITY LEVEL
.WORD ISR20, 340 ;FIS TRAP VECTOR

NOP
FADD SP ;FLOATING ADD ON THE STACK

RTA20: JSR PC, POPS ;POP THE 'ANSWER'
HLT+2 ;FIS TRAP DIDN'T OCCURE!
122 ;THE ERROR NUMBER IS 122
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
BR END20

ISR20: JSR PC, POPES ;POP ALL DATA OFF THE STACK
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ ISA20 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
HLT ;STACK POINTER FOULED UP
123 ;THE ERROR NUMBER IS 123
BR END20 ;SKIP REST OF TEST

ISA20: CMPB #340, \$PSW ;CHECK PS AFTER FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 340
124 ;THE ERROR NUMBER IS 124

CMP #RTA20, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
125 ;THE ERROR NUMBER IS 125

CMP #212, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 212
126 ;THE ERROR NUMBER IS 126

CMP #100377, ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (100377) CHANGED
127 ;THE ERROR NUMBER IS 127

CMP #177777, ANS4 ;CHECK DATA FROM STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (177777) CHANGED
130 ;THE ERROR NUMBER IS 130

```

00100 CVKACC MAR 11 30A(1052) 21-AUG-78 15:28 PAGE 25
00200 CVKACC.P11 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW SEQ 0033
00300
00400 1009 004004 022767 000200 174434 CMP #000200,ANS5 ;CHECK DATA FROM STACK
00500 1010 004012 001402 BEQ .+6 ;BRANCH IF OK
00600 1011 004014 104006 HLT+6 ;DATA ON STACK (000200) CHANGED
00700 1012 004016 000131 131 ;THE ERROR NUMBER IS 131
00800 1013
00900 1014 004020 005767 174424 TST ANS6 ;CHECK DATA FROM STACK
01000 1015 004024 001402 BEQ .+6 ;BRANCH IF OK
01100 1016 004026 104006 HLT+6 ;DATA ON STACK (000000) CHANGED
01200 1017 004030 000132 132 ;THE ERROR NUMBER IS 132
01300 1018
01400 1019 004032 122767 000020 174344 END20: CMPB #20, $TESTN ;CHECK THE TEST NUMBER
01500 1020 004040 001402 BEQ .+6 ;BRANCH IF OK
01600 1021 004042 104006 HLT ;WRONG TEST PC MUST HAVE FOULED JP.
01700 1022 004044 000133 133 ;THE ERROR NUMBER IS 133
01800 1023
01900 1024

```

```

*****
TEST 21:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
              177452,125253 + 177652,125252 --> OVERFLOW
              PS(ON STACK) 002,      STACK POINTER R1
*****

```

```

TST21:  SCOPE
        JSR   R5,   PUSH4   ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
        .WORD 177652,125252 ;SECOND OPERAND ON TOP
        .WORD 177452,125253 ;FIRST OPERAND ON BOTTOM
        .WORD 105           ;PROCESSOR PRIORITY LEVEL
        .WORD ISR21, 252    ;FIS TRAP VECTOR
        MOV   #STACK0,R1   ;SET UP R1 AS STACK POINTER

        NOP
        FADD  R1           ;FLOATING ADD ON THE R1 STACK

RTA21:  JSR   PC,   POPR    ;POP THE 'ANSWER'
        MOV   R1,   $SP    ;SAVE STACK POINTER (R1)
        HLT+2 ;FIS TRAP DIDN'T OCCURE!
        134 ;THE ERROR NUMBER IS 134
        BR   END21

ISR21:  JSR   PC,   POPER   ;POP ALL DATA OFF THE STACKS
        MOV   R1,   $SP    ;SAVE STACK POINTER (R1)
        CMPB #252, $PSW    ;CHECK PS AFTER FIS TRAP
        BEQ  .+6           ;BRANCH IF OK
        HLT  ;PS AFTER FIS TRAP NOT EQUAL TO 252
        135 ;THE ERROR NUMBER IS 135

        CMP  #STACK0,$SP  ;CHECK THE STACK POINTER (R1)
        BEQ  .+6           ;BRANCH IF OK
        HLT  ;STACK POINTER (R1) NOT EQUAL TO #STACK0
        136 ;THE ERROR NUMBER IS 136

        CMP  #RTA21,ANS1  ;CHECK FIS TRAP RETURN ADDRESS
        BEQ  .+6           ;BRANCH IF OK
        HLT+1 ;FIS TRAP AT WRONG ADDRESS
        137 ;THE ERROR NUMBER IS 137

        CMP  #002,ANS2    ;CHECK PS BEFORE FIS TRAP
        BEQ  .+6           ;BRANCH IF OK
        HLT+2 ;PS AT FIS TRAP TIME NOT 002
        140 ;THE ERROR NUMBER IS 140

        CMP  #177652,ANS3 ;CHECK DATA FROM THE STACK
        BEQ  .+6           ;BRANCH IF OK
        HLT+4 ;DATA ON STACK (177652) CHANGED
        141 ;THE ERROR NUMBER IS 141

        CMP  #125252,ANS4 ;CHECK DATA FROM STACK
        BEQ  .+6           ;BRANCH IF OK
        HLT+4 ;DATA ON STACK (125252) CHANGED
        142 ;THE ERROR NUMBER IS 142

```

```

00300 1025
00400 1026
00500 1027
00600 1028
00700 1029
00800 1030
00900 1031
01000 1032
01100 1033 004046 104400
01200 1034 004050 004567 012132
01300 1035 004054 177652 125252
01400 1036 004060 177452 125253
01500 1037 004064 000105
01600 1038 004066 004120 000252
01700 1039 004072 012701 000510
01800 1040
01900 1041 004076 000240
02000 1042 004100 075001
02100 1043
02200 1044 004102 004767 012132
02300 1045 004106 010167 174322
02400 1046 004112 104002
02500 1047 004114 000134
02600 1048 004116 000464
02700 1049
02800 1050 004120 004767 012144
02900 1051 004124 010167 174304
03000 1052 004130 122767 000252 174274
03100 1053 004136 001402
03200 1054 004140 104000
03300 1055 004142 000135
03400 1056
03500 1057 004144 022767 000510 174262
03600 1058 004152 001402
03700 1059 004154 104000
03800 1060 004156 000136
03900 1061
04000 1062 004160 022767 004102 174250
04100 1063 004166 001402
04200 1064 004170 104001
04300 1065 004172 000137
04400 1066
04500 1067 004174 022767 000002 174236
04600 1068 004202 001402
04700 1069 004204 104002
04800 1070 004206 000140
04900 1071
05000 1072 004210 022767 177652 174224
05100 1073 004216 001402
05200 1074 004220 104004
05300 1075 004222 000141
05400 1076
05500 1077 004224 022767 125252 174212
05600 1078 004232 001402
05700 1079 004234 104004
05800 1080 004236 000142

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900

(VKACC MACV11 30A(1052) 21-AUG-78 15:28 PAGE 27
(VKACC.P11 16-AUG-78 08:41
1081 004240 022767 177452 174200
1082 004246 001402
1083 004250 104006
1084 004252 000143
1085
1086 004254 022767 125253 174166
1087 004262 001402
1088 004264 104006
1089 004266 000144
1090
1091 004270 122767 000021 174106
1092 004276 001402
1093 004300 104000
1094 004302 000145
1095
1096

TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0035

CMP #177452,ANS5 :CHECK DATA FROM STACK
BEQ .+6 :BRANCH IF OK
HLT+6 :DATA ON STACK (177452) CHANGED
143 :THE ERROR NUMBER IS 143

CMP #125253,ANS6 :CHECK DATA FROM STACK
BEQ .+6 :BRANCH IF OK
HLT+6 :DATA ON STACK (125253) CHANGED
144 :THE ERROR NUMBER IS 144

END21: CMPB #21, \$TESTN :CHECK THE TEST NUMBER
BEQ .+6 :BRANCH IF OK
HLT :WRONG TEST! PC MUST HAVE FOULED UP.
145 :THE ERROR NUMBER IS 145

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900
05000
05100
05200
05300
05400
05500
05600
05700
05800
05900

(VKACC MA'V11 30A(1052) 21-AUG-78 15:28 PAGE 28
(VKACC.P'1 16-AUG-78 08:41
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152

004304 104400
004306 004567 011522
004312 077452 125252
004316 077652 125253
004322 000003
004324 004352 000344
004330 000240
004332 075006
004334 004767 011534
004340 104002
004342 000146
004344 012706 000600
004350 000464
004352 004767 011550
004356 022706 000600
004362 001405
004364 012706 000600
004370 104000
004372 000147
004374 000452
004376 122767 000344 174026
004404 001402
004406 104000
004410 000150
004412 022767 004334 174026
004420 001402
004422 104001
004424 000151
004426 022767 000002 174004
004434 001402
004436 104002
004440 000152
004442 022767 077452 173772
004450 001402
004452 104004
004454 000153
004456 022767 125252 173760
004464 001402
004466 104004
004470 000154

TEST 22: FADD (LSI-11 FLOATING ADD INSTRUCTION)
077452,125253 + 077452,125252 ==> OVERFLOW
PS(ON STACK) = 002, STACK POINTER SP

TST22: JSR R5, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 077452,125252 ;SECOND OPERAND ON TOP
.WORD 077652,125253 ;FIRST OPERAND ON BOTTOM
.WORD 003 ;PROCESSOR PRIORITY LEVEL
.WORD ISR22, 344 ;FIS TRAP VECTOR

NOP
FADD SP ;FLOATING ADD ON THE STACK

RTA22: JSR PC, POPS ;POP THE 'ANSWER'
HLT+2 ;FIS TRAP DIDN'T OCCURE!
146 ;THE ERROR NUMBER IS 146
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
BR END22

ISR22: JSR PC, POPES ;POP ALL DATA OFF THE STACK
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ ISA22 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
HLT ;STACK POINTER FOULED UP
147 ;THE ERROR NUMBER IS 147
BR END22 ;SKIP REST OF TEST

ISA22: CMPB #344, \$PSW ;CHECK PS AFTER FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 344
150 ;THE ERROR NUMBER IS 150

CMP #RTA22, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
151 ;THE ERROR NUMBER IS 151

CMP #002, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 002
152 ;THE ERROR NUMBER IS 152

CMP #077452, ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (077452) CHANGED
153 ;THE ERROR NUMBER IS 153

CMP #125252, ANS4 ;CHECK DATA FROM STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (125252) CHANGED
154 ;THE ERROR NUMBER IS 154

```

00100 (VKACC MACY 1 30A(1052) 21-AUG-78 15:28 PAGE 29
00200 (VKACC.P 1 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH OVERFLOW SEQ 0037
00300
00400 1153 004472 022767 077652 173746 CMP #077652,ANS5 ;CHECK DATA FROM STACK
00500 1154 004500 001402 BEQ .+6 ;BRANCH IF OK
00600 1155 004502 104006 HLT+6 ;DATA ON STACK (077652) CHANGED
00700 1156 004504 000155 155 ;THE ERROR NUMBER IS 155
00800 1157
00900 1158 004506 022767 125253 173734 CMP #125253,ANS6 ;CHECK DATA FROM STACK
01000 1159 004514 001402 BEQ .+6 ;BRANCH IF OK
01100 1160 004516 104006 HLT+6 ;DATA ON STACK (125253) CHANGED
01200 1161 004520 000156 156 ;THE ERROR NUMBER IS 156
01300 1162
01400 1163 004522 122767 000022 173654 END22: CMPB #22, $TESTN ;CHECK THE TEST NUMBER
01500 1164 004530 001402 BEQ .+6 ;BRANCH IF OK
01600 1165 004532 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
01700 1166 004534 000157 157 ;THE ERROR NUMBER IS 157
01800 1167
01900 1168

```

00300
00400 1169
00500 1170
00600 1171
00700 1172
00800 1173
00900 1174
01000 1175
01100 1176 004536 104400
01200 1177 004540 004567 011442
01300 1178 004544 135352 051107
01400 1179 004550 177520 017552
01500 1180 004554 000040
01600 1181 004556 016456 000340
01700 1182 004562 012701 000510
01800 1183
01900 1184 004566 000240
02000 1185 004570 075011
02100 1186
02200 1187 004572 004767 011442
02300 1188 004576 010167 173632
02400 1189 004602 122767 000010 173622
02500 1190 004610 001402
02600 1191 004612 104000
02700 1192 004614 000160
02800 1193
02900 1194 004616 022767 000514 173610
03000 1195 004624 001402
03100 1196 004626 104000
03200 1197 004630 000161
03300 1198
03400 1199 004632 022767 177520 173576
03500 1200 004640 001402
03600 1201 004642 104002
03700 1202 004644 000162
03800 1203
03900 1204 004646 022767 017552 173564
04000 1205 004654 001402
04100 1206 004656 104002
04200 1207 004660 000163
04300 1208
04400 1209 004662 122767 000023 173514
04500 1210 004670 001402
04600 1211 004672 104000
04700 1212 004674 000164
04800 1213
04900 1214

```

:*****
:TEST 23: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
:          177520,017552 - 135352,051107 = 177520,017552
:          PS = 010, STACK POINTER = R1
:*****

TST23: SCOPE
        JSR R5, PUSHR ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
        .WORD 135352,051107 ;SECOND OPERAND ON TOP
        .WORD 177520,017552 ;FIRST OPERAND ON BOTTOM
        .WORD 040 ;PROCESSOR PRIORITY LEVEL
        .WORD TRAPER, 340 ;FIS TRAP VECTOR
        MOV #STACK0,R1 ;SET UP STACK POINTER

        NOP
        FSUB R1 ;FLOATING SUBTRACT ON THE R1 STACK

        JSR PC, POPR ;POP THE ANSWER
        MOV R1, $SP ;SAVE 'STACK POINTER'
        CMPB #010, $PSW ;CHECK PS (EXCEPT T BIT)
        BEQ .+6 ;BRANCH IF OK
        HLT ;PS NOT EQUAL TO 010
        160 ;THE ERROR NUMBER IS 160

        CMP #STACK4,$SP ;CHECK THE STACK POINTER (R1)
        BEQ .+6 ;BRANCH IF OK
        HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK4
        161 ;THE ERROR NUMBER IS 161

        CMP #177520,ANS1 ;CHECK FIRST HALF OF ANSWER
        BEQ .+6 ;BRANCH IF OK
        HLT+2 ;ANS1 NOT EQUAL TO 177520
        162 ;THE ERROR NUMBER IS 162

        CMP #017552,ANS2 ;CHECK SECOND HALF OF ANSWER
        BEQ .+6 ;BRANCH IF OK
        HLT+2 ;ANS2 NOT EQUAL TO 017552
        163 ;THE ERROR NUMBER IS 163

END23: CMPB #23, $TESTN ;CHECK THE TEST NUMBER
        BEQ .+6 ;BRANCH IF OK
        HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
        164 ;THE ERROR NUMBER IS 164

```

```

00300
00400 1215
00500 1216
00600 1217
00700 1218
00800 1219
00900 1220
01000 1221
01100 1222 004676 104400
01200 1223 004700 004567 011302
01300 1224 004704 125252 125253
01400 1225 004710 125252 125252
01500 1226 004714 000047
01600 1227 004716 016456 000340
01700 1228 004722 012700 000510
01800 1229
01900 1230 004726 000240
02000 1231 004730 075010
02100 1232
02200 1233 004732 004767 011302
02300 1234 004736 010067 173472
02400 1235 004742 105767 173464
02500 1236 004746 001402
02600 1237 004750 104000
02700 1238 004752 000165
02800 1239
02900 1240 004754 022767 000514 173452
03000 1241 004762 001402
03100 1242 004764 104000
03200 1243 004766 000166
03300 1244
03400 1245 004770 022767 017400 173440
03500 1246 004776 001402
03600 1247 005000 104002
03700 1248 005002 000167
03800 1249
03900 1250 005004 005767 173430
04000 1251 005010 001402
04100 1252 005012 104002
04200 1253 005014 000170
04300 1254
04400 1255 005016 122767 000024 173360
04500 1256 005024 001402
04600 1257 005026 104000
04700 1258 005030 000171
04800 1259
04900 1260

```

```

*****
:TEST 24: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 125252,125252 - 125252,125253 = 017400,000000
: PS = 000, STACK POINTER R0
*****

```

```

TST24: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
:SECOND OPERAND ON TOP
:WORD 125252,125253 ;FIRST OPERAND ON BOTTOM
:WORD 125252,125252 ;PROCESSOR PRIORITY LEVEL
:WORD 047 ;FIS TRAP VECTOR
:WORD TRAPER, 340 ;SET UP STACK POINTER
MOV #STACK0,R0

NOP
FSUB R0 ;FLOATING SUBTRACT ON THE R0 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R0, $SP ;SAVE 'STACK POINTER'
TSTB $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
165 ;THE ERROR NUMBER IS 165

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R0)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
166 ;THE ERROR NUMBER IS 166

CMP #017400,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 017400
167 ;THE ERROR NUMBER IS 167

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
170 ;THE ERROR NUMBER IS 170

END24: CMPB #24, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
171 ;THE ERROR NUMBER IS 171

```



```

00100
00200
00300
00400 1261
00500 1262
00600 1263
00700 1264
00800 1265
00900 1266
01000 1267
01100 1268 005032 104400
01200 1269 005034 004567 010774
01300 1270 005040 100177 177777
01400 1271 005044 002460 123456
01500 1272 005050 000015
01600 1273 005052 016456 000340
01700 1274
01800 1275 005056 000240
01900 1276 005060 075016
02000 1277
02100 1278 005062 004767 011006
02200 1279 005066 022706 000600
02300 1280 005072 001405
02400 1281 005074 012706 000600
02500 1282
02600 1283
02700 1284
02800 1285
02900 1286
03000 1287
03100 1288 005100 104000
03200 1289 005102 000172
03300 1290 005104 000422
03400 1291
03500 1292 005106 122767 000000 173316
03600 1293 005114 001402
03700 1294 005116 104000
03800 1295 005120 000173
03900 1296
04000 1297 005122 022767 002460 173306
04100 1298 005130 001402
04200 1299 005132 104002
04300 1300 005134 000174
04400 1301
04500 1302 005136 022767 123456 173274
04600 1303 005144 001402
04700 1304 005146 104002
04800 1305 005150 000175
04900 1306
05000 1307 005152 122767 000025 173224
05100 1308 005160 001402
05200 1309 005162 104000
05300 1310 005164 000176
05400 1311
05500 1312

```

```

*****
:TEST 25: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 002460,123456 - 100177,177777 - 002460,123456
: PS 000, STACK POINTER SP
*****

```

```

TST25: SCOPE
JSR R5, PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 100177,177777 ;SECOND OPERAND ON TOP
.WORD 002460,123456 ;FIRST OPERAND ON BOTTOM
.WORD 015 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR

```

```

NOP
FSUB SP ;FLOATING SUBTRACT ON THE STACK

```

```

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA25 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER

```

```

HLT ;STACK POINTER FOULED UP
172 ;THE ERROR NUMBER IS 172
BR END25 ;SKIP REST OF TEST

```

```

TSA25: CMPB #000, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
173 ;THE ERROR NUMBER IS 173

```

```

CMP #002460,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 002460
174 ;THE ERROR NUMBER IS 174

```

```

CMP #123456,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 123456
175 ;THE ERROR NUMBER IS 175

```

```

END25: CMPB #25, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
176 ;THE ERROR NUMBER IS 176

```

00300 1313
00400 1314
00500 1315
00600 1316
00700 1317
00800 1318
00900 1319
01000 1320 005166 104400
01100 1321 005170 004567 011012
01200 1322 005174 000252 125252
01300 1323 005200 000425 052525
01400 1324 005204 000217
01500 1325 005206 016456 000340
01600 1326 005212 012704 000510
01700 1327
01800 1328 005216 000240
01900 1329 005220 075014
02000 1330
02100 1331 005222 004767 011012
02200 1332 005226 010467 173202
02300 1333 005232 122767 000200 173172
02400 1334 005240 001402
02500 1335 005242 104000
02600 1336 005244 000177
02700 1337
02800 1338 005246 022767 000514 173160
02900 1339 005254 001402
03000 1340 005256 104000
03100 1341 005260 000200
03200 1342
03300 1343 005262 022767 000200 173146
03400 1344 005270 001402
03500 1345 005272 104002
03600 1346 005274 000201
03700 1347
03800 1348 005276 005767 173136
03900 1349 005302 001402
04000 1350 005304 104002
04100 1351 005306 000202
04200 1352
04300 1353 005310 122767 000026 173066
04400 1354 005316 001402
04500 1355 005320 104000
04600 1356 005322 000203
04700 1357
04800 1358
04900

```

:*****
:TEST 26: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 000425,052525 - 000252,125252 = 000200,000000
: PS - 200, STACK POINTER = R4
:*****

TST26: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
.WORD 000252,125252 ;SECOND OPERAND ON TOP
.WORD 000425,052525 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R4 ;SET UP STACK POINTER

NOP
FSUB R4 ;FLOATING SUBTRACT ON THE R4 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R4, $SP ;SAVE 'STACK POINTER'
CMPB #200, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
177 ;THE ERROR NUMBER IS 177

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R4)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R4) NOT EQUAL TO #STACK4
200 ;THE ERROR NUMBER IS 200

CMP #000200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 000200
201 ;THE ERROR NUMBER IS 201

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
202 ;THE ERROR NUMBER IS 202

END26: CMPB #26, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
203 ;THE ERROR NUMBER IS 203

```


00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 35
CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

1411
1412
1413
1414
1415
1416
1417
1418 005460 104400
1419 005462 004567 010520
1420 005466 043125 052525
1421 005472 035152 125252
1422 005476 000147
1423 005500 016456 000340
1424 005504 012703 000510
1425
1426 005510 000240
1427 005512 075013
1428
1429 005514 004767 010520
1430 005520 010367 172710
1431 005524 122767 000010 172700
1432 005532 001402
1433 005534 104000
1434 005536 000211
1435
1436 005540 022767 000514 172666
1437 005546 001402
1438 005550 104000
1439 005552 000212
1440
1441 005554 022767 143125 172654
1442 005562 001402
1443 005564 104002
1444 005566 000213
1445
1446 005570 022767 052524 172642
1447 005576 001402
1448 005600 104002
1449 005602 000214
1450
1451 005604 122767 000030 172572
1452 005612 001402
1453 005614 104000
1454 005616 000215
1455
1456

```
*****
:TEST 30:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
:              035152,125252 - 043125,052525 = 143125,052524
:              PS - 010,      STACK POINTER = R3
*****

TST30:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
        .WORD   043125,052525      ;SECOND OPERAND ON TOP
        .WORD   035152,125252      ;FIRST OPERAND ON BOTTOM
        .WORD   147                ;PROCESSOR PRIORITY LEVEL
        .WORD   TRAPER, 340        ;FIS TRAP VECTOR
        MOV     #STACK0,R3        ;SET UP STACK POINTER

        NGP
        FSUB   R3                ;FLOATING SUBTRACT ON THE R3 STACK

        JSR   PC,      POPR      ;POP THE ANSWER
        MOV   R3,      $SP      ;SAVE 'STACK POINTER'
        CMPB #010,    $PSW      ;CHECK PS (EXCEPT T BIT)
        BEQ  .+6            ;BRANCH IF OK
        HLT  211          ;PS NOT EQUAL TO 010
                          ;THE ERROR NUMBER IS 211

        CMP   #STACK4,$SP      ;CHECK THE STACK POINTER (R3)
        BEQ  .+6            ;BRANCH IF OK
        HLT  212          ;STACK POINTER (R3) NOT EQUAL TO #STACK4
                          ;THE ERROR NUMBER IS 212

        CMP   #143125,ANS1     ;CHECK FIRST HALF OF ANSWER
        BEQ  .+6            ;BRANCH IF OK
        HLT+2 213          ;ANS1 NOT EQUAL TO 143125
                          ;THE ERROR NUMBER IS 213

        CMP   #052524,ANS2     ;CHECK SECOND HALF OF ANSWER
        BEQ  .+6            ;BRANCH IF OK
        HLT+2 214          ;ANS2 NOT EQUAL TO 052524
                          ;THE ERROR NUMBER IS 214

END30:  CMPB  #30,      $TESTN   ;CHECK THE TEST NUMBER
        BEQ  .+6            ;BRANCH IF OK
        HLT  215          ;WRONG TEST. PC MUST HAVE FOULED UP.
                          ;THE ERROR NUMBER IS 215
```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MARY11 30A(1052) 21-AUG-78 15:28 PAGE 36
CVKACC.P11 16-AUG-78 08:41

1457
1458
1459
1460
1461
1462
1463
1464 005620 104400
1465 005622 004567 010360
1466 005626 135152 125252
1467 005632 143325 052525
1468 005636 000243
1469 005640 016456 000340
1470 005644 012700 000510
1471
1472 005650 000240
1473 005652 075010
1474
1475 005654 004767 010360
1476 005660 010067 172550
1477 005664 122767 000210 172540
1478 005672 001402
1479 005674 104000
1480 005676 000216
1481
1482 005700 022767 000514 172526
1483 005706 001402
1484 005710 104000
1485 005712 000217
1486
1487 005714 022767 143325 172514
1488 005722 001402
1489 005724 104002
1490 005726 000220
1491
1492 005730 022767 052525 172502
1493 005736 001402
1494 005740 104002
1495 005742 000221
1496
1497 005744 122767 000031 172432
1498 005752 001402
1499 005754 104000
1500 005756 000222
1501
1502

FSUB TEST SECTION

F 4

SEQ 0044

:TEST 31: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 143325,052525 - 135152,125252 - 143325,052525
: PS = 210, STACK POINTER = R0
:*****

TST31: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
.WORD 135152,125252 ;SECOND OPERAND ON TOP
.WORD 143325,052525 ;FIRST OPERAND ON BOTTOM
.WORD 243 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R0 ;SET UP STACK POINTER

NOP
FSUB R0 ;FLOATING SUBTRACT ON THE R0 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R0, \$SP ;SAVE 'STACK POINTER'
CMPB #210, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
216 ;THE ERROR NUMBER IS 216

CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R0)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
217 ;THE ERROR NUMBER IS 217

CMP #143325,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 143325
220 ;THE ERROR NUMBER IS 220

CMP #052525,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 052525
221 ;THE ERROR NUMBER IS 221

END31: CMPB #31, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
222 ;THE ERROR NUMBER IS 222

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

LVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 37
CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

1503
1504
1505
1506
1507
1508
1509
1510 005760 104400
1511 005762 004567 010220
1512 005766 143325 052525
1513 005772 135152 125252
1514 005776 000357
1515 006000 016456 000340
1516 006004 012705 000510
1517
1518 006010 000240
1519 006012 075015
1520
1521 006014 004767 010220
1522 006020 010567 172410
1523 006024 122767 000200 172400
1524 006032 001402
1525 006034 104000
1526 006036 000223
1527
1528 006040 022767 000514 172366
1529 006046 001402
1530 006050 104000
1531 006052 000224
1532
1533 006054 022767 043325 172354
1534 006062 001402
1535 006064 104002
1536 006066 000225
1537
1538 006070 022767 052525 172342
1539 006076 001402
1540 006100 104002
1541 006102 000226
1542
1543 006104 122767 000032 172272
1544 006112 001402
1545 006114 104000
1546 006116 000227
1547
1548

:TEST 32: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 135152,125252 - 143325,052525 - 043325,052525
: PS = 200, STACK POINTER R5
:*****
TST32: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
.WORD 143325,052525 ;SECOND OPERAND ON TOP
.WORD 135152,125252 ;FIRST OPERAND ON BOTTOM
.WORD 357 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R5 ;SET UP STACK POINTER

NOP
FSUB R5 ;FLOATING SUBTRACT ON THE R5 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R5, \$SP ;SAVE 'STACK POINTER'
CMPB #200, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
223 ;THE ERROR NUMBER IS 223

CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R5)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R5) NOT EQUAL TO #STACK4
224 ;THE ERROR NUMBER IS 224

CMP #043325,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 043325
225 ;THE ERROR NUMBER IS 225

CMP #052525,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 052525
226 ;THE ERROR NUMBER IS 226

END32: CMPB #32, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST: PC MUST HAVE FOULED UP.
227 ;THE ERROR NUMBER IS 227

00300 1549
00400 1550
00500 1551
00600 1552
00700 1553
00800 1554
00900 1555
01000 1556
01100 1557 006120 104400
01200 1558 006122 004567 010060
01300 1559 006126 035152 125252
01400 1560 006132 043125 052525
01500 1561 006136 000040
01600 1562 006140 016456 000340
01700 1563 006144 012702 000510
01800 1564
01900 1565 006150 000240
02000 1566 006152 075012
02100 1567
02200 1568 006154 004767 010060
02300 1569 006160 010267 172250
02400 1570 006164 105767 172242
02500 1571 006170 001402
02600 1572 006172 104000
02700 1573 006174 000230
02800 1574
02900 1575 006176 022767 000514 172230
03000 1576 006204 001402
03100 1577 006206 104000
03200 1578 006210 000231
03300 1579
03400 1580 006212 022767 043125 172216
03500 1581 006220 001402
03600 1582 006222 104002
03700 1583 006224 000232
03800 1584
03900 1585 006226 022767 052524 172204
04000 1586 006234 001402
04100 1587 006236 104002
04200 1588 006240 000233
04300 1589
04400 1590 006242 122767 000033 172134
04500 1591 006250 001402
04600 1592 006252 104000
04700 1593 006254 000234
04800 1594
04900 1594

```

:*****
:TEST 33: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 043125,052525 - 035152,125252 = 043125,052524
: PS 000, STACK POINTER = R2
:*****

```

```

TST33: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
.WORD 035152,125252 ;SECOND OPERAND ON TOP
.WORD 043125,052525 ;FIRST OPERAND ON BOTTOM
.WORD 040 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R2 ;SET UP STACK POINTER

NOP
FSUB R2 ;FLOATING SUBTRACT ON THE R2 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R2, $SP ;SAVE 'STACK POINTER'
TSTB $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
230 ;THE ERROR NUMBER IS 230

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R2)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R2) NOT EQUAL TO #STACK4
231 ;THE ERROR NUMBER IS 231

CMP #043125,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 043125
232 ;THE ERROR NUMBER IS 232

CMP #052524,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 052524
233 ;THE ERROR NUMBER IS 233

END33: CMPB #33, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
234 ;THE ERROR NUMBER IS 234

```

```

00100
00200
00300
00400 1595
00500 1596
00600 1597
00700 1598
00800 1599
00900 1600
01000 1601
01100 1602 006256 104400
01200 1603 006260 004567 007722
01300 1604 006264 000000 000000
01400 1605 006270 000000 000000
01500 1606 006274 000217
01600 1607 006276 016456 000340
01700 1608 006302 012700 000510
01800 1609
01900 1610 006306 000240
02000 1611 006310 075010
02100 1612
02200 1613 006312 004767 007722
02300 1614 006316 010067 172112
02400 1615 006322 122767 000204 172102
02500 1616 006330 001402
02600 1617 006332 104000
02700 1618 006334 000235
02800 1619
02900 1620 006336 022767 000514 172070
03000 1621 006344 001402
03100 1622 006346 104000
03200 1623 006350 000236
03300 1624
03400 1625 006352 005767 172060
03500 1626 006356 001402
03600 1627 006360 104002
03700 1628 006362 000237
03800 1629
03900 1630 006364 005767 172050
04000 1631 006370 001402
04100 1632 006372 104002
04200 1633 006374 000240
04300 1634
04400 1635 006376 122767 000034 172000
04500 1636 006404 001402
04600 1637 006406 104000
04700 1638 006410 000241
04800 1639
04900 1640

```

```

*****
:TEST 34: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 000000,000000 - 000000,000000 = 000000,000000
: PS = 204, STACK POINTER = R0
*****

TST34: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
.WORD 000000,000000 ;SECOND OPERAND ON TOP
.WORD 000000,000000 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R0 ;SET UP STACK POINTER

NOP
FSUB R0 ;FLOATING SUBTRACT ON THE R0 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R0, $SP ;SAVE 'STACK POINTER'
CMPB #204, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 204
235 ;THE ERROR NUMBER IS 235

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R0)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
236 ;THE ERROR NUMBER IS 236

TST ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 000000
237 ;THE ERROR NUMBER IS 237

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
240 ;THE ERROR NUMBER IS 240

END34: CMPB #34, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
241 ;THE ERROR NUMBER IS 241

```



```

00100 1641
00200 1642
00300 1643
00400 1644
00500 1645
00600 1646
00700 1647
00800 1648 006412 104400
00900 1649 006414 004567 007566
01000 1650 006420 177777 177777
01100 1651 006424 000000 000000
01200 1652 006430 000100
01300 1653 006432 016456 000340
01400 1654 006436 012702 000510
01500 1655
01600 1656 006442 000240
01700 1657 006444 075012
01800 1658
01900 1659 006446 004767 007566
02000 1660 006452 010267 171756
02100 1661 006456 105767 171750
02200 1662 006462 001402
02300 1663 006464 104000
02400 1664 006466 000242
02500 1665
02600 1666 006470 022767 000514 171736
02700 1667 006476 001402
02800 1668 006500 104000
02900 1669 006502 000243
03000 1670
03100 1671 006504 022767 077777 171724
03200 1672 006512 001402
03300 1673 006514 104002
03400 1674 006516 000244
03500 1675
03600 1676 006520 022767 177777 171712
03700 1677 006526 001402
03800 1678 006530 104002
03900 1679 006532 000245
04000 1680
04100 1681 006534 122767 000035 171642
04200 1682 006542 001402
04300 1683 006544 104000
04400 1684 006546 000246
04500 1685
04600 1686
04700
04800
04900

```

```

*****
:TEST 35: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 000000,000000 - 177777,177777 = 077777,177777
: PS = 000, STACK POINTER = R2
*****

```

```

TST35: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
.WORD 177777,177777 ;SECOND OPERAND ON TOP
.WORD 000000,000000 ;FIRST OPERAND ON BOTTOM
.WORD 100 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R2 ;SET UP STACK POINTER

NOP
FSUB R2 ;FLOATING SUBTRACT ON THE R2 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R2, $SP ;SAVE 'STACK POINTER'
TSTB $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
242 ;THE ERROR NUMBER IS 242

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R2)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R2) NOT EQUAL TO #STACK4
243 ;THE ERROR NUMBER IS 243

CMP #077777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 077777
244 ;THE ERROR NUMBER IS 244

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177777
245 ;THE ERROR NUMBER IS 245

END35: CMPB #35, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
246 ;THE ERROR NUMBER IS 246

```

```

00300 1687
00400 1688
00500 1689
00600 1690
00700 1691
00800 1692
00900 1693
01000 1694
01100 1695 006550 104400
01200 1696 006552 004567 007256
01300 1697 006556 077777 177777
01400 1698 006562 000000 000000
01500 1699 006566 000217
01600 1700 006570 016456 000340
01700 1701
01800 1702 006574 000240
01900 1703 006576 075016
02000 1704
02100 1705 006600 004767 007270
02200 1706 006604 022706 000600
02300 1707 006610 001405
02400 1708 006612 012706 000600
02500 1709
02600 1710
02700 1711
02800 1712
02900 1713
03000 1714
03100 1715 006616 104000
03200 1716 006620 000247
03300 1717 006622 000422
03400 1718
03500 1719 006624 122767 000210 171600
03600 1720 006632 001402
03700 1721 006634 104000
03800 1722 006636 000250
03900 1723
04000 1724 006640 022767 177777 171570
04100 1725 006646 001402
04200 1726 006650 104002
04300 1727 006652 000251
04400 1728
04500 1729 006654 022767 177777 171556
04600 1730 006662 001402
04700 1731 006664 104002
04800 1732 006666 000252
04900 1733
05000 1734 006670 122767 000036 171506
05100 1735 006676 001402
05200 1736 006700 104000
05300 1737 006702 000253
05400 1738
05500 1738

```

```

*****
TEST 36: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
000000,000000 - 077777,177777 = 177777,177777
PS - 210, STACK POINTER - SP
*****

```

```

TST36: SCOPE
JSR R5, PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 077777,177777 ;SECOND OPERAND ON TOP
.WORD 000000,000000 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR

NOP
FSUB SP ;FLOATING SUBTRACT ON THE STACK

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA36 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER

```

```

HLT ;STACK POINTER FOULED UP
247 ;THE ERROR NUMBER IS 247
BR END36 ;SKIP REST OF TEST

```

```

TSA36: CMPB #210, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
250 ;THE ERROR NUMBER IS 250

```

```

CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 177777
251 ;THE ERROR NUMBER IS 251

```

```

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177777
252 ;THE ERROR NUMBER IS 252

```

```

END36: CMPB #36, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
253 ;THE ERROR NUMBER IS 253

```

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 42
00200 CVKACC.P11 16-AUG-78 08:41 TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW
00300
00400 1739
00500 1740
00600 1741
00700 1742
00800 1743
00900 1744
01000 1745
01100 1746 006704 104400
01200 1747 006706 004567 007122
01300 1748 006712 000252 125253
01400 1749 006716 000425 052525
01500 1750 006722 000257
01600 1751 006724 006752 000340
01700 1752
01800 1753 006730 000240
01900 1754 006732 075016
02000 1755
02100 1756 006734 004767 007134
02200 1757 006740 104002
02300 1758 006742 000254
02400 1759 006744 012706 000600
02500 1760 006750 000464
02600 1761
02700 1762 006752 004767 007150
02800 1763 006756 022706 000600
02900 1764 006762 001405
03000 1765 006764 012706 000600
03100 1766 006770 104000
03200 1767 006772 000255
03300 1768 006774 000452
03400 1769
03500 1770 006776 122767 000340 171426
03600 1771 007004 001402
03700 1772 007006 104000
03800 1773 007010 000256
03900 1774
04000 1775 007012 022767 006734 171416
04100 1776 007020 001402
04200 1777 007022 104001
04300 1778 007024 000257
04400 1779
04500 1780 007026 022767 000212 171404
04600 1781 007034 001402
04700 1782 007036 104002
04800 1783 007040 000260
04900 1784
05000 1785 007042 022767 000252 171372
05100 1786 007050 001402
05200 1787 007052 104004
05300 1788 007054 000261
05400 1789
05500 1790 007056 022767 125253 171360
05600 1791 007064 001402
05700 1792 007066 104004
05800 1793 007070 000262
05900 1794

```

```

*****
:TEST 37: FSUB (LSI-1) FLOATING SUBTRACT INSTRUCTION)
: 000425,052525 - 000252,125253 ==> UNDERFLOW
: PS(ON STACK) = 212, STACK POINTER = SP
*****

```

```

TST37: SCOPE
JSR R5, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 000252,125253 ;SECOND OPERAND ON TOP
.WORD 000425,052525 ;FIRST OPERAND ON BOTTOM
.WORD 257 ;PROCESSOR PRIORITY LEVEL
.WORD ISR37, 340 ;FIS TRAP VECTOR

```

```

NOP
FSUB SP ;FLOATING SUBTRACT ON THE STACK

```

```

RTA37: JSR PC, POPS ;POP THE 'ANSWER'
HLT+2 ;FIS TRAP DIDN'T OCCURE!
254 ;THE ERROR NUMBER IS 254
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
BR END37

```

```

ISR37: JSR PC, POPES ;POP ALL DATA OFF THE STACK
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ ISA37 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
HLT ;STACK POINTER FOULED UP
255 ;THE ERROR NUMBER IS 255
BR END37 ;SKIP REST OF TEST

```

```

ISA37: CMPB #340, $PSW ;CHECK PS AFTER FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 340
256 ;THE ERROR NUMBER IS 256

```

```

CMP #RTA37, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
257 ;THE ERROR NUMBER IS 257

```

```

CMP #212, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 212
260 ;THE ERROR NUMBER IS 260

```

```

CMP #000252, ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (000252) CHANGED
261 ;THE ERROR NUMBER IS 261

```

```

CMP #125253, ANS4 ;CHECK DATA FROM STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (125253) CHANGED
262 ;THE ERROR NUMBER IS 262

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900

CVKACC MACY11 30A(1052) 21-AUG-78 15:28
CVKACC.P11 16-AUG-78 08:41

1795	007072	022767	000425	171346
1796	007100	001402		
1797	007102	104006		
1798	007104	000263		
1799				
1800	007106	022767	052525	171334
1801	007114	001402		
1802	007116	104006		
1803	007120	000264		
1804				
1805	007122	122767	000037	171254
1806	007130	001402		
1807	007132	104000		
1808	007134	000265		
1809				
1810				

PAGE 43
TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW

SEQ 0051

END37:

CMP	#000425,ANS5	:CHECK DATA FROM STACK
BEQ	+.6	:BRANCH IF OK
HLT+6		:DATA ON STACK (000425) CHANGED
263		:THE ERROR NUMBER IS 263
CMP	#052525,ANS6	:CHECK DATA FROM STACK
BEQ	+.6	:BRANCH IF OK
HLT+6		:DATA ON STACK (052525) CHANGED
264		:THE ERROR NUMBER IS 264
CMPB	#37, \$TESTN	:CHECK THE TEST NUMBER
BEQ	+.6	:BRANCH IF OK
HLT		:WRONG TEST. PC MUST HAVE FOULED UP.
265		:THE ERROR NUMBER IS 265

```

00100 (VKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 44
00200 (VKACC.P11 16-AUG-78 08:41 TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW SEQ 0052
00300
00400 1811
00500 1812
00600 1813
00700 1814
00800 1815
00900 1816
01000 1817
01100 1818 007136 104400
01200 1819 007140 004567 007042
01300 1820 007144 177452 125252
01400 1821 007150 077652 125253
01500 1822 007154 000015
01600 1823 007156 007210 000344
01700 1824 007162 012703 000510
01800 1825
01900 1826 007166 000240
02000 1827 007170 075013
02100 1828
02200 1829 007172 004767 007042
02300 1830 007176 010367 171232
02400 1831 007202 104002
02500 1832 007204 000266
02600 1833 007206 000464
02700 1834
02800 1835 007210 004767 007054
02900 1836 007214 010367 171214
03000 1837 007220 122767 000344 171204
03100 1838 007226 001402
03200 1839 007230 104000
03300 1840 007232 000267
03400 1841
03500 1842 007234 022767 000510 171172
03600 1843 007242 001402
03700 1844 007244 104000
03800 1845 007246 000270
03900 1846
04000 1847 007250 022767 007172 171160
04100 1848 007256 001402
04200 1849 007260 104001
04300 1850 007262 000271
04400 1851
04500 1852 007264 022767 000002 171146
04600 1853 007272 001402
04700 1854 007274 104002
04800 1855 007276 000272
04900 1856
05000 1857 007300 022767 177452 171134
05100 1858 007306 001402
05200 1859 007310 104004
05300 1860 007312 000273
05400 1861
05500 1862 007314 022767 125252 171122
05600 1863 007322 001402
05700 1864 007324 104004
05800 1865 007326 000274

```

```

*****
:TEST 40: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
: 077652,125253 - 177452,125252 ==> OVERFLOW
: PS(ON STACK) = 002, STACK POINTER = R3
*****
TST40: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
.WORD 177452,125252 ;SECOND OPERAND ON TOP
.WORD 077652,125253 ;FIRST OPERAND ON BOTTOM
.WORD 015 ;PROCESSOR PRIORITY LEVEL
.WORD ISR40, 344 ;FIS TRAP VECTOR
MOV #STACK0,R3 ;SET UP R3 AS STACK POINTER

NOP
FSUB R3 ;FLOATING SUBTRACT ON THE R3 STACK

RTA40: JSR PC, POPR ;POP THE 'ANSWER'
MOV R3, $SP ;SAVE STACK POINTER (R3)
HLT+2 ;FIS TRAP DIDN'T OCCURE'
266 ;THE ERROR NUMBER IS 266
BR END40

ISR40: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
MOV R3, $SP ;SAVE STACK POINTER (R3)
CMPB #344, $PSW ;CHECK PS AFTER FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 344
267 ;THE ERROR NUMBER IS 267

CMP #STACK0,$SP ;CHECK THE STACK POINTER (R3)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R3) NOT EQUAL TO #STACK0
270 ;THE ERROR NUMBER IS 270

CMP #RTA40, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
271 ;THE ERROR NUMBER IS 271

CMP #002, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 002
272 ;THE ERROR NUMBER IS 272

CMP #177452,ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (177452) CHANGED
273 ;THE ERROR NUMBER IS 273

CMP #125252,ANS4 ;CHECK DATA FROM STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (125252) CHANGED
274 ;THE ERROR NUMBER IS 274

```

```

00100 (VKACC MARV11 30A(1052) 21-AUG-78 15:28 PAGE 45
00200 (VKACC.P'1 16-AUG-78 08:41 TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW SEQ 0053
00300
00400 1867 007330 022767 077652 171110 CMP #077652,ANS5 :CHECK DATA FROM STACK
00500 1868 007336 001402 BEQ .+6 :BRANCH IF OK
00600 1869 007340 104006 HLT+6 :DATA ON STACK (077652) CHANGED
00700 1870 007342 000275 275 :THE ERROR NUMBER IS 275
00800 1871
00900 1872 007344 022767 125253 171076 CMP #125253,ANS6 :CHECK DATA FROM STACK
01000 1873 007352 001402 BEQ .+6 :BRANCH IF OK
01100 1874 007354 104006 HLT+6 :DATA ON STACK (125253) CHANGED
01200 1875 007356 000276 276 :THE ERROR NUMBER IS 276
01300 1876
01400 1877 007360 122767 000040 171016 END40. CMPB #40, $TESTN :CHECK THE TEST NUMBER
01500 1878 007366 001402 BEQ .+6 :BRANCH IF OK
01600 1879 007370 104000 HLT :WRONG TEST PC MUST HAVE FOULED UP.
01700 1880 007372 000277 277 :THE ERROR NUMBER IS 277
01800 1881
01900 1882

```


00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 47
CVKACC.P11 16-AUG-78 08:41
1929
1930
1931
1932
1933
1934
1935
1936 007530 104400
1937 007532 004567 006450
1938 007536 052345 123456
1939 007542 140200 000000
1940 007546 000343
1941 007550 016456 000340
1942 007554 012702 000510
1943
1944 007560 000240
1945 007562 075022
1946
1947 007564 004767 006450
1948 007570 010267 170640
1949 007574 122767 000210 170630
1950 007602 001402
1951 007604 104000
1952 007606 000305
1953
1954 007610 022767 000514 170616
1955 007616 001402
1956 007620 104000
1957 007622 000306
1958
1959 007624 022767 152345 170604
1960 007632 001402
1961 007634 104002
1962 007636 000307
1963
1964 007640 022767 123456 170572
1965 007646 001402
1966 007650 104002
1967 007652 000310
1968
1969 007654 122767 000042 170522
1970 007662 001402
1971 007664 104000
1972 007666 000311
1973
1974

FMUL TEST SECTION

SEQ 0055

:TEST 42: FMUL (LS!-11 FLOATING MULTIPLY INSTRUCTION)
: 140200,000000 * 052345,123456 = 152345,123456
: PS 210, STACK POINTER R2
:*****

TST42: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
.WORD 052345,123456 ;SECOND OPERAND ON TOP
.WORD 140200,000000 ;FIRST OPERAND ON BOTOM
.WORD 343 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R2 ;SET UP STACK POINTER

NOP
FMUL R2 ;FLOATING MULTIPLY ON THE R2 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R2, \$SP ;SAVE 'STACK POINTER'
CMPB #210, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
305 ;THE ERROR NUMBER IS 305

CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R2)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R2) NOT EQUAL TO #STACK4
306 ;THE ERROR NUMBER IS 306

CMP #152345,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 152345
307 ;THE ERROR NUMBER IS 307

CMP #123456,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 123456
310 ;THE ERROR NUMBER IS 310

END42: CMPB #42, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
311 ;THE ERROR NUMBER IS 311

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MARY11 30A(1052) 21-AUG-78 15:28 PAGE 48
CVKACC.P1 16-AUG-78 08:41
1975
1976
1977
1978
1979
1980
1981
1982 007670 104400
1983 007672 004567 006310
1984 007676 135753 024642
1985 007702 100125 052525
1986 007706 000117
1987 007710 016456 000340
1988 007714 012705 000510
1989
1990 007720 000240
1991 007722 075025
1992
1993 007724 004767 006310
1994 007730 010567 170500
1995 007734 122767 000004 170470
1996 007742 001402
1997 007744 104000
1998 007746 000312
1999
2000 007750 022767 000514 170456
2001 007756 001402
2002 007760 104000
2003 007762 000313
2004
2005 007764 005767 170446
2006 007770 001402
2007 007772 104002
2008 007774 000314
2009
2010 007776 005767 170436
2011 010002 001402
2012 010004 104002
2013 010006 000315
2014
2015 010010 122767 000043 170366
2016 010016 001402
2017 010020 104000
2018 010022 000316
2019
2020

FMUL TEST SECTION

SEQ 0056

```

:*****
:TEST 43:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
:              100125,052525 * 135753,024642 = 000000,000000
:              PS = 004,      STACK POINTER = R5
:*****
TST43:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
        .WORD   135753,024642      ;SECOND OPERAND ON TOP
        .WORD   100125,052525      ;FIRST OPERAND ON BOTOM
        .WORD   117                ;PROCESSOR PRIORITY LEVEL
        .WORD   TRAPER, 340        ;FIS TRAP VECTOR
        MOV     #STACK0,R5        ;SET UP STACK POINTER

        NOP
        FMUL   R5                ;FLOATING MULTIPLY ON THE R5 STACK

        JSR   PC,      POPR      ;POP THE ANSWER
        MOV  R5,      $SP      ;SAVE 'STACK POINTER'
        CMPB #004,    $PSW     ;CHECK PS (EXCEPT T BIT)
        BEQ  .+6           ;BRANCH IF OK
        HLT  .+6           ;PS NOT EQUAL TO 004
        HLT  312          ;THE ERROR NUMBER IS 312

        CMP   #STACK4,$SP     ;CHECK THE STACK POINTER (R5)
        BEQ  .+6           ;BRANCH IF OK
        HLT  .+6           ;STACK POINTER (R5) NOT EQUAL TO #STACK4
        HLT  313          ;THE ERROR NUMBER IS 313

        TST  ANS1           ;CHECK FIRST HALF OF ANSWER
        BEQ  .+6           ;BRANCH IF OK
        HLT+2 314          ;ANS1 NOT EQUAL TO 000000
        HLT  314          ;THE ERROR NUMBER IS 314

        TST  ANS2           ;CHECK SECOND HALF OF ANSWER
        BEQ  .+6           ;BRANCH IF OK
        HLT+2 315          ;ANS2 NOT EQUAL TO 000000
        HLT  315          ;THE ERROR NUMBER IS 315

END43:  CMPB  #43,    $TESTN    ;CHECK THE TEST NUMBER
        BEQ  .+6           ;BRANCH IF OK
        HLT  .+6           ;WRONG TEST. PC MUST HAVE FOULED UP.
        HLT  316          ;THE ERROR NUMBER IS 316

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 49
CVKACC.P11 16-AUG-78 08:41
2021
2022
2023
2024
2025
2026
2027
2028 010024 104400
2029 010026 004567 006154
2030 010032 000052 125252
2031 010036 161616 161616
2032 010042 000217
2033 010044 016456 000340
2034 010050 012703 000510
2035
2036 010054 000240
2037 010056 075023
2038
2039 010060 004767 006154
2040 010064 010367 170344
2041 010070 122767 000204 170334
2042 010076 001402
2043 010100 104000
2044 010102 000317
2045
2046 010104 022767 000514 170322
2047 010112 001402
2048 010114 104000
2049 010116 000320
2050
2051 010120 005767 170312
2052 010124 001402
2053 010126 104002
2054 010130 000321
2055
2056 010132 005767 170302
2057 010136 001402
2058 010140 104002
2059 010142 000322
2060
2061 010144 122767 000044 170232
2062 010152 001402
2063 010154 104000
2064 010156 000323
2065
2066

FMUL TEST SECTION

SEQ 0057

:TEST 44: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
: 161616,161616 * 000052,125252 = 000000,000000
: PS = 204, STACK POINTER R3
:*****

TST44: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
.WORD 000052,125252 ;SECOND OPERAND ON TOP
.WORD 161616,161616 ;FIRST OPERAND ON BOTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R3 ;SET UP STACK POINTER

NOP
FMUL R3 ;FLOATING MULTIPLY ON THE R3 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R3, \$SP ;SAVE 'STACK POINTER'
CMPB #204, \$PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT 317 ;PS NOT EQUAL TO 204
;THE ERROR NUMBER IS 317

CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R3)
BEQ .+6 ;BRANCH IF OK
HLT 320 ;STACK POINTER (R3) NOT EQUAL TO #STACK4
;THE ERROR NUMBER IS 320

TST ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 321 ;ANS1 NOT EQUAL TO 000000
;THE ERROR NUMBER IS 321

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 322 ;ANS2 NOT EQUAL TO 000000
;THE ERROR NUMBER IS 322

END44: CMPB #44, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT 323 ;WRONG TEST! PC MUST HAVE FOULED UP.
;THE ERROR NUMBER IS 323


```

00300
00400
00500
00600
00700
00800
00900
01000
01100 2120 010314 104400
01200 2121 010316 004567 005664
01300 2122 010322 114100 000001
01400 2123 010326 124252 125252
01500 2124 010332 000200
01600 2125 010334 016456 000340
01700 2126 010340 012701 000510
01800
01900 2128 010344 000240
02000 2129 010346 075021
02100
02200 2131 010350 004767 005664
02300 2132 010354 010167 170054
02400 2133 010360 122767 000200 170044
02500 2134 010366 001402
02600 2135 010370 104000
02700 2136 010372 000331
02800
02900 2138 010374 022767 000514 170032
03000 2139 010402 001402
03100 2140 010404 104000
03200 2141 010406 000332
03300
03400 2143 010410 022767 000200 170020
03500 2144 010416 001402
03600 2145 010420 104002
03700 2146 010422 000333
03800
03900 2148 010424 005767 170010
04000 2149 010430 001402
04100 2150 010432 104002
04200 2151 010434 000334
04300
04400 2153 010436 122767 000046 167740
04500 2154 010444 001402
04600 2155 010446 104000
04700 2156 010450 000335
04800
04900

```

```

*****
:TEST 46: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
: 124252,125252 * 114100,000001 = 000200,000000
: PS 200, STACK POINTER = R1
*****
TST46: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
.WORD 114100,000001 ;SECOND OPERAND ON TOP
.WORD 124252,125252 ;FIRST OPERAND ON BOTOM
.WORD 200 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R1 ;SET UP STACK POINTER

NOP
FMUL R1 ;FLOATING MULTIPLY ON THE R1 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R1, $SP ;SAVE 'STACK POINTER'
CMPB #200, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
331 ;THE ERROR NUMBER IS 331

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R1)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK4
332 ;THE ERROR NUMBER IS 332

CMP #000200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 000200
333 ;THE ERROR NUMBER IS 333

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
334 ;THE ERROR NUMBER IS 334

END46: CMPB #46, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED JP.
335 ;THE ERROR NUMBER IS 335

```

```

00100
00200
00300
00400 2159
00500 2160
00600 2161
00700 2162
00800 2163
00900 2164
01000 2165
01100 2166 010452 104400
01200 2167 010454 004567 005700
01300 2168 010460 010504
01400 2169 010462 104000 104000
01500 2170 010466 104000 105004
01600 2171 010472 000252
01700 2172 010474 016456 000340
01800 2173
01900 2174 010500 000240
02000 2175 010502 075017
02100 2176 010504 104000
02200 2177 010506 104000
02300 2178 010510 104000
02400 2179 010512 105004
02500 2180
02600 2181 010514 004767 005670
02700 2182 010520 122767 000210 167704
02800 2183 010526 001402
02900 2184 010530 104000
03000 2185 010532 000336
03100 2186
03200 2187 010534 022767 104000 167674
03300 2188 010542 001402
03400 2189 010544 104002
03500 2190 010546 000337
03600 2191
03700 2192 010550 022767 104000 167662
03800 2193 010556 001402
03900 2194 010560 104002
04000 2195 010562 000340
04100 2196
04200 2197 010564 022767 100401 167650
04300 2198 010572 001402
04400 2199 010574 104004
04500 2200 010576 000341
04600 2201
04700 2202 010600 005767 167640
04800 2203 010604 001402
04900 2204 010606 104004
05000 2205 010610 000342
05100 2206
05200 2207 010612 122767 000047 167564
05300 2208 010620 001402
05400 2209 010622 104000
05500 2210 010624 000343
05600
05700

```

```

*****
TEST 47: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
          104000,105004 - 104000,104000 = 100401,000000
          PS - 210, STACK POINTER - PC
*****

```

```

TST47: SCOPE
        JSR R5, PUSH7 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
        .WORD STK47 ;TOP OF STACK
        .WORD 104000,104000 ;SECOND OPERAND ON TOP
        .WORD 104000,105004 ;FIRST OPERAND ON BOTTOM
        .WORD 252 ;PROCESSOR PRIORITY LEVEL
        .WORD TRAPER,340 ;FIS TRAP VECTOR

STK47: NOP
        FSUB PC ;FLOATING SUBTRACT ON FOLLOWING 4 WORDS
        104000 ;SHOULD CONTAIN 104000
        104000 ;SHOULD CONTAIN 104000
        104000 ;BEFORE FSUB, 104000; AFTER, 100401
        105004 ;BEFORE FSUB, 105004; AFTER, 000000

        JSR PC, POP7 ;POP THE ANSWER
        CMPB #210, $PSW ;CHECK PS (EXCEPT T BIT)
        BEQ .+6 ;BRANCH IF OK
        HLT ;PS NOT EQUAL TO 210
        336 ;THE ERROR NUMBER IS 336

        CMP #104000,ANS1 ;CHECK FIRST HALF OF INPUT DATA (STK47)
        BEQ .+6 ;BRANCH IF OK
        HLT+2 ;ANS1 NOT EQUAL TO 104000
        337 ;THE ERROR NUMBER IS 337

        CMP #104000,ANS2 ;CHECK SECOND HALF OF INPUT DATA (STK47+2)
        BEQ .+6 ;BRANCH IF OK
        HLT+2 ;ANS2 NOT EQUAL TO 104000
        340 ;THE ERROR NUMBER IS 340

        CMP #100401,ANS3 ;CHECK FIRST HALF OF ANSWER
        BEQ .+6 ;BRANCH IF OK
        HLT+4 ;ANS3 NOT EQUAL TO 100401
        341 ;THE ERROR NUMBER IS 341

        TST ANS4 ;CHECK SECOND HALF OF ANSWER
        BEQ .+6 ;BRANCH IF OK
        HLT+4 ;ANS4 NOT EQUAL TO 000000
        342 ;THE ERROR NUMBER IS 342

END47: CMPB #47, $TESTN ;CHECK THE TEST NUMBER
        BEQ .+6 ;BRANCH IF OK
        HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
        343 ;THE ERROR NUMBER IS 343

```

```

00300
00400 2213
00500 2214
00600 2215
00700 2216
00800 2217
00900 2218
01000 2219
01100 2220 010626 104400
01200 2221 010630 004567 005524
01300 2222 010634 010660
01400 2223 010636 104000 104000
01500 2224 010642 134600 073601
01600 2225 010646 000246
01700 2226 010650 016456 000340
01800 2227
01900 2228 010654 000240
02000 2229 010656 075027
02100 2230 010660 104000
02200 2231 010662 104000
02300 2232 010664 134600
02400 2233 010666 073601
02500 2234
02600 2235 010670 004767 005514
02700 2236 010674 122767 000200 167530
02800 2237 010702 001402
02900 2238 010704 104000
03000 2239 010706 000344
03100 2240
03200 2241 010710 022767 104000 167520
03300 2242 010716 001402
03400 2243 010720 104002
03500 2244 010722 000345
03600 2245
03700 2246 010724 022767 104000 167506
03800 2247 010732 001402
03900 2248 010734 104002
04000 2249 010736 000346
04100 2250
04200 2251 010740 022767 000401 167474
04300 2252 010746 001402
04400 2253 010750 104004
04500 2254 010752 000347
04600 2255
04700 2256 010754 005767 167464
04800 2257 010760 001402
04900 2258 010762 104004
05000 2259 010764 000350
05100 2260
05200 2261 010766 122767 000050 167410
05300 2262 010774 001402
05400 2263 010776 104000
05500 2264 011000 000351
05600 2265
05700 2266
05800 2267
05900 2268

```

```

:*****
:TEST 50: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
: 134600,073601 * 104000,104000 = 000401,000000
: PS = 200, STACK POINTER PC
:*****
TST50: SCOPE
JSR R5, PUSH7 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD STK50 ;TOP OF STACK
.WORD 104000,104000 ;SECOND OPERAND ON TOP
.WORD 134600,073601 ;FIRST OPERAND ON BOTTOM
.WORD 246 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER,340 ;FIS TRAP VECTOR

NOP
FMUL PC ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
STK50: 104000 ;SHOULD CONTAIN 104000
104000 ;SHOULD CONTAIN 104000
134600 ;BEFORE FMUL, 134600; AFTER, 000401
073601 ;BEFORE FMUL, 073601; AFTER, 000000

JSR PC, POP7 ;POP THE ANSWER
CMPB #200, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 200
344 ;THE ERROR NUMBER IS 344

CMP #104000,ANS1 ;CHECK FIRST HALF OF INPUT DATA (STK50)
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 104000
345 ;THE ERROR NUMBER IS 345

CMP #104000,ANS2 ;CHECK SECOND HALF OF INPUT DATA (STK50+2)
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 104000
346 ;THE ERROR NUMBER IS 346

CMP #000401,ANS3 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO 000401
347 ;THE ERROR NUMBER IS 347

TST ANS4 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO 000000
350 ;THE ERROR NUMBER IS 350

END50: CMPB #50, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
351 ;THE ERROR NUMBER IS 351
:*****

```

```

00300
00400 2269
00500 2270
00600 2271
00700 2272
00800 2273
00900 2274 011002 104400
01000 2275 011004 004567 005176
01100 2276 011010 114100 000000
01200 2277 011014 024252 125252
01300 2278 011020 000305
01400 2279 011022 011054 000057
01500 2280 011026 012700 000510
01600 2281
01700 2282 011032 000240
01800 2283 011034 075020
01900 2284
02000 2285 011036 004767 005176
02100 2286 011042 010067 167366
02200 2287 011046 104002
02300 2288 011050 000352
02400 2289 011052 000463
02500 2290
02600 2291 011054 004767 005210
02700 2292 011060 010067 167350
02800 2293 011064 122767 000057 167340
02900 2294 011072 001402
03000 2295 011074 104000
03100 2296 011076 000353
03200 2297
03300 2298 011100 022767 000510 167326
03400 2299 011106 001402
03500 2300 011110 104000
03600 2301 011112 000354
03700 2302
03800 2303 011114 022767 011036 167314
03900 2304 011122 001402
04000 2305 011124 104001
04100 2306 011126 000355
04200 2307
04300 2308 011130 022767 000212 167302
04400 2309 011136 001402
04500 2310 011140 104002
04600 2311 011142 000356
04700 2312
04800 2313 011144 022767 114100 167270
04900 2314 011152 001402
05000 2315 011154 104004
05100 2316 011156 000357
05200 2317
05300 2318 011160 005767 167260
05400 2319 011164 001402
05500 2320 011166 104004
05600 2321 011170 000360
05700 2322
05800 2323 011172 022767 024252 167240
05900 2324 011200 001402

```

```

:TEST 51: FMUL (LSJ-11 FLOATING MULTIPLY INSTRUCTION)
: 024252,125252 * 114100,000000 ==> UNDERFLOW
: PS(ON STACK) = 212, STACK POINTER = R0
:*****

```

```

TST51: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
.WORD 114100,000000 ;SECOND OPERAND ON TOP
.WORD 024252,125252 ;FIRST OPERAND ON BOTTOM
.WORD 305 ;PROCESSOR PRIORITY LEVEL
.WORD ISR51, 057 ;FIS TRAP VECTOR
MOV #STACK0,R0 ;SET UP R0 AS STACK POINTER

```

```

NOP
FMLI R0 ;FLOATING MULTIPLY ON THE R0 STACK

```

```

RTA51: JSR PC, POPR ;POP THE 'ANSWER'
MOV R0, $SP ;SAVE STACK POINTER (R0)
HLT+2 ;FIS TRAP DIDN'T OCCURE
352 ;THE ERROR NUMBER IS 352
BR END51

```

```

ISR51: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
MOV R0, $SP ;SAVE STACK POINTER (R0)
CMPB #057, $PSW ;CHECK PS AFTER FIS TRAP
BEQ +6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 057
353 ;THE ERROR NUMBER IS 353

```

```

CMP #STACK0,$SP ;CHECK THE STACK POINTER (R0)
BEQ +6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK0
354 ;THE ERROR NUMBER IS 354

```

```

CMP #RTA5,ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ +6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
355 ;THE ERROR NUMBER IS 355

```

```

CMP #212,ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ +6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 212
356 ;THE ERROR NUMBER IS 356

```

```

CMP #114100,ANS3 ;CHECK DATA FROM THE STACK
BEQ +6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (114100) CHANGED
357 ;THE ERROR NUMBER IS 357

```

```

TST ANS4 ;CHECK DATA FROM STACK
BEQ +6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (000000) CHANGED
360 ;THE ERROR NUMBER IS 360

```

```

CMP #024252,ANS5 ;CHECK DATA FROM STACK
BEQ +6 ;BRANCH IF OK

```

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 55
00200 CVKACC.P11 16-AUG-78 08:41 TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW SEQ 0063
00300
00400 2325 011202 104006 HLT+6 ;DATA ON STACK (024252) CHANGED
00500 2326 011204 000361 361 ;THE ERROR NUMBER IS 361
00600 2327
00700 2328 011206 022767 125252 167234 CMP #125252,ANS6 ;CHECK DATA FROM STACK
00800 2329 011214 001402 BEQ .+6 ;BRANCH IF OK
00900 2330 011216 104006 HLT+6 ;DATA ON STACK (125252) CHANGED
01000 2331 011220 000362 362 ;THE ERROR NUMBER IS 362
01100 2332
01200 2333 011222 122767 000051 167154 ENDS1: CMPB #51, $TESTN ;CHECK THE TEST NUMBER
01300 2334 011230 001402 BEQ .+6 ;BRANCH IF OK
01400 2335 011232 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
01500 2336 011234 000363 363 ;THE ERROR NUMBER IS 363
01600 2337
01700 2338

```


00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 56
 00200 CVKACC. P11 16-AUG-78 08:41
 00300
 00400 2339
 00500 2340
 00600 2341
 00700 2342
 00800 2343
 00900 2344
 01000 2345
 01100 2346 011236 104400
 01200 2347 011240 004567 004570
 01300 2348 011244 041500 000001
 01400 2349 011250 076452 125252
 01500 2350 011254 000105
 01600 2351 011256 011304 000357
 01700 2352
 01800 2353 011262 000240
 01900 2354 011264 075026
 02000 2355
 02100 2356 011266 004767 004602
 02200 2357 011272 104002
 02300 2358 011274 000364
 02400 2359 011276 012706 000600
 02500 2360 011302 000464
 02600 2361
 02700 2362 011304 004767 004616
 02800 2363 011310 022706 000600
 02900 2364 011314 001405
 03000 2365 011316 012706 000600
 03100 2366 011322 104000
 03200 2367 011324 000365
 03300 2368 011326 000452
 03400 2369
 03500 2370 011330 122767 000357 167074
 03600 2371 011336 001402
 03700 2372 011340 104000
 03800 2373 011342 000366
 03900 2374
 04000 2375 011344 022767 011266 167064
 04100 2376 011352 001402
 04200 2377 011354 104001
 04300 2378 011356 000367
 04400 2379
 04500 2380 011360 022767 000002 167052
 04600 2381 011366 001402
 04700 2382 011370 104002
 04800 2383 011372 000370
 04900 2384
 05000 2385 011374 022767 041500 167040
 05100 2386 011402 001402
 05200 2387 011404 104004
 05300 2388 011406 000371
 05400 2389
 05500 2390 011410 022767 000001 167026
 05600 2391 011416 001402
 05700 2392 011420 104004
 05800 2393 011422 000372
 05900 2394

TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW

SEQ 0064

```

:*****
:TEST 52:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
:              076452,125252 * 041500,000001 ==> OVERFLOW
:              PS(ON STACK) = 002,      STACK POINTER = SP
:*****

```

```

TST52:  SCOPE
        JSR      R5,      PUSH5      ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
        .WORD   041500,000001      ;SECOND OPERAND ON TOP
        .WORD   076452,125252      ;FIRST OPERAND ON BOTTOM
        .WORD   105                ;PROCESSOR PRIORITY LEVEL
        .WORD   ISR52, 357         ;FIS TRAP VECTOR

```

```

        NOP
        FMUL    SP                ;FLOATING MULTIPLY ON THE STACK

```

```

RTA52:  JSR      PC,      POPS       ;POP THE 'ANSWER'
        HLT+2
        364
        MOV     #BEGIN, SP        ;FIS TRAP DIDN'T OCCURE!
        BR      END52            ;THE ERROR NUMBER IS 364
        ;RESTORE THE STACK POINTER

```

```

ISR52:  JSR      PC,      POPES      ;POP ALL DATA OFF THE STACK
        CMP     #BEGIN, SP        ;CHECK THE STACK POINTER
        BEQ     ISA52            ;BRANCH IF OK
        MOV     #BEGIN, SP        ;RESTORE THE STACK POINTER
        HLT     365              ;STACK POINTER FOULED UP
        BR      END52            ;THE ERROR NUMBER IS 365
        ;SKIP REST OF TEST

```

```

ISA52:  CMPB    #357,  $PSW        ;CHECK PS AFTER FIS TRAP
        BEQ     .+6              ;BRANCH IF OK
        HLT     366              ;PS AFTER FIS TRAP NOT EQUAL TO 357
        ;THE ERROR NUMBER IS 366

```

```

        CMP     #RTA52, ANS1      ;CHECK FIS TRAP RETURN ADDRESS
        BEQ     .+6              ;BRANCH IF OK
        HLT+1  367              ;FIS TRAP AT WRONG ADDRESS
        ;THE ERROR NUMBER IS 367

```

```

        CMP     #002,  ANS2      ;CHECK PS BEFORE FIS TRAP
        BEQ     .+6              ;BRANCH IF OK
        HLT+2  370              ;PS AT FIS TRAP TIME NOT 002
        ;THE ERROR NUMBER IS 370

```

```

        CMP     #041500,ANS3      ;CHECK DATA FROM THE STACK
        BEQ     .+6              ;BRANCH IF OK
        HLT+4  371              ;DATA ON STACK (041500) CHANGED
        ;THE ERROR NUMBER IS 371

```

```

        CMP     #000001,ANS4      ;CHECK DATA FROM STACK
        BEQ     .+6              ;BRANCH IF OK
        HLT+4  372              ;DATA ON STACK (000001) CHANGED
        ;THE ERROR NUMBER IS 372

```

```

00100
00200
00300
00400 2395 011424 022767 076452 167014      CMP      #076452,ANS5      :CHECK DATA FROM STACK
00500 2396 011432 001402                        BEQ      .+6              :BRANCH IF OK
00600 2397 011434 104006                        HLT+6    373              :DATA ON STACK (076452) CHANGED
00700 2398 011436 000373                        373                      :THE ERROR NUMBER IS 373
00800 2399
00900 2400 011440 022767 125252 167002      CMP      #125252,ANS6      :CHECK DATA FROM STACK
01000 2401 011446 001402                        BEQ      .+6              :BRANCH IF OK
01100 2402 011450 104006                        HLT+6    374              :DATA ON STACK (125252) CHANGED
01200 2403 011452 000374                        374                      :THE ERROR NUMBER IS 374
01300 2404
01400 2405 011454 122767 000052 166722      END52:  CMPB     #52,      $TESTN    :CHECK THE TEST NUMBER
01500 2406 011462 001402                        BEQ      .+6              :BRANCH IF OK
01600 2407 011464 104000                        HLT      375              :WRONG TEST. PC MUST HAVE FOILED UP.
01700 2408 011466 000375                        375                      :THE ERROR NUMBER IS 375
01800 2409
01900 2410

```

```

00300 2411
00400 2412
00500 2413
00600 2414
00700 2415
00800 2416
00900 2417
01000 2418 011470 104400
01100 2419 011472 004567 004510
01200 2420 011476 127652 125252
01300 2421 011502 167452 125251
01400 2422 011506 000111
01500 2423 011510 016456 000340
01600 2424 011514 012700 000510
01800 2425
01900 2426 011520 000240
02000 2427 011522 075030
02100 2428
02200 2429 011524 004767 004510
02300 2430 011530 010067 166700
02400 2431 011534 105767 166672
02500 2432 011540 001402
02600 2433 011542 104000
02700 2434 011544 000376
02800 2435
02900 2436 011546 022767 000514 166660
03000 2437 011554 001402
03100 2438 011556 104000
03200 2439 011560 000377
03300 2440
03400 2441 011562 022767 077777 166646
03500 2442 011570 001402
03600 2443 011572 104002
03700 2444 011574 000400
03800 2445
03900 2446 011576 022767 177776 166634
04000 2447 011604 001402
04100 2448 011606 104002
04200 2449 011610 000401
04300 2450
04400 2451 011612 122767 000053 166564
04500 2452 011620 001402
04600 2453 011622 104000
04700 2454 011624 000402
04800 2455
04900 2456

```

```

*****
:TEST 53: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
: 167452,125251 / 127652,125252 = 077777,177776
: PS = 000, STACK POINTER = R0
*****

TST53: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
.WORD 127652,125252 ;SECOND OPERAND ON TOP
.WORD 167452,125251 ;FIRST OPERAND ON BOTTOM
.WORD 111 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R0 ;CHECK STACK POINTER

NOP
FDIV R0 ;FLOATING DIVIDE ON THE R0 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R0, $SP ;SAVE 'STACK POINTER'
TSTB $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
376 ;THE ERROR NUMBER IS 376

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R0)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
377 ;THE ERROR NUMBER IS 377

CMP #077777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 077777
400 ;THE ERROR NUMBER IS 400

CMP #177776,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177776
401 ;THE ERROR NUMBER IS 401

END53: CMPB #53, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
402 ;THE ERROR NUMBER IS 402

```

```

00300
00400 2457
00500 2458
00600 2459
00700 2460
00800 2461
00900 2462
01000 2463
01100 2464 011626 104400
01200 2465 011630 004567 004200
01300 2466 011634 027652 125253
01400 2467 011640 167452 125252
01500 2468 011644 000300
01600 2469 011646 016456 000340
01700 2470
01800 2471 011652 000240
01900 2472 011654 075036
02000 2473
02100 2474 011656 004767 004212
02200 2475 011662 022706 000600
02300 2476 011666 001405
02400 2477 011670 012706 000600
02500 2478 011674 104000
02600 2479 011676 000403
02700 2480 011700 000422
02800 2481
02900 2482 011702 122767 000210 166522
03000 2483 011710 001402
03100 2484 011712 104000
03200 2485 011714 000404
03300 2486
03400 2487 011716 022767 177777 166512
03500 2488 011724 001402
03600 2489 011726 104002
03700 2490 011730 000405
03800 2491
03900 2492 011732 022767 177777 166500
04000 2493 011740 001402
04100 2494 011742 104002
04200 2495 011744 000406
04300 2496
04400 2497 011746 122767 000054 166430
04500 2498 011754 001402
04600 2499 011756 104000
04700 2500 011760 000407
04800 2501
04900 2502

```

```

*****
:TEST 54: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
: 167452,125252 / 027652,125253 = 177777,177777
: PS 210, STACK POINTER = SP
*****

```

```

TST54: SCOPE
JSR R5, PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 027652,125253 ;SECOND OPERAND ON TOP
.WORD 167452,125252 ;FIRST OPERAND ON BOTTOM
.WORD 300 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR

NOP
FDIV SP ;FLOATING DIVIDE ON THE STACK

JSR PC, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA54 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER
HLT ;STACK POINTER FOULED UP
403 ;THE ERROR NUMBER IS 403
BR END54 ;SKIP REST OF TEST

TSA54: CMPB #210, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
404 ;THE ERROR NUMBER IS 404

CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 177777
405 ;THE ERROR NUMBER IS 405

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177777
406 ;THE ERROR NUMBER IS 406

END54: CMPB #54, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
407 ;THE ERROR NUMBER IS 407

```

00300 2503
00400 2504
00500 2505
00600 2506
00700 2507
00800 2508
00900 2509
01000 2510 011762 104400
01100 2511 011764 004567 004216
01200 2512 011770 065252 125252
01300 2513 011774 125252 125252
01400 2514 012000 000217
01500 2515 012002 016456 000340
01600 2516 012006 012702 000510
01700 2517
01800 2518 012012 000240
01900 2519 012014 075032
02000 2520
02100 2521 012016 004767 004216
02200 2522 012022 010267 166406
02300 2523 012026 122767 000210 166376
02400 2524 012034 001402
02500 2525 012036 104000
02600 2526 012040 000410
02700 2527
02800 2528 012042 022767 000514 166364
02900 2529 012050 001402
03000 2530 012052 104000
03100 2531 012054 000411
03200 2532
03300 2533 012056 022767 100200 166352
03400 2534 012064 001402
03500 2535 012066 104002
03600 2536 012070 000412
03700 2537
03800 2538 012072 005767 166342
03900 2539 012076 001402
04000 2540 012100 104002
04100 2541 012102 000413
04200 2542
04300 2543 012104 122767 000055 166272
04400 2544 012112 001402
04500 2545 012114 104000
04600 2546 012116 000414
04700 2547
04800 2548
04900 2548

```

:*****
:TEST 55: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
: 125252,125252 / 065252,125252 = 100200,000000
: PS 210, STACK POINTER = R2
:*****

```

```

TST55: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
.WORD 065252,125252 ;SECOND OPERAND ON TOP
.WORD 125252,125252 ;FIRST OPERAND ON BOTTOM
.WORD 217 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R2 ;CHECK STACK POINTER

NOP
FDIV R2 ;FLOATING DIVIDE ON THE R2 STACK

JSR PC, POPR ;POP THE ANSWER
MOV R2, $SP ;SAVE 'STACK POINTER'
CMPB #210, $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
410 ;THE ERROR NUMBER IS 410

CMP #STACK4,$SP ;CHECK THE STACK POINTER (R2)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R2) NOT EQUAL TO #STACK4
411 ;THE ERROR NUMBER IS 411

CMP #100200,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 100200
412 ;THE ERROR NUMBER IS 412

TST ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 000000
413 ;THE ERROR NUMBER IS 413

END55: CMPB #55, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
414 ;THE ERROR NUMBER IS 414

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900

CVKACC MAY11 30A(1052) 21-AUG-78 15:28
CVKACC.P11 16-AUG-78 08:41

PAGE 61
FDIV TEST SECTION

E 6

SEQ 0069

```
*****  
:TEST 56:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)  
:      00000,00000 / 140670,123456 = 00000,00000  
:      PS 004,      STACK POINTER = R3  
:*****
```

```
TST56: SCOPE  
JSR R5, PUSH R5 :PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY  
.WORD 140670,123456 :SECOND OPERAND ON TOP  
.WORD 00000,00000 :FIRST OPERAND ON BOTTOM  
.WORD 105 :PROCESSOR PRIORITY LEVEL  
.WORD TRAPER, 340 :FIS TRAP VECTOR  
MOV #STACK0,R3 :CHECK STACK POINTER  
  
NOP  
FDIV R3 :FLOATING DIVIDE ON THE R3 STACK  
  
JSR PC, POPR :POP THE ANSWER  
MOV R3, $SP :SAVE 'STACK POINTER'  
CMPB #004, $PSW :CHECK PS (EXCEPT T BIT)  
BEQ .+6 :BRANCH IF OK  
HLT :PS NOT EQUAL TO 004  
415 :THE ERROR NUMBER IS 415  
  
CMP #STACK4,$SP :CHECK THE STACK POINTER (R3)  
BEQ .+6 :BRANCH IF OK  
HLT :STACK POINTER (R3) NOT EQUAL TO #STACK4  
416 :THE ERROR NUMBER IS 416  
  
TST ANS1 :CHECK FIRST HALF OF ANSWER  
BEQ .+6 :BRANCH IF OK  
HLT+2 :ANS1 NOT EQUAL TO 000000  
417 :THE ERROR NUMBER IS 417  
  
TST ANS2 :CHECK SECOND HALF OF ANSWER  
BEQ .+6 :BRANCH IF OK  
HLT+2 :ANS2 NOT EQUAL TO 000000  
420 :THE ERROR NUMBER IS 420  
  
END56: CMPB #56, $TESTN :CHECK THE TEST NUMBER  
BEQ .+6 :BRANCH IF OK  
HLT :WRONG TEST. PC MUST HAVE FOULED UP.  
421 :THE ERROR NUMBER IS 421
```

2549
2550
2551
2552
2553
2554
2555
2556 012120 104400
2557 012122 004567 004060
2558 012126 140670 123456
2559 012132 000000 000000
2560 012136 000105
2561 012140 016456 000340
2562 012144 012703 000510
2563
2564 012150 000240
2565 012152 075033
2566
2567 012154 004767 004060
2568 012160 010367 166250
2569 012164 122767 000004 165240
2570 012172 001402
2571 012174 104000
2572 012176 000415
2573
2574 012200 022767 000514 166226
2575 012206 001402
2576 012210 104000
2577 012212 000416
2578
2579 012214 005767 166216
2580 012220 001402
2581 012222 104002
2582 012224 000417
2583
2584 012226 005767 166206
2585 012232 001402
2586 012234 104002
2587 012236 000420
2588
2589 012240 122767 000056 166136
2590 012246 001402
2591 012250 104000
2592 012252 000421
2593
2594

00300 2595
00400 2596
00500 2597
00600 2598
00700 2599
00800 2600
00900 2601
01000 2602 012254 104400
01100 2603 012256 004567 004076
01200 2604 012262 012306
01300 2605 012264 104000 104000
01400 2606 012270 102500 146000
01500 2607 012274 000357
01600 2608 012276 016456 000340
01700 2609
01800 2610 012302 000240
01900 2611 012304 075037
02000 2612 012306 104000
02100 2613 012310 104000
02200 2614 012312 102500
02300 2615 012314 146000
02400 2616
02500 2617 012316 004767 004066
02600 2618 012322 122767 000200 166102
02700 2619 012330 001402
02800 2620 012332 104000
02900 2621 012334 000422
03000 2622
03100 2623 012336 022767 104000 166072
03200 2624 012344 001402
03300 2625 012346 104002
03400 2626 012350 000423
03500 2627
03600 2628 012352 022767 104000 166060
03700 2629 012360 001402
03800 2630 012362 104002
03900 2631 012364 000424
04000 2632
04100 2633 012366 022767 036700 166046
04200 2634 012374 001402
04300 2635 012376 104004
04400 2636 012400 000425
04500 2637
04600 2638 012402 005767 166036
04700 2639 012406 001402
04800 2640 012410 104004
04900 2641 012412 000426
05000 2642
05100 2643 012414 122767 000357 165762
05200 2644 012422 001402
05300 2645 012424 104000
05400 2646 012426 000427
05500 2647
05600 2648
05700 2649
05800 2650
05900 2651

```

:*****
:TEST 57: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
:          102500,146000 / 104000,104000 = 036700,000000
:          PS = 200,          STACK POINTER = PC
:*****

```

```

TST57: SCOPE
        JSR      R5,      PUSH7      ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
        .WORD   STK57      ;TOP OF STACK
        .WORD   104000,104000 ;SECOND OPERAND ON TOP
        .WORD   102500,146000 ;FIRST OPERAND ON BOTTOM
        .WORD   357        ;PROCESSOR PRIORITY LEVEL
        .WORD   TRAPER,340 ;FIS TRAP VECTOR

STK57:  NOP
        FDIV   PC          ;FLOATING DIVIDE ON FOLLOWING 4 WORDS
        104000 ;SHOULD CONTAIN 104000
        104000 ;SHOULD CONTAIN 104000
        102500 ;BEFORE FDIV, 102500; AFTER, 036700
        146000 ;BEFORE FDIV, 146000; AFTER, 000000

        JSR      PC,      POP7       ;POP THE ANSWER
        CMPB    #200,    $PSW      ;CHECK PS (EXCEPT T BIT)
        BEQ     .+6         ;BRANCH IF OK
        HLT     422        ;PS NOT EQUAL TO 200
                          ;THE ERROR NUMBER IS 422

        CMP     #104000,ANS1      ;CHECK FIRST HALF OF INPUT DATA (STK57)
        BEQ     .+6         ;BRANCH IF OK
        HLT+2  423        ;ANS1 NOT EQUAL TO 104000
                          ;THE ERROR NUMBER IS 423

        CMP     #104000,ANS2      ;CHECK SECOND HALF OF INPUT DATA (STK57+2)
        BEQ     .+6         ;BRANCH IF OK
        HLT+2  424        ;ANS2 NOT EQUAL TO 104000
                          ;THE ERROR NUMBER IS 424

        CMP     #036700,ANS3      ;CHECK FIRST HALF OF ANSWER
        BEQ     .+6         ;BRANCH IF OK
        HLT+4  425        ;ANS3 NOT EQUAL TO 036700
                          ;THE ERROR NUMBER IS 425

        TST     ANS4          ;CHECK SECOND HALF OF ANSWER
        BEQ     .+6         ;BRANCH IF OK
        HLT+4  426        ;ANS4 NOT EQUAL TO 000000
                          ;THE ERROR NUMBER IS 426

END57:  CMPB    #57,      $TESTN   ;CHECK THE TEST NUMBER
        BEQ     .+6         ;BRANCH IF OK
        HLT     427        ;WRONG TEST: PC MUST HAVE FOULED UP.
                          ;THE ERROR NUMBER IS 427

```

:*****

OC100 (VKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 63
00200 (VKACC.P11 16-AUG-78 08:41

TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW

SEU 0071

```

00400 2651 ;TEST 60: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
00500 2652 ; 025252,125251 / 065252,125252 ==> UNDERFLOW
00600 2653 ; PS(ON STACK) = 012, STACK POINTER = R1
00700 2654 ;*****
00800 2655
00900 2656 012430 104400
01000 2657 012432 004567 003550 TST60: SCOPE
01100 2658 012436 065252 125252 JSR R5, PUSHR ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
01200 2659 012442 025252 125251 .WORD 065252,125252 ;SECOND OPERAND ON TOP
01300 2660 012446 000015 .WORD 025252,125251 ;FIRST OPERAND ON BOTTOM
01400 2661 012450 012502 000300 .WORD 015 ;PROCESSOR PRIORITY LEVEL
01500 2662 012454 012701 000510 .WORD ISR60, 300 ;FIS TRAP VECTOR
01600 2663 MOV #STACK0,R1 ;SET UP R1 AS STACK POINTER
01700 2664 012460 000240 NOP
01800 2665 012462 075031 FDIV R1 ;FLOATING DIVIDE ON THE R1 STACK
01900 2666
02000 2667 012464 004767 003550 RTA60: JSR PC, POPR ;POP THE 'ANSWER'
02100 2668 012470 010167 165740 MOV R1, $SP ;SAVE STACK POINTER (R1)
02200 2669 012474 104002 HLT+2 ;FIS TRAP DIDN'T OCCURE!
02300 2670 012476 000430 430 ;THE ERROR NUMBER IS 430
02400 2671 012500 000464 BR END60
02500 2672
02600 2673 012502 004767 003562 ISR60: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
02700 2674 012506 010167 165722 MOV R1, $SP ;SAVE STACK POINTER (R1)
02800 2675 012512 122767 000300 165712 CMPB #300, $PSW ;CHECK PS AFTER FIS TRAP
02900 2676 012520 001402 BEQ .+6 ;BRANCH IF OK
03000 2677 012522 104000 HLT ;PS AFTER FIS TRAP NOT EQUAL TO 300
03100 2678 012524 000431 431 ;THE ERROR NUMBER IS 431
03200 2679
03300 2680 012526 022767 000510 165700 CMP #STACK0,$SP ;CHECK THE STACK POINTER (R1)
03400 2681 012534 001402 BEQ .+6 ;BRANCH IF OK
03500 2682 012536 104000 HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK0
03600 2683 012540 000432 432 ;THE ERROR NUMBER IS 432
03700 2684
03800 2685 012542 022767 012464 165666 LMP #RTA60, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
03900 2686 012550 001402 BEQ .+6 ;BRANCH IF OK
04000 2687 012552 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
04100 2688 012554 000433 433 ;THE ERROR NUMBER IS 433
04200 2689
04300 2690 012556 022767 000012 165654 CMP #012, ANS2 ;CHECK PS BEFORE FIS TRAP
04400 2691 012564 001402 BEQ .+6 ;BRANCH IF OK
04500 2692 012566 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 012
04600 2693 012570 000434 434 ;THE ERROR NUMBER IS 434
04700 2694
04800 2695 012572 022767 065252 165642 CMP #065252,ANS3 ;CHECK DATA FROM THE STACK
04900 2696 012600 001402 BEQ .+6 ;BRANCH IF OK
05000 2697 012602 104004 HLT+4 ;DATA ON STACK (065252) CHANGED
05100 2698 012604 000435 435 ;THE ERROR NUMBER IS 435
05200 2699
05300 2700 012606 022767 125252 165630 CMP #125252,ANS4 ;CHECK DATA FROM STACK
05400 2701 012614 001402 BEQ .+6 ;BRANCH IF OK
05500 2702 012616 104004 HLT+4 ;DATA ON STACK (125252) CHANGED
05600 2703 012620 000436 436 ;THE ERROR NUMBER IS 436
05700 2704
05800 2705 012622 022767 025252 165610 CMP #025252,ANS5 ;CHECK DATA FROM STACK
05900 2706 012630 001402 BEQ .+6 ;BRANCH IF OK

```


00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700

CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 64
CVKACC.P'1 16-AUG-78 08:41

TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW

SEQ 007

2707 012632 104006
2708 012634 000437
2709
2710 012636 022767 125251 165604
2711 012644 001402
2712 012646 104006
2713 012650 000440
2714
2715 012652 122767 000060 165524
2716 012660 001402
2717 012662 104000
2718 012664 000441
2719
2720

HLT+6
437

:DATA ON STACK (025252) CHANGED
:THE ERROR NUMBER IS 437

CMP #125251,ANS6
BEQ .+6
HLT+6
440

:CHECK DATA FROM STACK
:BRANCH IF OK
:DATA ON STACK (125251) CHANGED
:THE ERROR NUMBER IS 440

END60: CMPB #60, \$TESTN
BEQ .+6
HLT
441

:CHECK THE TEST NUMBER
:BRANCH IF OK
:WRONG TEST. PC MUST HAVE FOULED UP.
:THE ERROR NUMBER IS 441

```

00200 2721
00300 2722
00400 2723
00500 2724
00600 2725
00700 2726
00800 2727
00900 2728 012666 104400
01000 2729 012670 004567 003312
01100 2730 012674 127652 125252
01200 2731 012700 067452 125252
01300 2732 012704 000242
01400 2733 012706 012740 000357
01500 2734 012712 012704 000510
01600 2735
01700 2736 012716 000240
01800 2737 012720 075034
01900 2738
02000 2739 012722 004767 003312
02100 2740 012726 010467 165502
02200 2741 012732 104002
02300 2742 012734 000442
02400 2743 012736 000464
02500 2744
02600 2745 012740 004767 003324
02700 2746 012744 010467 165464
02800 2747 012750 122767 000357 165454
02900 2748 012756 001402
03000 2749 012760 104000
03100 2750 012762 000443
03200 2751
03300 2752 012764 022767 000510 165442
03400 2753 012772 001402
03500 2754 012774 104000
03600 2755 012776 000444
03700 2756
03800 2757 013000 022767 012722 165430
03900 2758 013006 001402
04000 2759 013010 104001
04100 2760 013012 000445
04200 2761
04300 2762 013014 022767 000202 165416
04400 2763 013022 001402
04500 2764 013024 104002
04600 2765 013026 000446
04700 2766
04800 2767 013030 022767 127652 165404
04900 2768 013036 001402
05000 2769 013040 104004
05100 2770 013042 000447
05200 2771
05300 2772 013044 022767 125252 165322
05400 2773 013052 001402
05500 2774 013054 104004
05600 2775 013056 000450
05700
05800
05900

```

```

*****
:TEST #1: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
:067452,125252 / 127652,125252 ==> OVERFLOW
:PS(ON STACK) - 202, STACK POINTER - R4
*****

TST61: SCOPE
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
:WORD 127652,125252 ;SECOND OPERAND ON TOP
:WORD 067452,125252 ;FIRST OPERAND ON BOTTOM
:WORD 242 ;PROCESSOR PRIORITY LEVEL
:WORD ISR61, 357 ;FIS TRAP VECTOR
MOV #STACK0,R4 ;SET UP R4 AS STACK POINTER

NOP
FDIV R4 ;FLOATING DIVIDE ON THE R4 STACK

RTA61: JSR PC, POPR ;POP THE 'ANSWER'
MOV R4, $SP ;SAVE STACK POINTER (R4)
HLT+2 ;FIS TRAP DIDN'T OCCURE.
442 ;THE ERROR NUMBER IS 442
BR END61

ISR61: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
MOV R4, $SP ;SAVE STACK POINTER (R4)
CMPB #357, $PSW ;CHECK PS AFTER FIS TRAP
BEQ +6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 357
443 ;THE ERROR NUMBER IS 443

CMP #STACK0,$SP ;CHECK THE STACK POINTER (R4)
BEQ +6 ;BRANCH IF OK
HLT ;STACK POINTER (R4) NOT EQUAL TO #STACK0
444 ;THE ERROR NUMBER IS 444

CMP #RTA61, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ +6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
445 ;THE ERROR NUMBER IS 445

CMP #202, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ +6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 202
446 ;THE ERROR NUMBER IS 446

CMP #127652,ANS3 ;CHECK DATA FROM THE STACK
BEQ +6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (127652) CHANGED
447 ;THE ERROR NUMBER IS 447

CMP #125252,ANS4 ;CHECK DATA FROM STACK
BEQ +6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (125252) CHANGED
450 ;THE ERROR NUMBER IS 450

```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900

CVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 66
CVKACC.P11 16-AUG-78 08:41

TEST FLOATING DIV INSTRUCTION WITH OVERFLOW

SEQ 0074

2777 013060 022767 067452 165360
2778 013066 001402
2779 013070 104006
2780 013072 000451
2781
2782 013074 022767 125252 165346
2783 013102 001402
2784 013104 104006
2785 013106 000452
2786
2787 013110 122767 000061 165266
2788 013116 001402
2789 013120 104000
2790 013122 000453
2791
2792

END61:

CMP #067452,ANS5
BEQ .+6
HLT+6
451
CMP #125252,ANS6
BEQ .+6
HLT+6
452
CMPB #61, \$TESTN
BEQ .+6
HLT
453

;CHECK DATA FROM STACK
;BRANCH IF OK
;DATA ON STACK (067452) CHANGED
;THE ERROR NUMBER IS 451
;CHECK DATA FROM STACK
;BRANCH IF OK
;DATA ON STACK (125252) CHANGED
;THE ERROR NUMBER IS 452
;CHECK THE TEST NUMBER
;BRANCH IF OK
;WRONG TEST. PC MUST HAVE FOULED UP.
;THE ERROR NUMBER IS 453

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900
05000
05100
05200
05300
05400
05500
05600
05700
05800
05900

2793
2794
2795
2796
2797
2798
2799
2800 013124 104400
2801 013126 004567 003054
2802 013132 100125 125252
2803 013136 052525 052525
2804 013142 000047
2805 013144 013176 000113
2806 013150 012705 000510
2807
2808 013154 000240
2809 013156 075035
2810
2811 013160 004767 003054
2812 013164 010567 165244
2813 013170 104002
2814 013172 000454
2815 013174 000464
2816
2817 013176 004767 003066
2818 013202 010567 165226
2819 013206 122747 000113 165216
2820 013214 001402
2821 013216 104000
2822 013220 000455
2823
2824 013222 022767 000510 165204
2825 013230 001402
2826 013232 104000
2827 013234 000456
2828
2829 013236 022767 013160 165172
2830 013244 001402
2831 013246 104001
2832 013250 000457
2833
2834 013252 022767 000013 165160
2835 013260 001402
2836 013262 104002
2837 013264 000460
2838
2839 013266 022767 100125 165146
2840 013274 001402
2841 013276 104004
2842 013300 000461
2843
2844 013302 022767 125252 165134
2845 013310 001402
2846 013312 104004
2847 013314 000462
2848

```
*****
:TEST 62:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
:              052525,052525 / 100125,125252 ==> DIVIDE BY ZERO
:              PS(ON STACK) - 013,      STACK POINTER = R5
*****

TST62:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
        .WORD    100125,125252      ;SECOND OPERAND ON TOP
        .WORD    052525,052525      ;FIRST OPERAND ON BOTTOM
        .WORD    047                ;PROCESSOR PRIORITY LEVEL
        .WORD    ISR62, 113          ;FIS TRAP VECTOR
        MOV      #STACK0,R5         ;SET UP R5 AS STACK POINTER

        NOP
        FDIV    R5                  ;FLOATING DIVIDE ON THE R5 STACK

RT462:  JSR      PC,      POPR       ;POP THE 'ANSWER'
        MOV      R5,      $SP       ;SAVE STACK POINTER (R5)
        HLT+2    454                ;FIS TRAP DIDN'T OCCURE!
        BR      END62               ;THE ERROR NUMBER IS 454

ISR62:  JSR      PC,      POPER      ;POP ALL DATA OFF THE STACKS
        MOV      R5,      $SP       ;SAVE STACK POINTER (R5)
        CMPB    #113,      $PSW     ;CHECK PS AFTER FIS TRAP
        BEQ     .+6                 ;BRANCH IF OK
        HLT     455                 ;PS AFTER FIS TRAP NOT EQUAL TO 113
        .455                         ;THE ERROR NUMBER IS 455

        CMP     #STACK0,$SP         ;CHECK THE STACK POINTER (R5)
        BEQ     .+6                 ;BRANCH IF OK
        HLT     456                 ;STACK POINTER (R5) NOT EQUAL TO #STACK0
        .456                         ;THE ERROR NUMBER IS 456

        CMP     #RTA62, ANS1        ;CHECK FIS TRAP RETURN ADDRESS
        BEQ     .+6                 ;BRANCH IF OK
        HLT+1   457                 ;FIS TRAP AT WRONG ADDRESS
        .457                         ;THE ERROR NUMBER IS 457

        CMP     #013, ANS2          ;CHECK PS BEFORE FIS TRAP
        BEQ     .+6                 ;BRANCH IF OK
        HLT+2   460                 ;PS AT FIS TRAP TIME NOT 013
        .460                         ;THE ERROR NUMBER IS 460

        CMP     #100125,ANS3        ;CHECK DATA FROM THE STACK
        BEQ     .+6                 ;BRANCH IF OK
        HLT+4   461                 ;DATA ON STACK (100125) CHANGED
        .461                         ;THE ERROR NUMBER IS 461

        CMP     #125252,ANS4        ;CHECK DATA FROM STACK
        BEQ     .+6                 ;BRANCH IF OK
        HLT+4   462                 ;DATA ON STACK (125252) CHANGED
        .462                         ;THE ERROR NUMBER IS 462
```

```

CVKACC MACY*1 30A(1052) 21-AUG-78 15:28 PAGE 68
CVKACC.P*1 16-AUG-78 08:41 TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO
SEQ 0076

00100
00200
00300
00400 2849 013316 022767 052525 165122      CMP      #052525,ANS5 ;CHECK DATA FROM STACK
00500 2850 013324 001402      BEQ      .+6        ;BRANCH IF OK
00600 2851 013326 104006      HLT+6    463        ;DATA ON STACK (052525) CHANGED
00700 2852 013330 000463      463        ;THE ERROR NUMBER IS 463
00800 2853
00900 2854 013332 022767 052525 165110      CMP      #052525,ANS6 ;CHECK DATA FROM STACK
01000 2855 013340 001402      BEQ      .+6        ;BRANCH IF OK
01100 2856 013342 104006      HLT+6    464        ;DATA ON STACK (052525) CHANGED
01200 2857 013344 000464      464        ;THE ERROR NUMBER IS 464
01300 2858
01400 2859 013346 122767 000062 165030      END62:  CMPB     #62,  $TESTN ;CHECK THE TEST NUMBER
01500 2860 013354 001402      BEQ      .+6        ;BRANCH IF OK
01600 2861 013356 104000      HLT      465        ;WRONG TEST! PC MUST HAVE FOULED UP.
01700 2862 013360 000465      465        ;THE ERROR NUMBER IS 465
01800 2863
01900 2864

```

00300 2865
00400 2866
00500 2867
00600 2868
00700 2869
00800 2870
00900 2871
01000 2872 013362 104400
01100 2873 013364 004567 002444
01200 2874 013370 000006 123456
01300 2875 013374 100052 052525
01400 2876 013400 000357
01500 2877 013402 013430 000311
01600 2878
01700 2879 013406 000240
01800 2880 013410 075036
01900 2881
02000 2882 013412 004767 002456
02100 2883 013416 104002
02200 2884 013420 000466
02300 2885 013422 012706 000600
02400 2886 013426 000464
02500 2887
02600 2888 013430 004767 002472
02700 2889 013434 022706 000600
02800 2890 013440 001405
02900 2891 013442 012706 000600
03000 2892 013446 104000
03100 2893 013450 000467
03200 2894 013452 000452
03300 2895
03400 2896 013454 122767 000311 164750
03500 2897 013462 001402
03600 2898 013464 104000
03700 2899 013466 000470
03800 2900
03900 2901 013470 022767 013412 164740
04000 2902 013476 001402
04100 2903 013500 104001
04200 2904 013502 000471
04300 2905
04400 2906 013504 022767 000213 164726
04500 2907
04600 2908
04700 2909
04800 2910
04900 2911
05000 2912
05100 2913 013512 001402
05200 2914 013514 104002
05300 2915 013516 000472
05400 2916
05500 2917 013520 022767 000006 164714
05600 2918 013526 001402
05700 2919 013530 104004
05800 2920 013532 000473
05900

:TEST 63: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
PS(ON STACK) = 213, STACK POINTER = SP

TST63: SCOPE
JSR R5, PUSHS ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
.WORD 000006,123456 ;SECOND OPERAND ON TOP
.WORD 100052,052525 ;FIRST OPERAND ON BOTTOM
.WORD 357 ;PROCESSOR PRIORITY LEVEL
.WORD ISR63, 311 ;FIS TRAP VECTOR

NOP
FDIV SP ;FLOATING DIVIDE ON THE STACK

RTA63: JSR PC, POPS ;POP THE 'ANSWER'
HLT+2 ;FIS TRAP DIDN'T OCCURE!
466 ;THE ERROR NUMBER IS 466
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
BR END63

ISR63: JSR PC, POPES ;POP ALL DATA OFF THE STACK
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ ISA63 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE THE STACK POINTER
HLT ;STACK POINTER FOULED UP
467 ;THE ERROR NUMBER IS 467
BR END63 ;SKIP REST OF TEST

ISA63: CMPB #311, \$PSW ;CHECK PS AFTER FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER FIS TRAP NOT EQUAL TO 311
470 ;THE ERROR NUMBER IS 470

CMP #RTA63, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
471 ;THE ERROR NUMBER IS 471

CMP #213, ANS2 ;CHECK PS BEFORE FIS TRAP

BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 213
472 ;THE ERROR NUMBER IS 472

CMP #000006,ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (000006) CHANGED
473 ;THE ERROR NUMBER IS 473

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 70
00200 CVKACC.P1 16-AUG-78 08:41 TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO
00300
00400
00500 2921
00600 2922 013534 022767 123456 164702 CMP #123456,ANS4 ;CHECK DATA FROM STACK
00700 2923 013542 001402 BEQ .+6 ;BRANCH IF OK
00800 2924 013544 104004 HLT+4 ;DATA ON STACK (123456) CHANGED
00900 2925 013546 000474 474 ;THE ERROR NUMBER IS 474
01000 2926
01100 2927 013550 022767 100052 164670 CMP #100052,ANS5 ;CHECK DATA FROMONG TK
01200 2928 013556 001402 BEQ .+6 ;BRANCH IF OK
01300 2929 013560 104006 HLT+6 ;DATA ON STACK (100052) CHANGED
01400 2930 013562 000475 475 ;THE ERROR NUMBER IS 475
01500 2931
01600 2932 013564 022767 052525 164656 CMP #052525,ANS6 ;CHECK DATA FROM STACK
01700 2933 013572 001402 BEQ .+6 ;BRANCH IF OK
01800 2934 013574 104006 HLT+6 ;DATA ON STACK (052525) CHANGED
01900 2935 013576 000476 476 ;THE ERROR NUMBER IS 476
02000 2936
02100 2937 013600 122767 000063 64576 END63: CMPB #63, $TESTN ;CHECK THE TEST NUMBER
02200 2938 013606 001402 BEQ .+6 ;BRANCH IF OK
02300 2939 013610 104006 HLT ;WREST! PC MUST HAVE FOULED UP.
02400 2940 013612 000477 477 ;THE ERROR NUMBER IS 477
02500 2941
2942

```

00300 2943
00400 2944
00500 2945
00600 2946
00700 2947
00800 2948
00900 2949
01000 2950
01100 2951
01200 2952 013614 104400
01300 2953 013616 012704 000532
01400 2954 013622 012744 107070
01500 2955 013626 012744 134343
01600 2956 013632 012744 065432
01700 2957 013636 012744 032107
01800 2958 013642 012744 123456
01900 2959 013646 012744 045670
02000 2960 013652 012744 125252
02100 2961 013656 012744 135252
02200 2962 013662 012744 016161
02300 2963 013666 012744 040616
02400 2964 013672
02500 2965 013672 106427
02600 2966
02700 2967 013676 000240
02800 2968 013700 075014
02900 2969 013702 075034
03000 2970 013704 075024
03100 2971 013706 075004
03200 2972
03300 2973 013710
03400 2974 013710 106767
03500 2975 013714 042767 000020 164510
03600 2976 013722 012467 164510
03700 2977 013726 012467 164506
03800 2978 013732 010467 164476
03900 2979 013736 122767 000010 164466
04000 2980 013744 001402
04100 2981 013746 104000
04200 2982 013750 000500
04300 2983
04400 2984 013752 022767 000532 164454
04500 2985 013760 001402
04600 2986 013762 104000
04700 2987 013764 000501
04800 2988
04900 2989
05000 2990 013766 022767 137201 164442
05100 2991 013774 001402
05200 2992 013776 104002
05300 2993 014000 000502
05400 2994
05500 2995 014002 022767 115230 164430
05600 2996 014010 001402
05700 2997 014012 104002
05800 2998 014014 000503
05900

```

*****
:TEST 64:      TEST ALL INSTRUCTION TOGETHER
:              032107,065432 * 045670,123456
:              134343,107070 + ----- - 137201,115230
:              (135252,125252 - 040616,016161)
:              PS=010, STACK POINTER=R4
*****

```

```

TST64:  SCOPE
MOV     #STAK10,R4      ;SET STACK POINTER
MOV     #107070,-(R4)   ;LOAD DATA ONTO STACK
MOV     #134343,-(R4)
MOV     #065432,-(R4)
MOV     #032107,-(R4)
MOV     #123456,-(R4)
MOV     #045670,-(R4)
MOV     #125252,-(R4)
MOV     #135252,-(R4)
MOV     #016161,-(R4)
MOV     #040616,-(R4)
MTPS   #144            ;SET PROCESSOR STATUS
.WORD  '06400...C

NOP
FSUB   R4              ;135252,125252-040616,016161=140616,017434
FDIV   R4              ;045670,123456/140616,017434=145246,047065
FMUL   R4              ;032107,065432*145246,047065=137201,106137
FADD   R4              ;134343,107070+137201,106137-137201,115230

MFPS   $PSW           ;SAVE FINAL PS
.WORD  106700...C
BIC    #20,$PSW       ;CLR T-BIT
MOV    (R4)+,ANS1     ;SAVE FIRST HALF OF ANSWER
MOV    (R4)+,ANS2     ;SAVE SECOND HALF OF ANSWER
MOV    R4,$SSP        ;SAVE STACK POINTER
CMPB   #010,$PSW     ;CHECK PS (EXCEPT T BIT)
BEQ    .+6            ;BRANCH IF OK
HLT    .               ;PS NOT EQUAL TO 010
500    .               ;THE ERROR NUMBER IS 500

CMP    #STAK10,$SSP  ;CHECK THE STACK POINTER (R4)
BEQ    .+6            ;BRANCH IF OK
HLT    .               ;STACK POINTER (R4) NOT EQUAL TO THE
501    .               ;THE ERROR NUMBER IS 501
                    ;ADDRESS OF STAK10

CMP    #137201,ANS1  ;CHECK FIRST HALF OF ANSWER
BEQ    .+6            ;BRANCH IF OK
HLT+2  .               ;ANS1 NOT EQUAL TO 137201
502    .               ;THE ERROR NUMBER IS 502

CMP    #115230,ANS2  ;CHECK SECOND HALF OF ANSWER
BEQ    .+6            ;BRANCH IF OK
HLT+2  .               ;ANS2 NOT EQUAL TO 115230
503    .               ;THE ERROR NUMBER IS 503

```


00100
00200
00300
00400
00500
00600
00700
00800
00900

CVKACC MACY11 30A(1052) 21-AUG-78 15:28
CVK4CC.P11 16-AUG-78 08:41

PAGE 72
TEST OF ALL FIS AT ONCE

SEQ 0080

2999
3000 014016 122767 000064 164360
3001 014024 001402
3002 014026 104000
3003 014030 000504
3004

END64: CMPB #64.
BEQ .+6
HLT
504

\$TESTN :CHECK THE TEST NUMBER
:BRANCH IF OK
:WRONG TEST! PC MUST HAVE FOULED UP.
:THE ERROR NUMBER IS 504

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200
04300
04400
04500
04600
04700
04800
04900
05000
05100
05200
05300
05400
05500
05600
05700
05800
05900

CVKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 73
CVKACC.P11 16-AUG-78 08:41
3005
3006
3007
3008
3009
3010
3011 014032 104400
3012 014034 012737 014124 000004
3013 014042 012737 000340 000006
3014 014050 004567 002132
3015 014054 070707 016161
3016 014060 146314 143434
3017 014064 000143
3018 014066 016456 000340
3019 014072
3020 014072 106427
3021 014076 012702 177777
3022
3023 014102 000240
3024 014104 075002
3025
3026 014106 004767 002126
3027 014112 010267 164316
3028 014116 104002
3029 014120 000505
3030 014122 000464
3031
3032 014124 004767 002140
3033 014130 010267 16300
3034 014134 122767 000340 164270
3035 014142 001402
3036 014144 104000
3037 014146 000506
3038
3039 014150 022767 177777 164256
3040 014156 001402
3041 014160 104000
3042 014162 000507
3043
3044 014164 022767 014106 164244
3045 014172 001402
3046 014174 104001
3047 014176 000510
3048
3049 014200 022767 000151 164232
3050 014206 001402
3051 014210 104002
3052 014212 000511
3053
3054 014214 022767 070707 164220
3055 014222 001402
3056 014224 104004
3057 014226 000512
3058
3059 014230 022767 016161 164206
3060 014236 001402

ADDRESS ERROR TEST

SEQ 0081

```

:*****
:TEST 65: TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
:INSTRUCTION = FADD, STACK POINTER = R2
:*****
TST65: SCOPE
MOV #ISR65, @#4 ;SET UP ADDRESS TRAP VECTOR
MOV #340, @#6
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
.WORD 070707,016161 ;SECOND OPERAND ON TOP
.WORD 146314,143434 ;FIRST OPERAND ON BOTTOM
.WORD 143 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MTPS #143 ;SET PROCESSOR STATUS
.WORD 106400, .C
MOV #177777, R2 ;SET UP R2 AS STACK POINTER

NOP
FADD R2 ;FLOATING ADD ON THE R2 STACK

RTA65: JSR PC, POPR ;POP THE 'ANSWER'
MOV R2, $SP ;SAVE STACK POINTER (R2)
HLT+2 ;FIS TRAP DIDN'T OCCURE.
505 ;THE ERROR NUMBER IS 505
BR END65

ISR65: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
MOV R2, $SP ;SAVE STACK POINTER (R2)
CMPB #340, $PSW ;CHECK PS AFTER ADR. ERR. TRAP
BEQ .+6 ;BRANCH IF OK
HLT ;PS AFTER TRAP NOT EQUAL TO 340
506 ;THE ERROR NUMBER IS 506

CMP #177777, $SP ;CHECK THE STACK POINTER (R2)
BEQ .+6 ;BRANCH IF OK
HLT ;STACK POINTER (R2) NOT EQUAL TO #177777
507 ;THE ERROR NUMBER IS 507

CMP #RTA65, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 ;FIS TRAP AT WRONG ADDRESS
510 ;THE ERROR NUMBER IS 510

CMP #151, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 ;PS AT FIS TRAP TIME NOT 151
511 ;THE ERROR NUMBER IS 511

CMP #070707, ANS3 ;CHECK DATA FROM THE STACK
BEQ .+6 ;BRANCH IF OK
HLT+4 ;DATA ON STACK (070707) CHANGED
512 ;THE ERROR NUMBER IS 512

CMP #016161, ANS4 ;CHECK DATA FROM STACK
BEQ .+6 ;BRANCH IF OK

```

:TEST 66: TEST THAT STACK POINTER ADDRESS ERROR CAUSES ABORT
: INSTRUCTION - FMUL, STACK POINTER = R5

TST66: SCOPE
MOV #ISR66, @#4 ;SET UP ADDRESS TRAP VECTOR
MOV #340, @#6
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R5 STACK, SET PRIORITY
.WORD 065432,123456 ;SECOND OPERAND ON TOP
.WORD 037654,032107 ;FIRST OPERAND ON BOTTOM
.WORD 202 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MTPS #202 ;SET PROCESSOR STATUS
.WORD 106400...C
MOV #160000,R5 ;SET UP R5 AS STACK POINTER

NOP
FMUL R5 ;FLOATING MULTIPLY ON THE R5 STACK

RTA66: MFPS \$PSW ;SAVE THE PSW
.WORD 106700...C
MOV R5, \$SP ;SAVE STACK POINTER (R5)
HLT 517 ;FIS TRAP DIDN'T OCCURE
BR ;THE ERROR NUMBER IS 517

ISR66: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
MOV R5, \$SP ;SAVE STACK POINTER (R5)
CMPB #340, \$PSW ;CHECK PS AFTER ADR. ERR. TRAP
BEQ .+6 ;BRANCH IF OK
HLT 520 ;PS AFTER TRAP NOT EQUAL TO 340
 ;THE ERROR NUMBER IS 520

CMP #160000,\$SP ;CHECK THE STACK POINTER (R5)
BEQ .+6 ;BRANCH IF OK
HLT 521 ;STACK POINTER (R5) NOT EQUAL TO #160000
 ;THE ERROR NUMBER IS 521

CMP #RTA66, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
BEQ .+6 ;BRANCH IF OK
HLT+1 522 ;FIS TRAP AT WRONG ADDRESS
 ;THE ERROR NUMBER IS 522

CMP #210, ANS2 ;CHECK PS BEFORE FIS TRAP
BEQ .+6 ;BRANCH IF OK
HLT+2 523 ;PS AT FIS TRAP TIME NOT 210
 ;THE ERROR NUMBER IS 523

END66: CMPB #66, \$TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT 524 ;WRONG TEST! PC MUST HAVE FOULED UP.
 ;THE ERROR NUMBER IS 524

00300 3077
00400 3078
00500 3079
00600 3080
00700 3081
00800 3082
00900 3083
01000 3083 014310 104400
01100 3084 014312 012737 014402 000004
01200 3085 014320 012737 000340 000006
01300 3086 014326 004567 001654
01400 3087 014332 065432 123456
01500 3088 014336 037654 032107
01600 3089 014342 000202
01700 3090 014344 016456 000340
01800 3091 014350
01900 3092 014350 106427
02000 3093 014354 012705 160000
02100 3094
02200 3095 014360 000240
02300 3096 014362 075025
02400 3097
02500 3098 014364
02600 3099 014364 106767
02700 3100 014370 010567 164040
02800 3101 014374 104000
02900 3102 014376 000517
03000 3103 014400 000434
03100 3104
03200 3105 014402 004767 001662
03300 3106 014406 010567 164022
03400 3107 014412 122767 000340 164012
03500 3108 014420 001402
03600 3109 014422 104000
03700 3110 014424 000520
03800 3111
03900 3112 014426 022767 160000 164000
04000 3113 014434 001402
04100 3114 014436 104000
04200 3115 014440 000521
04300 3116
04400 3117 014442 022767 014364 163766
04500 3118 014450 001402
04600 3119 014452 104001
04700 3120 014454 000522
04800 3121
04900 3122 014456 022767 000210 163754
05000 3123 014464 001402
05100 3124 014466 104002
05200 3125 014470 000523
05300 3126
05400 3127 014472 122767 000066 163704
05500 3128 014500 001402
05600 3129 014502 104000
05700 3130 014504 000524

```

00100
00200
00300
00400 3131 014506 012737 000006 000004 52200
00500 3132 014514 005037 000006 52300
00600 3133 014520 012767 000003 164010 52700
00700 3134
00800 3135
00900 3136
01000 3137
01100 3138
01200 3139
01300 3140
01400 3141 014526 104400
01500 3142 014530 132737 000040 000421
01600 3143 014536 001177
01700 3144 014540 013704 000546
01800 3145 014544 012724 014646
01900 3146 014550 012714 000340
02000 3147 014554 032737 004000 000422
02100 3148 014562 001005
02200 3149 014564 000004 000473
02300 3150
02400 3151 014570 012767 014576 163672
02500 3152 014576 004567 001404
02600 3153 014602 000177 134543
02700 3154 014606 035700 143235
02800 3155 014612 000143
02900 3156 014614 016456 000340
03000 3157 014620 012701 000510
03100 3158 014624 012767 000060 163702
03200 3159 014632 112777 000100 163710
03300 3160
03400 3161 014640 075001
03500 3162 014642 024141
03600 3163 014644 000775
03700 3164
03800 3165 014646 105077 163676
03900 3166 014652 022716 014640
04000 3167 014656 001424
04100 3168 014660 022766 014640 000004
04200 3169 014666 001423
04300 3170 014670 005337 000534
04400 3171 014674 001517
04500 3172 014676 112777 000015 163646
04600 3173 014704 105777 163640
04700 3174 014710 100375
04800 3175 014712 112777 000015 163632
04900 3176 014720 012777 000100 163622
05000 3177 014726 000002
05100 3178
05200 3179 014730 004767 001334
05300 3180 014734 000403
05400 3181
05500 3182 014736 022626
05600 3183 014740 004767 001330
05700 3184 014744 005746
05800 3185 014746 012746 014640
05900 3186 014752 022706 000574

```

```

:*****
:TEST 67: TEST THAT FIS ABORTS PROPERLY WHEN INTERRUPTED
: 035700,143235 + 000177,134543 = 035700,143235
: PS - .PS, STACK POINTER = R1
:*****

TST67: SCOPE
BITB #40,@#SENVM
BNE END67+2 ;EXIT THIS TEST IF BIT 5 OF SENVM IS HIGH
MOV @#TTYOUT,R4
MOV #ISR67,(R4)+ ;SET UP TELEPRINTER INTERRUPT VECTOR
MOV #340,(R4)
BIT #SW11,@#SSWREG ;TEST FOR ITERATIONS
BNE 1$ ;BRANCH TO AVOID HANG UP
TYPE, RETURN+1 ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS 0
;TO TYPE CARRIAGE RETURN, LINE FEED

1$: MOV #.+6, LAD$ ;RESET LOOP ADDRESS
JSR R5, PUSHR ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
.WORD 000177,134543 ;SECOND OPERAND ON TOP
.WORD 035700,143235 ;FIRST OPERAND ON BOTTOM
.WORD 143 ;PROCESSOR PRIORITY LEVEL
.WORD TRAPER, 340 ;FIS TRAP VECTOR
MOV #STACK0,R1 ;SET UP STACK POINTER
MOV #60,TEMP ;INITIALIZE COUNTER FOR FIS INTERRUPTS
MOVB #100,@#STPS ;SET TTY INTERRUPT ENABLE

RTA67: FADD R1 ;FLOATING ADD ON THE STACK
CMP -(R1), -(R1) ;RESET THE STACK POINTER FOR NEXT PASS
BR RTA67 ;REPEAT UNTIL INTERRUPTED

ISR67: CLR @#STPS ;CLEAR THE INTERRUPT ENABLE
CMP #RTA67,(SP) ;CHECK IF INTERRUPT AT FIS INSTR.
BEQ 3$ ;BRANCH IF IT DID
CMP #RTA67,4(SP) ;CHECK FOR INTERRUPT WITH T-BIT SET
BEQ 4$ ;BRANCH IF IT DID
DEC @#TEMP ;DID INTERRUPTS OCCUR TEMP # OF TIMES
BEQ END67 ;IF YES THEN SKIP TO END OF THIS TEST
1$: MOVB #15,@#TPB ;CONTINUE TO TYPE 'CR'
2$: TST @#TPS ;LOOP HERE UNTILL DONE BIT COMES ON
BPL 2$
MOVB #15,@#TPB ;TYPE ANOTHER 'CR'
MOV #100,@#STPS ;SET TTY INTERRUPT ENABLE
RTI

3$: JSR PC, POPER ;SAVE ALL THE STUFF ON THE STACK
BR 5$

4$: CMP (SP)+,(SP)+ ;RESET THE STACK TO IGNORE THE TRACE TRAP
JSR PC, POPER1 ;POP ALL THE STUFF OFF THE STACK
5$: TST -(SP) ;SAVE PSW FOR FUTURE RTI
MCL #RTA67,-(SP) ;PLACE THE RTI ADDRESS BACK IN SP
CMP #BEGIN-4,SP ;CHECK THE STACK POINTER

```

00100	3187	014756	001407			BEQ	6\$:BRANCH IF OK
00200	3188	014760	010667	163450		MOV	SP, \$SP		:SAVE FOR TYPING
00300	3189	014764	012706	000574		MOV	#BEGIN-4,SP		:RESTORE THE STACK POINTER
00400	3190	014770	104000			HLT			:STACK POINTER FOULED UP
00500	3191	014772	000525			525			:THE ERROR NUMBER IS 525
00600	3192	014774	000457			BR	END67		:SKIP REST OF TEST
00700	3193								
00800	3194	014776	010167	163432		MOV	R1, \$SP		:SAVE STACK POINTER
00900	3195	015002	122767	000344	163422	CMPB	#344, \$PSW		:CHECK PS AFTER INTERUPT
01000	3196	015010	001402			BEQ	+.6		:BRANCH IF OK
01100	3197	015012	104000			HLT			:PS AFTER INTERUPT NOT EQUAL TO LVLA
01200	3198	015014	000526			526			:THE ERROR NUMBER IS 526
01300	3199								
01400	3200	015016	022767	000510	163410	CMP	#STACK0,\$SP		:CHECK THE STACK POINTER (R1)
01500	3201	015024	001402			BEQ	+.6		:BRANCH IF OK
01600	3202	015026	104000			HLT			:STACK POINTER (R1) NOT EQUAL TO #STACK0
01700	3203	015030	000527			527			:THE ERROR NUMBER IS 527
01800	3204								
01900	3205	015032	022767	014640	163376	CMP	#RTA67,ANS1		:CHECK FIS TRAP RETURN ADDRESS
02000	3206	015040	001402			BEQ	+.6		:BRANCH IF OK
02100	3207	015042	104001			HLT+1			:FIS TRAP AT WRONG ADDRESS
02200	3208	015044	000530			530			:THE ERROR NUMBER IS 530
02300	3209								
02400	3210								
02500	3211	015046	022767	000177	163366	CMP	#000177,ANS3		:CHECK DATA FROM THE STACK
02600	3212	015054	001402			BEQ	+.6		:BRANCH IF OK
02700	3213	015056	104004			HLT+4			:DATA ON STACK (000177) CHANGED
02800	3214	015060	000531			531			:THE ERROR NUMBER IS 531
02900	3215								
03000	3216	015062	022767	134543	163354	CMP	#134543,ANS4		:CHECK DATA FROM STACK
03100	3217	015070	001402			BEQ	+.6		:BRANCH IF OK
03200	3218	015072	104004			HLT+4			:DATA ON STACK (134543) CHANGED
03300	3219	015074	000532			532			:THE ERROR NUMBER IS 532
03400	3220								
03500	3221	015076	022767	035700	163342	CMP	#035700,ANS5		:CHECK DATA FROM STACK
03600	3222	015104	001402			BEQ	+.6		:BRANCH IF OK
03700	3223	015106	104006			HLT+6			:DATA ON STACK (035700) CHANGED
03800	3224	015110	000533			533			:THE ERROR NUMBER IS 533
03900	3225								
04000	3226	015112	022767	143235	163330	CMP	#143235,ANS6		:CHECK DATA FROM STACK
04100	3227	015120	001402			BEQ	+.6		:BRANCH IF OK
04200	3228	015122	104006			HLT+6			:DATA ON STACK (143235) CHANGED
04300	3229	015124	000534			534			:THE ERROR NUMBER IS 534
04400	3230								
04500	3231	015126	005367	163402		DEC	TEMP		:STAY IN THE LOOP FOR 30 TIMES
04600	3232	015132	001261			BNE	1\$		
04700	3233								
04800	3234	015134	022626						
04900	3235	015136	122767	000067	163240	CMP	(SP)+, (SP)+		:RESTORE STACK POINTER TO 500
05000	3236	015144	001402			CMPB	#67, \$TESTN		:CHECK THE TEST NUMBER
05100	3237	015146	104000			BEQ	+.6		:BRANCH IF OK
05200	3238	015150	000535			HLT			:WRONG TEST. PC MUST HAVE FOULED UP.
05300	3239	015152				535			:THE ERROR NUMBER IS 535
05400	3240	015152	106427			MTPS	#340		
05500	3240					.WORD	*06400...		
05600									
05700									
05800									

END67:

```

00100
00200
00300
00400 3242
00500 3243
00600 3244
00700 3245
00800 3246
00900 3247
01000 3248
01100 3249 015156 104400
01200 3250 015160 132737 000040 000421
01300 3251 015166 001176
01400 3252 015170 013704 000546
01500 3253 015174 012724 015276
01600 3254 015200 012714 000340
01700 3255 015204 032737 004000 000422
01800 3256 015212 001005
01900 3257 015214 000004 000473
02000 3258
02100 3259 015220 012767 015226 163242
02200 3260 015226 004567 000754
02300 3261 015232 040200 000000
02400 3262 015236 107070 070707
02500 3263 015242 000100
02600 3264 015244 016456 000340
02700 3265 015250 012700 000510
02800 3266 015254 012767 000060 163252
02900 3267 015262 112777 000100 163260
03000 3268
03100 3269 015270 075020
03200 3270 015272 024040
03300 3271 015274 000775
03400 3272
03500 3273 015276 105077 163246
03600 3274 015302 022716 015270
03700 3275 015306 001424
03800 3276 015310 022766 015270 000004
03900 3277 015316 001423
04000 3278 015320 005337 000534
04100 3279 015324 001516
04200 3280 015326 112777 000015 163216
04300 3281 015334 105777 163210
04400 3282 015340 100375
04500 3283 015342 112777 000015 163202
04600 3284 015350 012777 000100 163172
04700 3285 015356 000002
04800 3286
04900 3287 015360 004767 000704
05000 3288 015364 000403
05100 3289
05200 3290 015366 022626
05300 3291 015370 004767 000700
05400 3292 015374 005746
05500 3293 015376 012746 015270
05600 3294 015402 022706 000574
05700 3295 015406 001407
05800 3296 015410 010667 163020
05900 3297 015414 012706 000574

```

```

:*****
:TEST 70: TEST THAT FIS ABORTS PROPERLY WHEN INTERRUPTED
:          107070,070707 * 040200,000000 107070,070707
:          PS - .PS, STACK POINTER R0
:*****
TST70: SCOPE
      BITB #40,@$SENV#M ;EXIT THIS TEST IF BIT 5 OF $SENV#M IS HIGH
      BNE END70+2
      MOV @TTYOUT,R4 ;SET UP TELEPRINTER INTERRUPT VECTOR
      MOV #ISR70,(R4)+
      MOV #340,(R4)
      BIT #SW11,@$SWREG ;TEST FOR ITERATIONS
      BNE 1$ ;BRANCH TO AVOID HANG UP
      TYPE, RETURN+1 ;RETURN+1 CAN BE REPLACED WITH THE ADDRESS 0
                          ;TO TYPE CARRIAGE RETURN, LINE FEED
                          ;RESET LOOP ADDRESS
1$: MOV #.+6,LAD$ ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
      JSR R5,PUSHR ;SECOND OPERAND ON TOP
      .WORD 040200,000000 ;FIRST OPERAND ON BOTTOM
      .WORD 107070,070707 ;PROCESSOR PRIORITY LEVEL
      .WORD 100 ;FIS TRAP VECTOR
      .WORD TRAPER,340 ;SET UP STACK POINTER
      MOV #STACK0,R0 ;INITIALIZE COUNTER FOR FIS INTERRUPTS
      MOV #60,TEMP ;SET TTY INTERRUPT ENABLE
      MOVB #100,@$STPS
RTA70: FMUL R0 ;FLOATING MULTIPLY ON THE STACK
      CMP -(R0),-(R0) ;RESET THE STACK POINTER FOR NEXT PASS
      BR RTA70 ;REPEAT UNTIL INTERRUPTED
ISR70: CLRB @$STPS ;CLEAR THE INTERRUPT ENABLE
      CMP #RTA70,(SP) ;CHECK IF INTERRUPT AT FIS INSTR.
      BEQ 3$ ;BRANCH IF IT DID
      CMP #RTA70,4(SP) ;CHECK FOR INTERRUPT WITH T-BIT SET
      BEQ 4$ ;BRANCH IF IT DID
      DEC @TEMP ;DID INTERRUPTS OCCUR TEMP # OF TIMES
      BEQ END70 ;IF YES THEN SKIP TO END OF THIS TEST
1$: MOVB #15,@$STPB ;CONTINUE TO TYPE 'CR'
2$: TSTB @$STPS ;LOOP HERE UNTILL DONE BIT COMES ON
      BPL 2$
      MOVB #15,@$STPB ;TYPE ANOTHER 'CR'
      MCV #100,@$STPS ;SET TTY INTERRUPT ENABLE
3$: JSR PC,POPER ;SAVE ALL THE STUFF ON THE STACK
      BR 5$
4$: CMP (SP)+,(SP)+ ;RESET THE STACK TO IGNORE THE TRACE TRAP
      JSR PC,POPER1 ;POP ALL THE STUFF OFF THE STACK
5$: TST -(SP) ;SAVE PSW FOR FUTURE RTI
      MOV #RTA70,-(SP) ;PLACE THE RTI ADDRESS BACK IN SP
      CMP #BEGIN-4,SP ;CHECK THE STACK POINTER
      BEQ 6$ ;BRANCH IF OK
      MOV SP,$SP ;SAVE FOR TYPING
      MOV #BEGIN-4,SP ;RESTORE THE STACK POINTER

```


00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 80
00200 CVKACC.P11 16-AUG-78 08:41 INTERRUPT ABORT TEST SECTION

SEQ 0088

00300									
00400	3350	015604	012767	000377	162724	53000	MOV	#377,	TIMES ;SET NUMBER OF ITERATIONS TO 377
00500	3351	015612	010477	162730		53100	MOV	R4,	@TTYOUT ;RESTORE TTY INTERRUPT VECTOR
00600	3352	015616	005014			53200	(LR	(R4)	
00700	3353					53300			

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000
03100
03200
03300
03400
03500
03600
03700
03800
03900
04000
04100
04200

CVKACC MARY11 30A(1052) 21-AUG-78 15:28 PAGE 81
CVKACC.P11 16-AUG-78 08:41
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365 015620
3366 015620 104400
3367 015622 005267 162560
3368 015626 042767 100000 162552
3369 015634 005327
3370 015636 000001
3371 015640 003015
3372 015642 012737
3373 015644 000001
3374 015646 015636
3375 015650 000004 015700
3376 015654
3377
3378 015654 013700 000042
3379 015660 001405
3380 015662 000005
3381 015664 004710
3382 015666 000240
3383 015670 000240
3384 015672 000240
3385 015674
3386 015674 000137 000600
3387 015700 005015 047105 020104
3388 015706 040520 051523
3389 015712 377 377 000
3390 015716
3391
3392 015644 000001 53600 54100

INTERUPT ABORT TEST SECTION

SEQ 0089

```
*****
.SBTTL END OF PASS ROUTINE
;*INCREMENT THE PASS NUMBER ($PASS)
;*TYPE 'END PASS'
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO BEGIN
;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE 'END OF PASS' LOCATION
;*$SENDMG CAN BE CHANGED TO 7.

$EOP:
SCOPE
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,@(PC)+ ;;RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT
TYPE , $SENDMG ;;TYPE 'END PASS'

$GET42:
MOV @#42,R0 ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11

$DOAGN: JMP @#BEGIN ;;RETURN
$SENDMG: .ASCII <15><12>/END PASS/

$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
.EVEN

ENDCT: :
```

00100
00200
00300
00400
00500
00600
00700
00800
00900
01000
01100
01200
01300
01400
01500
01600
01700
01800
01900
02000
02100
02200
02300
02400
02500
02600
02700
02800
02900
03000

CVKACC MARY11 30A(1052) 21-AUG-78 15:28
CVKACC.P11 16-AUG-78 08:41

3393					
3394					
3395					
3396					
3397	015716	032737	000400	000422	
3398	015724	001404			
3399	015726	123767	000422	162450	
3400	015734	001431			
3401	015736	032737	040000	000422	
3402	015744	001023			
3403	015746	032737	004000	000422	
3404	015754	001412			
3405	015756	105767	162521		
3406	015762	001404			
3407	015764	126767	162546	162511	
3408	015772	001010			
3409	015774	112767	000001	62501	
3410	016002	105267	162376		
3411	016006	011667	162456		
3412	016012	000002			
3413					
3414	016014	105267	162463		
3415	016020	005767	162444		
3416	016024	001766			
3417	016026	016716	162436		
3418	016032	000002			
3419					

END OF PASS ROUTINE

SEQ JUNE

.SBTTL SCOPE ROUTINE

SCOPE\$: BIT #SW08,@#SSWREG ;KILL LDUB OR LOOP ON SPEC. TEST
BEQ 1\$
CMPB @#SSWREG,\$TESTN ;ON RIGHT TEST? *SW7-0*
BEQ OVER\$
1\$: BIT #SW14,@#SSWREG ;LOOP ON TEST
BNE KITS
BIT #SW11,@#SSWREG ;KILL ITERATIONS
BEQ SVLAD\$
TSTB \$ICNT
BEQ 2\$;BRANCH IF FIRST
CMPB TIMES,\$ICNT ;DONE?
BNE KITS ;BRANCH IF NOT
2\$: MOVB #1,\$ICNT ;FIRST ITERATION
SVLAD\$: INCB \$TESTN ;COUNT TEST NUMBERS
MOV (6),LAD\$;SAVE LOOP ADDRESS
RTI ;RETURN

KITS: INCB \$ICNT
OVER\$: TST LAD\$;FIRST ONE?
BEQ SVLAD\$
MOV LAD\$,(6) ;FUDGE RETURN ADDRESS
RTI ;FIXES PS

```

00300
00400 3420 55000
00500 3421 55100
00600 3422 55200
00700 3423 016034 005726 55300
00800 3424 016036 062705 000010 55400
00900 3425 016042 014546 55500
01000 3426 016044 014546 55600
01100 3427 016046 014546 55700
01200 3428 016050 014546 55800
01300 3429 016052 062705 000010 55900
01400 3430 016056 56300
01500 3431 016056 106425
01600 3432 016060 005205 56700
01700 3433 016062 012577 162376 56800
01800 3434 016066 012577 162374 56900
01900 3435 016072 000115 57000
02000 3436 57100
02100 3437 57200
02200 3438 57300
02300 3439 57400
02400 3440 57500
02500 3441 016074 57900
02600 3442 016074 106767
02700 3443 016100 042767 000020 162324 58300
02800 3444 016106 012604 58400
02900 3445 016110 012667 162322 58500
03000 3446 016114 012667 162320 58600
03100 3447 016120 010667 162310 58700
03200 3448 016124 000114 58800
03300 3449 58900
03400 3450 59000
03500 3451 59100
03600 3452 59200
03700 3453 59300
03800 3454 59400
03900 3455 59500
04000 3456 016126 59900
04100 3457 016126 106767
04200 3458 016132 012604 60300
04300 3459 016134 012667 162276 60400
04400 3460 016140 011667 162274 60500
04500 3461 016144 042767 000020 162266 60600
04600 3462 016152 012746 016160 60700
04700 3463 016156 000002 60800
04800 3464 016160 012667 162256 60900
04900 3465 016164 012667 162254 61000
05000 3466 016170 012667 162252 61100
05100 3467 016174 012667 162250 61200
05200 3468 016200 010667 162230 61300
05300 3469 016204 000114 61400
05400 3470 61500
05500 3471 61600
05600 3472 61700
05700 3473 016206 012704 000510 61800
05800 3474 016212 012524 61900
05900 3475 016214 012524 62000

```

```

;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
PUSHS: TST (SP)+ ;POP STACK BY 1
        ADD #10, R5 ;POINT TO END OF DATA
        MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
        MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
        MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
        MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
        ADD #10, R5 ;POINT TO END OF DATA
        MTPS (R5)+ ;SET THE PROCESSOR STATUS
        .WORD 106400...C
        INC R5
        MOV (R5)+, @FISVEC ;SET UP FIS ERROR TRAP VECTOR
        MOV (R5)+, @FISLVL ;TRAP STATUS
        JMP (R5) ;RETURN

```

```

;SUBROUTINE TO POP 2 WORDS OFF THE STACK
;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
POPS: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
       .WORD 106700...C
       BIC #20, $PSW ;CLEAR T-BIT
       MOV (SP)+, R4 ;SAVE RTS ADDRESS
       MOV (SP)+, ANS1 ;SAVE THE ANSWER
       MOV (SP)+, ANS2
       MOV SP, $SP ;SAVE THE STACK POINTER
       JMP (R4) ;RETURN

```

```

;SUBROUTINE TO POP 6 WORDS OFF THE STACK.
;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
;ALSO SAVES THE PS AND STACK POINTER.
POPES: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
       .WORD 106700...C
       MOV (SP)+, R4 ;SAVE RTS ADDRESS
       MOV (SP)+, ANS1 ;SAVE RTI ADDRESS
       MOV (SP)+, ANS2 ;SAVE RTI STATUS
       BIC #20, ANS2 ;CLEAR THE T-BIT
       MOV #1$, -(SP)
       RTI ;RESTORE THE PROCESSOR STATUS
       1$: MOV (SP)+, ANS3 ;SAVE DATA
          MOV (SP)+, ANS4
          MOV (SP)+, ANS5
          MOV (SP)+, ANS6
          MOV SP, $SP ;SAVE SP
          JMP (R4) ;RTS

```

```

;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
PUSHR: MOV #STACK0, R4 ;SET R4 TO STACK
        MOV (R5)+, (R4)+ ;PUT DATA ON STACK
        MOV (R5)+, (R4)+

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 84
 00200 CVKACC.P11 16-AUG-78 08:41 PUSH AND POP SUBROUTINES

SEQ 0092

```

00300
00400 3476 016216 012524 62100 MOV (R5)+, (R4)+ ;
00500 3477 016220 012524 62200 MOV (R5)+, (R4)+ ;
00600 3478 016222 62600 MTPS (R5)+ ;SET THE PROCESSOR STATUS
00700 3479 016222 106425 .WORD 106400!..C
00800 3480 016224 005205 63000 INC R5
00900 3481 016226 012577 162232 63100 MOV (R5)+, @FISVEC ;SET UP FIS ERROR TRAP VECTOR
01000 3482 016232 012577 162230 63200 MOV (R5)+, @FISLVL ;TRAP STATUS
01100 3483 016236 000205 63300 RTS R5 ;RETURN
01200 3484 63400
01300 3485 63500
01400 3486 63600 ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
01500 3487 63700 ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
01600 3488 63800
01700 3489 016240 64200 POPR: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
01800 3490 016240 106767 .WORD 106700!..C
01900 3491 016244 042767 000020 162160 64600 BIC #20, $PSW ;CLEAR T-BIT
02000 3492 016252 016767 162236 62156 64700 MOV STACK4, ANS1 ;SAVE THE ANSWER
02100 3493 016260 016767 162232 162152 64800 MOV STACK6, ANS2 ;
02200 3494 016266 000207 64900 RTS PC
02300 3495 65000
02400 3496 65100
02500 3497 65200 ;SUBROUTINE TO POP 6 WORDS OFF THE STACKS.
02600 3498 65300 ;THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP.
02700 3499 65400 ;THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
02800 3500 65500 ;ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
02900 3501 65600
03000 3502 016270 66000 POPER: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
03100 3503 016270 106767 .WORD 106700!..C
03200 3504 016274 012667 000056 66400 POPER1: MOV (SP)+, SAVRTS ;SAVE RTS ADDRESS
03300 3505 016300 012667 162132 66500 MOV (SP)+, ANS1 ;SAVE RTI ADDRESS
03400 3506 016304 011667 162130 66600 MOV (SP), ANS2 ;SAVE RTI STATUS
03500 3507 016310 042767 000020 162122 66700 BIC #20, ANS2 ;CLEAR THE T-BIT
03600 3508 016316 012746 016324 66800 MOV #1$, -(SP)
03700 3509 016322 000002 66900 RTI ;RESTORE PROCESSOR STATUS
03800 3510 016324 016767 162160 162110 67000 1$: MOV STACK0, ANS3 ;SAVE DATA
03900 3511 016332 016767 162154 162104 67100 MOV STACK2, ANS4 ;
04000 3512 016340 016767 162150 162100 67200 MOV STACK4, ANS5 ;
04100 3513 016346 016767 162144 162074 67300 MOV STACK6, ANS6 ;
04200 3514 016354 000137 67400 JMP @PC)+ ;SIMULATED RTS
04300 3515 016356 000000 67500 SAVRTS: 0
04400 3516 67600
04500 3517 67700 ;SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
04600 3518 67800
04700 3519 016360 012504 67900 PUSH7: MOV (R5)+, R4 ;SET R4 TO STACK
04800 3520 016362 012524 68000 MOV (R5)+, (R4)+ ;PUT DATA ON STACK
04900 3521 016364 012524 68100 MOV (R5)+, (R4)+ ;
05000 3522 016366 012524 68200 MOV (R5)+, (R4)+ ;
05100 3523 016370 012524 68300 MOV (R5)+, (R4)+ ;
05200 3524 016372 68700 MTPS (R5)+ ;SET THE PROCESSOR STATUS
05300 3525 016372 106425 .WORD 106400!..C
05400 3526 016374 005205 69100 INC R5
05500 3527 016376 012577 162062 69200 MOV (R5)+, @FISVEC ;SET UP FIS ERROR TRAP VECTOR
05600 3528 016402 012577 162060 69300 MOV (R5)+, @FISLVL ;TRAP STATUS
05700 3529 016406 000205 69400 RTS R5 ;RETURN
05800 3530 69500
05900 3531 69600 ;SUBROUTINE TO POP 4 WORDS OFF THE PC "STACK"

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 85
00200 CVKACC.P11 16-AUG-78 08:41 PUSH AND POP SUBROUTINES

SEQ 0093

```

00300
00400 3532 69700 ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
00500 3533 69800
00600 3534 016410 70200 POP7: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
00700 3535 016410 106767 .WORD 106700...C
00800 3536 016414 042767 000020 162010 70600 BIC #20, $PSW ;CLEAR T-BIT
00900 3537 016422 011600 MOV (SP), R0 ;GET RETURN ADDRESS
01000 3538 016424 162700 000014 70800 SUB #14, R0 ;POINT TO TOP OF 'PC STACK'
01100 3539 016430 012067 162002 70900 MOV (R0)+, ANS1 ;SAVE 1ST HALF INPUT DATA
01200 3540 016434 012067 162000 71000 MOV (R0)+, ANS2 ;SAVE 2ND HALF INPUT DATA
01300 3541 016440 010067 161770 71100 MOV R0, $SP ;SAVE ASSUMED END PC 'STACK POINTER'
01400 3542 016444 012067 161772 71200 MOV (R0)+, ANS3 ;SAVE 1ST HALF OF ANSWER
01500 3543 016450 012067 161770 71300 MOV (R0)+, ANS4 ;SAVE 2ND HALF OF ANSWER
01600 3544 016454 000207 71400 RTS PC
01700 3545 71500
01800 3546 71600 ;ERRONIOUS TRAP SERVICE ROUTINE
01900 3547 71700
02000 3548 016456 104000 71800 TRAPER: HLT ;FIS SHOULDN'T HAVE TRAPED
02100 3549 016460 000547 547 ;THE ERROR NUMBER IS 547
02200 3550 016462 000002 72000 RTI
02300 3551 72100

```

```

00300
00400 3552 ;*****
00500 3553 .SBTTL HLT ROUTINE (ERROR TYPEOUT)
00600 3554 016464 032737 002000 000422 HLT$: BIT #SW10,@#SWREG ;SHOULD IT RING THE BELL ON ERROR?
00700 3555 016472 001402 BEQ 1$ ;NO - SKIP
00800 3556 016474 000004 000504 TYPE ,SBELL ;RING BELL
00900 3557 016500 005267 161756 1$: INC ERRORS ;COUNT THE NUMBER OF ERRORS
01000 3558 016504 032737 020000 000422 BIT #SW13,@#SWREG ;SKIP TYPEOUT IF SET
01100 3559 016512 001023 BNE 2$ ;SKIP TYPEOUTS
01200 3560 016514 000004 000472 TYPE ,RETURN
01300 3561 016520 013637 000402 MOV @ (6)+,@#SFATAL ;PLACE THE ERROR NUMBER IN LOCATION SFATAL
01400 3562 016524 014667 161700 MOV -(6),HLTADS ;PUT ADDRESS OF INSTRUCTION ON STACK
01500 3563 016530 162767 000002 161672 SUB #2,HLTADS
01600 3564 016536 017605 000000 MOV @ (6),TTY ;TYPE @ (6) IN OCTAL
01700 3565 016542 004767 000124 JSR %7,PRINTR ;TYPE LEADING ZERO'S
01800 3566 016546 062716 000002 ADD #2,(6) ;ADJUST THE RETURN ADDRESS
01900 3567 016552 000004 000500 TYPE ,SPACE+3
02000 3568 016556 004767 000046 JSR PC,ERRORS ;GO TO USER ERROR ROUTINE
02100 3569 016562 105767 161632 2$: TSTB $ENV ;ARE WE RUNNING UNDER APT?
02200 3570 016566 001403 BEQ 4$ ;IF NOT THEN GO TO 4$
02300 3571 016570 005237 000400 INC @#MSGTY ;OTHERWISE INFORM APT
02400 3572 016574 000777 BR ;AND LOOP
02500 3573 016576 005737 000422 4$: TST @#SWREG ;HALT ON ERROR
02600 3574 016602 100001 BPL .+4 ;SKIP IF CONTINUE
02700 3575 016604 000000 HALT ;HALT ON ERROR!
02800 3576 016606 032737 001000 000422 BIT #SW09,@#SWREG ;CHECK FOR INHIBIT LOOP ON ERROR
02900 3577 016614 001001 BNE .+4 ;SKIP IF LOOP ON ERROR
03000 3578 016616 000002 RTI
03100 3579 016620 105067 161657 CLRB $ICNT
03200 3580 016624 000167 177164 JMP <ITS ;LOOP ON TEST UNTIL NO ERRORS
03300 3581
03400 3582

```

72300

00300	3583					72500
00400	3584					
00500	3585					72700
00600	3586					72800
00700	3587					72900
00800	3588	016630	117767	161574	161702	73000
00900	3589	016636	062767	000002	161674	73100
01000	3590	016644	012703	000430		73200
01100	3591	016650				73300
01200	3592	016650	012305			
01300	3593	016652	004767	000014		
01400	3594	016656	000004	000501		73400
01500	3595	016662	105367	161652		73500
01600	3596	016666	100370			73600
01700	3597	016670	000207			73700
01800	3598					73800
01900						

.SBTTL USER ERROR ROUTINE

ERROR\$: MOVB @HLTADS, TYP CNT ;TYPE COUNT IS LOW BYTE OF HLT
 ADD #2, TYP CNT ;TYPE COUNT - X+2
 MOV #HLTADS, R3 ;TOP OF DATA TO BE TYPED

ERR1\$: MOV (R3)+, TTY ;TYPE (R3)+ IN OCTAL
 JSR %7, PRINTR ;TYPE LEADING ZERO'S
 TYPE, SPACE+4 ;SPACE
 DECB TYP CNT ;CHECK FOR DONE
 BPL ERR1\$;BRANCH IF NOT DONE
 RTS PC

00100	3599	016672	112767	000001	161644	PRINTR: MOVB	#1,.PR	:SET ZERO FILL SWITCH
00200	3600	016700	000402			BR	+.6	:SKIP
00300	3601	016702	005067	161636		PRINTS: CLR	.PR	:SUPPRESS LEADING ZERO'S
00400	3602	016706	112767	177772	161631	MOVB	#-6,.PR+1	:SET COUNT
00500	3603	016714	010446			MOV	R4,-(6)	:SAVE R4
00600	3604	016716	012704	017020		MOV	#.PRBUF,R4	:SET POINTER TO FIRST ASCII CHAR.
00700	3605	016722	105014			CLRB	(4)	:CLEAR FIRST BYTE
00800	3606	016724	000405			BR	.PRF	:ROTATE FIRST BIT
00900	3607	016726	105014			.PRL: CLRB	(4)	:CLEAR BYTE OF CHARACTER
01000	3608	016730	006105			ROL	TTY	:ROTATE BIT INTO C
01100	3609	016732	106114			ROLB	(4)	:PACK IT
01200	3610	016734	006105			ROL	TTY	:ROTATE BIT INTO C
01300	3611	016736	106114			ROLB	(4)	:PACK IT
01400	3612	016740	006105			.PRF: ROL	TTY	:ROTATE BIT INTO C
01500	3613	016742	106114			ROLB	(4)	:PACK IT
01600	3614	016744	105714			TSTB	(4)	:IS IT ZERO?
01700	3615	016746	001402			BEQ	+.6	:SKIP INC
01800	3616	016750	105267	161570		INCB	.PR	:SET FILL SWITCH
01900	3617	016754	105767	161564		TSTB	.PR	:CHECK FILL SWITCH
02000	3618	016760	001402			BEQ	+.6	:SKIP BITSET
02100	3619	016762	152724	000060		BISB	#'0,(4)+	:MAKE INTO ASCII CHAR
02200	3620	016766	105267	161553		INCB	.PR+	:INC COUNT
02300	3621	016772	001355			BNE	.PRL	:REPEAT
02400	3622	016774	022704	017020		CMP	#.PRBUF,R4	:EMPTY BUFFER?
02500	3623	017000	001002			BNE	+.6	:SKIP IF NOT
02600	3624	017002	112724	000060		MOVB	#'0,(4)+	:LOAD 1 ZERO
02700	3625	017006	105014			CLRB	(4)	:NULL TERMINATOR
02800	3626	017010	000004	017020		TYPE	.PRBUF	:TYPE IT
02900	3627	017014	012604			MOV	(6)+,R4	:RESTORE R4
03000	3628	017016	000207			RTS	PC	:RETURN
03100	3629	017020	000004			.PRBUF: .BLKW	4	:OUTPUT BUFFER
03200	3630							

```

00300
00400 3631
00500 3632
00600 3633
00700 3634
00800 3635
00900 3636 017030 012737 017152 000024
01000 3637 017036 012737 000340 000026
01100 3638 017044 010046
01200 3639 017046 010146
01300 3640 017050 010246
01400 3641 017052 010346
01500 3642 017054 010446
01600 3643 017056 010546
01700 3644 017060 010667 000072
01800 3645 017064 012737 017076 000024
01900 3646 017072 000000
02000 3647 017074 000776
02100 3648
02200 3649
02300 3650 017076 016706 000054
02400 3651 017102 005067 000050
02500 3652 017106 005267 000044
02600 3653 017112 001375
02700 3654 017114 012605
02800 3655 017116 012604
02900 3656 017120 012603
03000 3657 017122 012602
03100 3658 017124 012601
03200 3659 017126 012600
03300 3660 017130 012737 017030 000024
03400 3661 017136 012737 000340 000026
03500 3662 017144 000004
03600 3663 017146 017160
03700 3664 017150 000002
03800 3665 017152 000000
03900 3666 017154 000776
04000 3667 017156 000000
04100 3668 017160 005015 047520 142527
04200 3669 017166 000122
04300 367

```

.SBTTL POWER DOWN AND UP ROUTINES

.POWER DOWN ROUTINE

```

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

```

.POWER UP ROUTINE

```

$PWRUP: MOV $SAVR6,SP ;;GET SP
CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
1$: INC $SAVR6 ;;WAIT FOR THE INC
BNE 1$ ;;OF WORD
MOV (SP)+,R5 ;;POP STACK INTO R5
MOV (SP)+,R4 ;;POP STACK INTO R4
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
MOV (SP)+,R0 ;;POP STACK INTO R0
MOV #PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
MOV #340,@#PWRVEC+2 ;;PRIO:7
TYPE $POWER ;;REPORT THE POWER FAILURE
$PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
RTI
$SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
BR -2 ;;BEFORE THE POWER DOWN WAS COMPLETE
$SAVR6: 0 ;;PUT THE SP HERE
$POWER: .ASCII <15><12>'POWER'
.EVEN

```

```

00300
00400 3671 74700
00500 3672 74800 : * TYPE OUT ROUTINE
00600 3673 74900 : * -----
00700 3674 75000 : *
00800 3675 75100 : *
00900 3676 75200 : * THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
01000 3677 75300 : *
01100 3678 75400
01200 3679 017170 132737 000040 000421 75500 $TYPE: BITB #40,@$SENVM ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
01300 3680 017176 001007 75600 BNE 3$ ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 3
01400 3681 017200 010046 75700 MOV R0,-(SP) ;OTHERWISE SAVE R0
01500 3682 017202 017600 000002 75800 MOV @2(SP),R0 ;GET THE ADDRESS OF THE ASCIZ STRING
01600 3683 017206 112046 75900 2$: MOVB (R0)+,-(SP) ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
01700 3684 017210 001005 76000 BNE 4$ ;BRANCH IF IT IS NOT THE TERMINATOR
01800 3685 017212 005726 76100 TST (SP)+
01900 3686 017214 012600 76200 MOV (SP)+,R0 ;OTHERWISE RESTORE THE STACK AND R0
02000 3687 017216 062716 000002 76300 3$: ADD #2,(SP) ;ADJUST THE RETURN PC
02100 3688 017222 00000? 76400 RTI ;AND RETURN
02200 3689 76500
02300 3690 017224 105777 161320 76600 4$: TSTB @$TPS ;IS THE PRINTER AVAILABLE?
02400 3691 017230 100375 76700 BP_ 4$ ;IF NOT THEN LOOP HERE
02500 3692 017232 112677 161314 76800 MOVB (SP)+,@$TPB ;OUT PUT THE CHARACTER
02600 3693 017236 000763 76900 BR 2$ ;AND GO BACK
02700 3694 017240 005015 053104 040513 77000 $TITLE: .ASCIZ <15><12>/DVKACC - LSI-11 FIS INSTRUCTION TEST/
02800 3695 017246 041503 026440 046040
02900 3696 017254 044523 030455 020061
03000 3697 017262 044506 020123 044440
03100 3698 017270 051516 051124 041525
03200 3699 017276 044524 047117 052040
03300 3700 017304 051505 000124
03400 3701 77100 .EVEN
03500 3702 017310 012706 000600 77200 NOOOP: MOV #BEGIN,SP ;THAT WAS THE HEADING FOR THE DIAGNOSTIC
03600 3703 017314 132737 000001 000420 77300 BITB #1,@$SENV ;INITIALIZE STACK POINTER TO 600
03700 3704 017322 001011 77400 BNE 22$ ;ARE WE UNDER APT
03800 3705 017324 132737 000040 000421 77450 BITB #40,@$SENV ;IF SO THEN DO NOT TYPE HEADING
03900 3706 017332 001005 77460 BNE 22$ ;HAVE THE CONSOLE OUTPUTS BEEN SUPPRESSED
04000 3707 017334 012737 017170 000020 77500 MOV #STYPF,@#20 ;IF SO THEN DO NOT PRINT HEADING
04100 3708 017342 000004 017240 77600 TYPE . $TITLE ;SET UP VECTOR 20 TO PRINT HEADING
04200 3709 017346 012767 000001 161162 77700 22$: MOV #1,TIMES ;TYPE HEADING 'VKACC -LSI-11 ...'
04300 3710 017354 012700 000410 77800 MOV #SDEVCT,R0 ; # OF ITERATIONS IN THE FIRST PASS
04400 3711 017360 005040 77900 2$: CLR -(R0) ;PREPARE TO INITIALIZE THE PROGRAM
04500 3712 017362 022700 000400 78000 CMP #SMALL,R0
04600 3713 017366 001374 78100 BNE 2$
04700 3714 017370 000167 160610 78200 JMP RESTRT ;START THE PROGRAM
04800 3715 000001 78300 .END

```


00100						
00200						
00300						
00400	TST11	002272	605#			
00500	TST12	002430	651#			
00600	TST13	002566	697#			
00700	TST14	002720	743#			
00800	TST15	003054	789#			
00900	TST16	003214	835#			
01000	TST17	003366	889#			
01100	TST2	001056	283#			
01200	TST20	003620	961#			
01300	TST21	004050	1033#			
01400	TST22	004306	1105#			
01500	TST23	004540	1177#			
01600	TST24	004700	1223#			
01700	TST25	005034	1269#			
01800	TST26	005170	1321#			
01900	TST27	005326	1367#			
02000	TST3	001216	329#			
02100	TST30	005462	1419#			
02200	TST31	005622	1465#			
02300	TST32	005762	1511#			
02400	TST33	006122	1557#			
02500	TST34	006260	1603#			
02600	TST35	006414	1649#			
02700	TST36	006552	1695#			
02800	TST37	006706	1747#			
02900	TST4	001352	375#			
03000	TST40	007140	1819#			
03100	TST41	007376	1891#			
03200	TST42	007532	1937#			
03300	TST43	007672	1983#			
03400	TST44	010026	2029#			
03500	TST45	010162	2075#			
03600	TST46	010316	2121#			
03700	TST47	010454	2167#			
03800	TST5	001506	421#			
03900	TST50	010630	2221#			
04000	TST51	011004	2275#			
04100	TST52	011240	2347#			
04200	TST53	011472	2419#			
04300	TST54	011630	2465#			
04400	TST55	011764	2511#			
04500	TST56	012122	2557#			
04600	TST57	012256	2603#			
04700	TST6	001640	467#			
04800	TST60	012432	2657#			
04900	TST61	012670	2729#			
05000	TST62	013126	2801#			
05100	TST63	013364	2873#			
05200	TST64	013616	2953#			
05300	TST65	014034	3012#			
05400	TST66	014312	3084#			
05500	TST67	014530	3142#			
05600	TST7	001772	513#			
05700	TST70	015160	3250#			
05800	TYOUT	000546	191#	3744	3252	3351*
05900	TYCNT	000540	188#	3588*	3589*	3595*

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 98
00200 CVKACC.P11 16-AUG-78 08:41

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0105

		76#	3149	3257	3375	3556	3560	3567	3594	3626	3662	3708		
00400	TYPE = 000004													
00500	YESRT 000542	189#	209											
00600	SAPTHD 000430	136	142#	149										
00700	SBELL 000504	176#	3556											
00800	SCPUPP 000426	118#												
00900	SDEVCT 000410	109#	3710											
01000	SDOAGN 015674	3371	3379	3385#										
01100	SENDAD 015664	89	3381#											
01200	SENDCT 015644	3373#	3392											
01300	SENDMG 015700	3375	3387#											
01400	SENULL 015712	3389#												
01500	SENV 000420	114#	219	3569	3703									
01600	SENVN 000421	115#	3142	3250	3679	3705								
01700	SEOP 015620	3365#												
01800	SEOPCT 015636	3370#	3374											
01900	SETABL 000420	113#												
02000	SETEND 000430	125#	148											
02100	SF - 000550	78#	252	253#	257	258#	262	263#	267	268#	272	273#	298	29
02200		303	304#	308	309#	313	314#	318	319#	344	345#	349	350#	35
02300		355#	359	360#	364	365#	390	391#	395	396#	400	401#	405	40
02400		410	411#	435	436#	441	442#	446	447#	451	452#	456	457#	48
02500		482#	487	488#	492	493#	497	498#	502	503#	528	529#	533	53
02600		538	539#	543	544#	548	549#	574	575#	579	580#	584	585#	58
02700		590#	594	595#	620	621#	625	626#	630	631#	635	636#	640	64
02800		606	667#	671	672#	676	677#	681	682#	686	687#	711	712#	71
02900		718#	722	723#	727	728#	732	733#	757	758#	763	764#	768	76
03000		773	774#	778	779#	804	805#	809	810#	814	815#	819	820#	82
03100		825#	853	854#	858	859#	863	864#	868	869#	873	874#	878	87
03200		902	903#	910	911#	915	916#	920	921#	925	926#	930	931#	93
03300		936#	940	941#	945	946#	950	951#	972	973#	981	982#	987	98
03400		992	993#	997	998#	1002	1003#	1007	1008#	1012	1013#	1017	1018#	102
03500		1023#	1046	1047#	1054	1055#	1059	1060#	1064	1065#	1069	1070#	1074	107
03600		1079	1080#	1084	1085#	1089	1090#	1094	1095#	1116	1117#	1125	1126#	113
03700		1132#	1136	1137#	1141	1142#	1146	1147#	1151	1152#	1156	1157#	1161	116
03800		166	1167#	1192	1193#	1197	1198#	1202	1203#	1207	1208#	1212	1213#	123
03900		1239#	1243	1244#	1248	1249#	1253	1254#	1258	1259#	1289	1290#	1295	129
04000		1300	1301#	1305	1306#	1310	1311#	1336	1337#	1341	1342#	1346	1347#	135
04100		1352#	1356	1357#	1387	1388#	1393	1394#	1398	1399#	1403	1404#	1408	140
04200		1434	1435#	1439	1440#	1444	1445#	1449	1450#	1454	1455#	1480	1481#	148
04300		1486#	1490	1491#	1495	1496#	1500	1501#	1526	1527#	1531	1532#	1536	153
04400		1541	1542#	1546	1547#	1572	1573#	1577	1578#	1582	1583#	1587	1588#	159
04500		1593#	1618	1619#	1623	1624#	1628	1629#	1633	1634#	1638	1639#	1664	156
04600		1669	1670#	1674	1675#	1679	1680#	1684	1685#	1715	1716#	1721	1722#	172
04700		1727#	1731	1732#	1736	1737#	1758	1759#	1767	1768#	1773	1774#	1778	177
04800		1783	1784#	1788	1789#	1797	1794#	1798	1799#	1803	1804#	1808	1809#	183
04900		1833#	1840	1841#	1845	1848#	1850	1851#	1855	1856#	1860	1861#	1865	186
05000		1870	1871#	1875	1876#	1880	1881#	1906	1907#	1911	1912#	1916	1917#	192
05100		1922#	1926	1927#	1952	1953#	1957	1958#	1962	1963#	1967	1968#	1972	197
05200		1998	1999#	2003	2004#	2008	2009#	2013	2014#	2018	2019#	2044	2045#	204
05300		2050#	2054	2055#	2059	2060#	2064	2065#	2089	2090#	2095	2096#	2100	210
05400		2105	2106#	2110	2111#	2136	2137#	2141	2142#	2146	2147#	2151	2152#	215
05500		2157#	2185	2186#	2190	2191#	2195	2196#	2200	2201#	2205	2206#	2210	221
05600		2239	2240#	2244	2245#	2249	2250#	2254	2255#	2259	2260#	2264	2265#	228
05700		2289#	2296	2297#	2301	2302#	2306	2307#	2311	2312#	2316	2317#	2321	232
05800		2326	2327#	2331	2332#	2336	2337#	2358	2359#	2367	2368#	2373	2374#	237
05900		2379#	2383	2384#	2388	2389#	2393	2394#	2398	2399#	2403	2404#	2408	240

CROSS REFERENCE TABLE -- USER SYMBOLS

00300															
00400		2434	2435#	2439	2440#	2444	2445#	2449	2450#	2454	2455#	2479	2480#	248	
00500		2486#	2490	2491#	2495	2496#	2500	2501#	2526	2527#	2531	2532#	2536	253	
00600		2541	2542#	2546	2547#	2572	2573#	2577	2578#	2582	2583#	2587	2588#	259	
00700		2593#	2621	2622#	2626	2627#	2631	2632#	2636	2637#	2641	2642#	2646	264	
00800		2670	2671#	2678	2679#	2683	2684#	2688	2689#	2693	2694#	2698	2699#	270	
00900		2704#	2708	2709#	2713	2714#	2718	2719#	2742	2747#	2750	2751#	2755	275	
01000		2760	2761#	2765	2766#	2770	2771#	2775	2776#	2780	2781#	2785	2786#	279	
01100		2791#	2814	2815#	2822	2823#	2827	2828#	2832	2833#	2837	2838#	2842	284	
01200		2847	2848#	2852	2853#	2857	2858#	2862	2863#	2884	2885#	2893	2894#	289	
01300		2900#	2904	2905#	2915	2916#	2920	2921#	2925	2926#	2930	2931#	2935	293	
01400		2940	2941#	2982	2983#	2987	2988#	2993	2994#	2998	2999#	3003	3004#	302	
01500		3030#	3037	3038#	3042	3043#	3047	3048#	3052	3053#	3057	3058#	3062	306	
01600		3067	3068#	3072	3073#	3076	3077#	3102	3103#	3110	3111#	3115	3116#	312	
01700		3121#	3125	3126#	3130	3131#	3191	3192#	3198	3199#	3203	3204#	3208	320	
01800		3214	3215#	3219	3220#	3224	3225#	3229	3230#	3238	3239#	3299	3300#	330	
01900		3307#	3311	3312#	3316	3317#	3322	3323#	3327	3328#	3332	3333#	3337	333	
02000		3346	3347#	3549	3550#										
02100	\$FATAL	000402	106#	3561*											
02200	\$GET42	015654	3376#												
02300	\$HD	000003	14	15											
02400	\$HIBTS	000430	143#												
02500	\$ICNT	000503	174#	3405	3407	3409*	3414*	3579*							
02600	\$ILLUP	017152	3636	3665#											
02700	\$MAIL	000400	104#	144	148	3712									
02800	\$MBADR	000432	144#												
02900	\$MSGAD	000414	111#												
03000	\$MSGLG	000416	112#												
03100	\$MSGTY	000400	105#	3571*											
03200	\$PASS	000406	108#	3367*	3368*	3387									
03300	\$PASTM	000436	146#												
03400	\$POWER	017160	3663	3668#											
03500	\$PSW	000432	152#	153	249	295	341	387	438	484	525	571	617	663	71
03600			760	801	850	907	984	1051	1128	1189	1235	1292	1333	1390	143
03700			1477	1523	1569	1615	1661	1718	1770	1837	1903	1949	1995	2041	209
03800			2133	2182	2236	2293	2370	2431	2482	2523	2569	2616	2675	2747	281
03900			2896	2974*	2975*	2979	3034	3099*	3107	3195	3303	3442*	3443*	3457*	349
04000			3491*	3503*	3535*	3536*									
04100	\$PWRDN	017030	21	3636#	3660										
04200	\$PWRMG	017146	3663#												
04300	\$PWRUP	017076	3645	3650#											
04400	\$SAVR6	017156	3644*	3650	3651*	3652*	3667#								
04500	\$SETUP=	000020	82#	3367											
04600	\$SP	000434	154#	155	248*	254	294*	300	340*	346	386*	392	524*	530	57
04700			576	616*	622	662*	668	800*	806	900*	906*	912	1044*	1050*	105
04800			1188*	1194	1234*	1240	1332*	1338	1430*	1436	1476*	1482	1522*	1528	156
04900			1574	1614*	1620	1660*	1666	1830*	1836*	1842	1902*	1908	1948*	1954	199
05000			2000	2040*	2046	2132*	2138	2286*	2292*	2298	2430*	2436	2522*	2528	256
05100			2574	2668*	2674*	2680	2740*	2746*	2752	2812*	2818*	2824	2978*	2984	302
05200			3033*	3039	3100*	3106*	3112	3188*	3194*	3200	3296*	3302*	3308	3447*	346
05300			3541*												
05400	\$STUP	177777	82#												
05500	\$SVPC	001000	87#	92											
05600	\$SWR	160000	14	15#	3360	3367	3377	3386	3387	3664					
05700	\$SWREG	000422	116#	3147	3255	3397	3399	3401	3403	3554	3558	3573	3576		
05800	\$TFSTN	000404	107#	225*	269	315	361	407	453	499	545	591	637	683	72
05900			775	821	875	947	1019	1091	1163	1209	1255	1307	1353	1405	145

CVK100 (VKAC MARY 1 30A(1052) 21-AUG-78 15:28 PAGE 104
00200 (VKACC.P11 16-AUG-78 08:41 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0110

00300			
00400	.SDB2D	1#	
00500	.SDB2I	1#	
00600	.SDI	1#	
00700	.SEOP	1#	4# 3354
00800	.SEKR	1#	
00900	.SEKRT	1#	
01000	.SMULT	1#	
01100	.SPOWE	1#	4# 3631
01200	.SRAND	1#	
01300	.SRDDE	1#	
01400	.SRDOC	1#	
01500	.SREAD	1#	
01600	.SR2AZ	1#	
01700	.SSAVE	1#	
01800	.SSB2D	1#	
01900	.SSB2O	1#	
02000	.SSCOP	1#	
02100	.SSIZE	1#	
02200	.SSUPR	1#	
02300	.STRAP	1#	
02400	.STYPB	1#	
02500	.STYPD	1#	
02600	.STYPE	1#	
02700	.STYPO	1#	
02800	.S40CA	1#	

02900
03000
03100 . ABS. 017374 000

03200
03300
03400 ERRORS DETECTED: 0

03500
03600 CVKACC.BIN,CVKACC.LST/CRF/SOL=CVKACC.SML,CVKACC.P11
03700 RUN-TIME: 17 23 1 SECONDS
03800 RUN-TIME RATIO: 575/42 13.4
03900 CORE USED: 41K (81 PAGES)

04000
04100