

LSI-11

FIS INSTRUCTION
CVKACC0

AH-8196C-MC

COPYRIGHT 75-78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230
231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260
261	262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279	280
281	282	283	284	285	286	287	288	289	290
291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310
311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330
331	332	333	334	335	336	337	338	339	340
341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370
371	372	373	374	375	376	377	378	379	380
381	382	383	384	385	386	387	388	389	390
391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410
411	412	413	414	415	416	417	418	419	420
421	422	423	424	425	426	427	428	429	430
431	432	433	434	435	436	437	438	439	440
441	442	443	444	445	446	447	448	449	450
451	452	453	454	455	456	457	458	459	460
461	462	463	464	465	466	467	468	469	470
471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490
491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510
511	512	513	514	515	516	517	518	519	520
521	522	523	524	525	526	527	528	529	530
531	532	533	534	535	536	537	538	539	540
541	542	543	544	545	546	547	548	549	550
551	552	553	554	555	556	557	558	559	560
561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576	577	578	579	580
581	582	583	584	585	586	587	588	589	590
591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610
611	612	613	614	615	616	617	618	619	620
621	622	623	624	625	626	627	628	629	630
631	632	633	634	635	636	637	638	639	640
641	642	643	644	645	646	647	648	649	650
651	652	653	654	655	656	657	658	659	660
661	662	663	664	665	666	667	668	669	670
671	672	673	674	675	676	677	678	679	680
681	682	683	684	685	686	687	688	689	690
691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710
711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730
731	732	733	734	735	736	737	738	739	740
741	742	743	744	745	746	747	748	749	750
751	752	753	754	755	756	757	758	759	760
761	762	763	764	765	766	767	768	769	770
771	772	773	774	775	776	777	778	779	780
781	782	783	784	785	786	787	788	789	790
791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810
811	812	813	814	815	816	817	818	819	820
821	822	823	824	825	826	827	828	829	830
831	832	833	834	835	836	837	838	839	840
841	842	843	844	845	846	847	848	849	850
851	852	853	854	855	856	857	858	859	860
861	862	863	864	865	866	867	868	869	870
871	872	873	874	875	876	877	878	879	880
881	882	883	884	885	886	887	888	889	890
891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910
911	912	913	914	915	916	917	918	919	920
921	922	923	924	925	926	927	928	929	930
931	932	933	934	935	936	937	938	939	940
941	942	943	944	945	946	947	948	949	950
951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970
971	972	973	974	975	976	977	978	979	980
981	982	983	984	985	986	987	988	989	990
991	992	993	994	995	996	997	998	999	1000

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 RESTART (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 \$ENVN 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 \$ENVN 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 \$ENVN 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 'SPASS' (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 'STESTN' (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.

00008.5 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.

00009.0 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

00100	CVKACC	MA(Y11 30A(1052)	21-AUG-78	15:28	
00200	CVKACC.P11	16-AUG-78	08:41	TABLE OF CONTENTS	
00300					
00400	32	55100		SWITCH OPTIONS AND ASSIGNMENTS	
00500	81			ACT11 HOOKS	
00600	91	60500		VECTOR AREA, STACKS, ANSWER AREA, AND SETUP ROUTINE	
00700	94			APT MAILBOX-ETABLE	
00800	126			APT PARAMETER BLOCK	
00900	193	38300		STARTING OF THE PROGRAM	
01000	226	41700		FADD TEST SECTION	
01100	879	43500		TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW	
01200	1024	44000		TEST FLOATING ADD INSTRUCTION WITH OVERFLOW	
01300	1169	44500		FSUB TEST SECTION	
01400	1740	46000		TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW	
01500	1813	46400		TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW	
01600	1886	46800		FMUL TEST SECTION	
01700	2271	48100		TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW	
01800	2344	48500		TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW	
01900	2417	48900		FDIV TEST SECTION	
02000	2656	49800		TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW	
02100	2729	50200		TEST FLOATING DIV INSTRUCTION WITH OVERFLOW	
02200	2802	50600		TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO	
02300	2953	51200		TEST OF ALL FIS AT ONCE	
02400	3016	51700		ADDRESS ERROR TEST	
02500	3145	52500		INTERUPT ABORT TEST SECTION	
02600	3369			END OF PASS ROUTINE	
02700	3408			SCOPE ROUTINE	
02800	3433	54700		PUSH AND POP SUBROUTINES	
02900	3567			HIT ROUTINE (ERROR TYPEOUT)	
03000	3600	72800		USER ERROR ROUTINE	
03100	3613			OCTAL WORD & ADDRESS TYPER	
03200	3648			POWER DOWN AND UP ROUTINES	
03300	3686	74400		ASCII TYPE OUT ROUTINE	

00100 (VKACC MA(V11 30A(1052) 21-AUG-78 15:28 PAGE 2
00200 (VKACC.P11 16-AUG-78 08:41
00300
00400 31700

K 1

SEQ 0010

00100 CVKACC MAC111 30A(1052) 21-AUG-78 15:28 PAGE 3
 00200 CVKACC.P11 16-AUG-78 08:41
 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

000001
 160000

104000
 000000

31900
 32000

32900
 33600
 34000
 35400
 35500
 36900
 40200
 40700
 41200
 42000
 45100
 49600
 54900
 55300

.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

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

55500 ;ERROR MESSAGE FORMAT:
 55600 ; ERRNM ADR PSW SP ANS1 ANS2 ANS3 ANS4 ANS5 ANS6
 55700
 55800 ;WHERE ERRNM- ERROR NUMBER
 55900 ; ADR - ADDRESS OF 'HLT' INSTRUCTION + 2
 56000 ; PSW - PROCESSOR STATUS WORD
 56100 ; SP - STACK POINTER
 56200 ; ANS1 THRU ANS6 - DATA OFF THE STACK(S)
 56300 ; NOTE: ANS1 THRU ANS6 ARE NOT ALWAYS TYPED, DEPENDING ON THE
 56400 ; NUMBER ADDED TO THE 'HLT'. 'HLT' ALONE TYPES NONE,
 56500 ; 'HLT+1' TYPES ANS1, 'HLT+2' TYPES ANS1 AND ANS2, ETC.
 56600
 56700 HLT EMT
 56800 R0- %0

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			56900	R1	%1	
00400	58	000001	57000	R2-	%2	
00500	59	000002	57100	R3	%3	
00600	60	000003	57200	R4	%4	
00700	61	000004	57300	R5	%5	
00800	62	000005	57400	TTY=	%5	
00900	63	000005	57500	SP	%6	
01000	64	000006	57600	PC-	%7	
01100	65	000007	57700	PWRVEC	24	
01200	66	000024	57800	SCOPE=	TRAP	
01300	67	104400	57900	SW15=	100000	
01400	68	100000	58000	SW14-	40000	
01500	69	040000	58100	SW13-	20000	
01600	70	020000	58200	SW12=	10000	
01700	71	010000	58300	SW11-	4000	
01800	72	004000	58400	SW10	2000	
01900	73	002000	58500	SW09	1000	
02000	74	001000	58600	SW08=	400	
02100	75	000400	58700	TYPE-	IOT	
02200	76	000004	58800	N=	1	
02300	77	000001	58900	\$F	1	
02400	78	000001				
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			

0 ;TRAP CATCHER FROM 0 - 776

.SBTTL ACT11 HOOKS
 ;HOOKS REQUIRED BY ACT11
 \$SVPC-
 .-46
 \$ENDAD
 . 52
 .WORD 0
 .-\$SVPC

;SAVE PC
 ;;1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .SEOP
 ;;2)SET LOC.52 TO ZERO
 ;; RESTORE PC

(C100 CVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 5
 00200 CVKACC.P11 16-AUG-78 08:41 VECTOR AREA, STACKS, ANSWER AREA, AND SETUP ROUTINE

SEQ 0013

```

00300
00400 95 60800
00500 96 000400 60900
00600 97 :*****
00700 98
00800 99 .SBTTL APT MAILBOX-ETABLE
00900 100
01000 101
01100 102
01200 103 .EVEN
01300 104 000400 $MAIL: ::APT MAILBOX
01400 105 000400 000000 $MSGTY: .WORD AMSGTY ::MESSAGE TYPE CODE
01500 106 000402 000000 $FATAL: .WORD AFATAL ::FATAL ERROR NUMBER
01600 107 000404 000000 $TESTN: .WORD ATESTN ::TEST NUMBER
01700 108 000406 000000 $PASS: .WORD APASS ::PASS COUNT
01800 109 000410 000000 $DEVCT: .WORD ADEVCT ::DEVICE COUNT
01900 110 000412 000000 $UNIT: .WORD AUNIT ::I/O UNIT NUMBER
02000 111 000414 000000 $MSGAD: .WORD AMSGAD ::MESSAGE ADDRESS
02100 112 000416 000000 $MSGLG: .WORD AMSGLG ::MESSAGE LENGTH
02200 113 000420 $ETABLE: ::APT ENVIRONMENT TABLE
02300 114 000420 000 $ENV: .BYTE AENV ::ENVIRONMENT BYTE
02400 115 000421 000 $ENVM: .BYTE AENVM ::ENVIRONMENT MODE BITS
02500 116 000422 000000 $SWREG: .WORD ASWREG ::APT SWITCH REGISTER
02600 117 000424 000000 $USWR: .WORD AUSWR ::USER SWITCHES
02700 118 000426 000000 $CPUOP: .WORD ACPUOP ::CPU TYPE,OPTIONS
02800 119 :* BITS 15-11=CPU TYPE
02900 120 :* 11/04=01,11/05-02,11/20-03,11/40 04,11/45=05
03000 121 :* 11/70=06,PDQ=07,Q-10
03100 122 :* BIT 10=REAL TIME CLOCK
03200 123 :* BIT 9=FLOATING POINT PROCESSOR
03300 124 :* BIT 8=MEMORY MANAGEMENT
03400 125 000430 $ETEND:
03500 126 .MEXIT
03600 127 :*****
03700 128
03800 129 .SBTTL APT PARAMETER BLOCK
03900 130 :SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
04000 131 :*****
04100 132 000430 .$. ::SAVE CURRENT LOCATION
04200 133 000024 .=24 ::SET POWER FAIL TO POINT TO START OF PROGRAM
04300 134 000024 000200 200 ::FOR APT START UP
04400 135 000044 .=44 ::POINT TO APT INDIRECT ADDRESS PNTR.
04500 136 000044 000430 $APTHDR ::POINT TO APT HEADER BLOCK
04600 137 000430 .-$. ::RESET LOCATION COUNTER
04700 138 :*****
04800 139 :SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
04900 140 :INTERFACE SPEC.
05000 141
05100 142 000430 $APTHD:
05200 143 000430 000000 $HIBTS: .WORD 0 ::TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
05300 144 000432 000400 $MBADR: .WORD $MAIL ::ADDRESS OF APT MAILBOX (BITS 0-15)
05400 145 000434 000003 $STM: .WORD 3 ::RUN TIME OF LONGEST TEST
05500 146 000436 000005 $PASTM: .WORD 5 ::RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VER
05600 147 000440 000000 $UNITM: .WORD ::ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDI
05700 148 000442 000014 .WORD $ETEND-$MAIL/2 ::LENGTH MAILBOX-ETABLE (WORD )
05800 149 000430 $APTHD
05900 150 000430 HL TAD$:
  
```

61200
 61300

```

00100 CVKACC MAC(11 30A(1052) 21-AUG-78 15:28 PAGE 6
00200 CVKACC.P11 16-AUG-78 08:41 APT PARAMETER BLOCK
00300
00400 151 000432 000432 61400 HLTAD$+2
00500 152 000432 000432 61500 $PSW: .- ;PROCESSOR STATUS WORD
00600 153 000434 000434 61600 $PSW+2
00700 154 000434 000434 61700 $SP: . ;STACK POINTER
00800 155 000436 000436 61800 $SP+2
00900 156 000436 000436 61900 ANS1: . ;FIRST ANSWER (SEE CODE)
01000 157 000440 000440 62000 ANS1+2
01100 158 000440 000440 62100 ANS2: .
01200 159 000442 000442 62200 ANS2+2
01300 160 000442 000442 62300 ANS3: .
01400 161 000444 000444 62400 ANS3+2
01500 162 000444 000000 62500 ANS4: 0
01600 163 000446 000000 62600 ANS5: 0
01700 164 000450 000000 62700 ANS6: 0
01800 165 000452 000000 000000 000000 62800 0,0,0,0 ;NON-16 STACK BUFFER
01900 166 000460 000000
02000 167 000462 000000 62900 ERRORS: 0
02100 168 000464 000244 63000 FISVEC: 244 ;FIS TRAP VECTOR ADDRESS
02200 169 000466 000246 63100 FISLVL: 246
02300 170 000470 000000 63200 LADS: 0
02400 171 000472 006412 000 020012 020040 63300 RETURN: .ASCIZ <12><15>.. ;RETURN AND LINEFEED
02500 172 000475 015 020012 020040 63400 SPACE: .ASCIZ <15><12>.. ;RETURN AND 3 SPACES
02600 173 000502 000
02700 174 000503 000
02800 175
02900 176 000504 000007 63500 $ICNT: .BYTE 0
03000 177 000506 000000 63600 .EVEN
03100 178 000510 000000 63700 $BELL: .WORD 7 ;RING A BELL
03200 179 000512 000000 63800 SAVTIPS: 0 ;LOC TO SAVE TELEPRINTER STATUS
03300 180 000514 000000 63900 STACK0: 0 ;NON-16 STACK NORMAL LIMIT
03400 181 000516 000000 64000 STACK2: 0
03500 182 000520 000000 000000 000000 64100 STACK4: 0
03600 183 000526 000000 000000 000000 64200 STACK6: 0
03700 184 000532 000000 000000 000000 64300 STACK8: 0,0,0,0,0 ;NON-16 STACK BUFFER
03800 185 000511 000511 64400 STAK10: 0
03900 186 000534 000000 64500 STACK1 = STACK0+1
04000 187 000536 000000 64600 TEMP: 0
04100 188 000540 000000 64700 TIMES: 0
04200 189 000542 000006 64800 TYPCNT: 0
04300 190 000544 000000 64900 YESRT: RTT ;RETURN FROM TRACE TRAP
04400 191 000546 000064 65000 .PR: 0 ;COUNT AND SWITCH
04500 192 000550 177564 65100 TTYOUT: 64
04600 193 000552 177566 65200 $TPS: 177564 ;TTY PRINTER STATUS REG.
65300 $TPB: 177566 ;TTY PRINTER BUFFER REG.

```

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 7
 CVKACC.P11 16-AUG-78 08:41 APT PARAMETER BLOCK

SEQ

```

00100 194
00200 195
00300 196
00400 197
00500 198
00600 199
00700 200
00800 201
00900 202
01000 203
01100 204
01200 205
01300 206
01400 207
01500 208
01600 209
01700 210
01800 211
01900 212
02000 213
02100 214
02200 215
02300 216
02400 217
02500 218
02600 219
02700 220
02800 221
02900 222
03000 223
03100 224
03200 225
03300 226
03400 227
03500 228
03600
03700
03800

000200 000200 017104
000204 000167 000370

000600 012706 000600
000604 012737 000542 000014
000612 012737 017170 000020
000620 012737 017030 000024

000626 012700 000030
000632 012720 016464
000636 012720 000340
000642 012720 015716
000646 012710 000340
000652 012737 000006 000004
000660 132737 000000 000420
000666 001410
000670 012700 000554
000674 012740 176566
000700 012740 176564
000704 012740 000074
000710 005067 177470
000714 005067 177550

38200
38300
38400
38500
38600
38700
38800
38900
39000
39100
39200
39300
39400
39500
39600
39700
39800
39900
40000
40100
40200
40300
40400
40500
40600
40700
40800
40900
41000
41100
41200
41300
41400
41500

;*****
.SBTTL STARTING OF THE PROGRAM

.      200
      JMP      NOOP      ;GO TO TYPE HEADING & INITIALIZATION
RESTR: 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
                                ;SET RO TO VECTOR 30
                                ;SET EMT VECTOR
                                ;SET TRAP VECTOR
1$:   MOV      #30, R0
      MOV      #HLTS, (0)+
      MOV      #340, (0)+
      MOV      #SCOPE$, (0)+
      MOV      #340, (0)
      MOV      #6, @#4
      BITB     #1, @#SENV
      BEQ      2$
      MOV      #STPB+2, R0
      MOV      #176566, -(R0)
      MOV      #176564, -(R0)
      MOV      #74, -(R0)
2$:   CLR      $TESTN
      CLR      LADS          ;CLEAR LOOP ADDRESS

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 8
 00200 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0016

```

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 2 ;STACK POINTER (R0) NOT EQUAL TO #STACK4
 ;THE ERROR NUMBER IS 2

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

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

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

00100 CVKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 9
 00200 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0017

```

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      290
01900      291 001104 000240
02000      292 001106 075003
02100      293
02200      294 001110 004767 015124
02300      295 001114 010367 177314
02400      296 001120 122767 000010 177304
02500      297 001126 001402
02600      298 001130 104000
02700      299 001132 000006
02800      300
02900      301 001134 022767 000514 177272
03000      302 001142 001402
03100      303 001144 104000
03200      304 001146 000007
03300      305
03400      306 001150 022767 152525 177260
03500      307 001156 001402
03600      308 001160 104002
03700      309 001162 000010
03800      310
03900      311 001164 022767 052524 177246
04000      312 001172 001402
04100      313 001174 104002
04200      314 001176 000011
04300      315
04400      316 001200 122767 000002 177176
04500      317 001206 001402
04600      318 001210 104000
04700      319 001212 000012
04800      320
04900

```

 :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

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 10
 00200 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0018

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
 .WORD 040200,000000 ;SECOND OPERAND ON TOP
 .WORD 040200,000000 ;FIRST OPERAND ON BOTTOM
 .WORD 040 ;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'
 TSTB \$PSW ;CHECK PS (EXCEPT T BJJ)
 BEQ .+6 ;BRANCH IF OK
 HLT 13 ;PS NOT EQUAL TO 000
 ;THE ERROR NUMBER IS 13

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

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

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

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

00100 CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 11
 00200 CVKACC.F11 16-AUG-78 08:41 FADD TEST SECTION

SEU 0019

00300 367
 00400 368
 00500 369
 00600 370
 00700 371
 00800 372
 00900 373
 01000 374 001350 104400
 01100 375 001352 004567 014630
 01200 376 001356 177777 177777
 01300 377 001362 077777 177777
 01400 378 001366 000100
 01500 379 001370 016456 000340
 01600 380 001374 012702 000510
 01700 381
 01800 382 001400 000240
 01900 383 001402 075002
 02000 384
 02100 385 001404 004767 014630
 02200 386 001410 010267 177020
 02300 387 001414 122767 000004 177010
 02400 388 001422 001402
 02500 389 001424 104000
 02600 390 001426 000020
 02700 391
 02800 392 001430 022767 000514 176776
 02900 393 001436 001402
 03000 394 001440 104000
 03100 395 001442 000021
 03200 396
 03300 397 001444 005767 176766
 03400 398 001450 001402
 03500 399 001452 104002
 03600 400 001454 000022
 03700 401
 03800 402 001456 005767 176756
 03900 403 001462 001402
 04000 404 001464 104002
 04100 405 001466 000023
 04200 406
 04300 407 001470 122767 000004 176706
 04400 408 001476 001402
 04500 409 001500 104000
 04600 410 001502 000024
 04700 411
 04800 412
 04900 413

```

*****
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
      .WORD 177777,177777 :SECOND OPERAND ON TOP
      .WORD 077777,177777 :FIRST OPERAND ON BOTTOM
      .WORD 100 :PROCESSOR PRIORITY LEVEL
      .WORD TRAPER,340 :FIS TRAP VECTOR
      MOV #STACK0,R2 :SET UP STACK POINTER

      NOP
      FADD R2 :FLOATING ADD ON THE R2 STACK

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

      CMP #STACK4,SSP :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 CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 12
 00200 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0020

00300 413
 00400 414
 00500 415
 00600 416
 00700 417
 00800 418
 00900 419
 01000 420
 01100 421 001504 104400
 01200 422 001506 004567 014322
 01300 423 001512 152525 052524
 01400 424 001516 052525 052525
 01500 425 001522 000217
 01600 426 001524 016456 000340
 01700 427
 01800 428 001530 000240
 01900 429 001532 075006
 02000 430
 02100 431 001534 004767 014334
 02200 432 001540 022706 000600
 02300 433 001544 001405
 02400 434 001546 012706 000600
 02500 435 001552 104000
 02600 436 001554 000025
 02700 437 001556 000421
 02800 438
 02900 439 001560 122767 000200 176644
 03000 440 001566 001402
 03100 441 001570 104000
 03200 442 001572 000026
 03300 443
 03400 444 001574 022767 044600 176634
 03500 445 001602 001402
 03600 446 001604 104002
 03700 447 001606 000027
 03800 448
 03900 449 001610 005767 176624
 04000 450 001614 001402
 04100 451 001616 104002
 04200 452 001620 000030
 04300 453
 04400 454 001622 122767 000005 176554
 04500 455 001630 001402
 04600 456 001632 104000
 04700 457 001634 000031
 04800 458
 04900 459

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

SCOPE
 TST5: JSR R5 PUSH5 ; 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	CVKACC	MA	30A(1052)	21-AUG-78	15
00200	CVKACC	P	16-AUG-78	08:41	
00300					
00400	459				
00500	460				
00600	461				
00700	462				
00800	463				
00900	464				
01000	465				
01100	466	001636	104400		
01200	467	001640	004567	014170	
01300	468	001644	025177	177777	
01400	469	001650	125200	000000	
01500	470	001654	000307		
01600	471	001656	016456	000340	
01700	472				
01800	473	001662	000240		
01900	474	001664	075006		
02000	475				
02100	476	001666	004767	014202	
02200	477	001672	022706	000600	
02300	478	001676	001405		
02400	479	001700	012706	000600	
02500	480	001704	104000		
02600	481	001706	000032		
02700	482	001710	000421		
02800	483				
02900	484	001712	122767	000210	176512
03000	485	001720	001402		
03100	486	001722	104000		
03200	487	001724	000033		
03300	488				
03400	489	001726	022767	117200	176502
03500	490	001734	001402		
03600	491	001736	104002		
03700	492	001740	000034		
03800	493				
03900	494	001742	005767	176472	
04000	495	001746	001402		
04100	496	001750	104002		
04200	497	001752	000035		
04300	498				
04400	499	001754	122767	000006	176422
04500	500	001762	001402		
04600	501	001764	104000		
04700	502	001766	000036		
04800	503				
04900	504				

```

*****
TEST 6:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
             125200.000000 + 025177.177777 = 117200.000000
             PS      210.      STACK POINTER      SP
*****

```

```
SCOPE
TST6:      JSR      R5, PUSH5      ;PUSH 4 WORDS ONTO STACK, SET PRIORIT
          .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
RR      END6           ;SKIP REST OF TEST

```

```

ISA6:  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
BEG      .+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

```

CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 14
 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0022

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

505
 506
 507
 508
 509
 510
 511
 512 001770 104400
 513 001772 004567 014210
 514 001776 100125 052525
 515 002002 135753 024642
 516 002006 000347
 517 002010 016456 000340
 518 002014 012705 000510
 519
 520 002020 000240
 521 002022 075005
 522
 523 002024 004767 014210
 524 002030 010567 176400
 525 002034 122767 000210 176370
 526 002042 001402
 527 002044 104000
 528 002046 000037
 529
 530 002050 022767 000514 176356
 531 002056 001402
 532 002060 104000
 533 002062 000040
 534
 535 002064 022767 135753 176344
 536 002072 001402
 537 002074 104002
 538 002076 000041
 539
 540 002100 022767 024642 176332
 541 002106 001402
 542 002110 104002
 543 002112 000042
 544
 545 002114 122767 000007 176262
 546 002122 001402
 547 002124 104000
 548 002126 000043
 549
 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
 .WORD 100125,052525 ; SECOND OPERAND ON TOP
 .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 (VKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 15
00200 (VKACC.P1 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0023

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

```

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

TST10:  SCOPE
        JSR      R5,      PUSH4      ;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
        44                          ;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
        45                          ;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
        46                          ;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
        47                          ;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.
        50                          ;THE ERROR NUMBER IS 50

```


00100 (VKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 17
 00200 (VKACC.P'1 16-AUG-78 08:41 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 56 ;PS NOT EQUAL TO 200
 ;THE ERROR NUMBER IS 56
 CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R4)
 BEQ .+6 ;BRANCH IF OK
 HLT 57 ;STACK POINTER (R4) NOT EQUAL TO #STACK4
 ;THE ERROR NUMBER IS 57
 CMP #000200,ANS1 ;CHECK FIRST HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 60 ;ANS1 NOT EQUAL TO 000200
 ;THE ERROR NUMBER IS 60
 TST ANS2 ;CHECK SECOND HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 61 ;ANS2 NOT EQUAL TO 000000
 ;THE ERROR NUMBER IS 61
 END12: CMPB #12, \$TESTN ;CHECK THE TEST NUMBER
 BEQ .+6 ;BRANCH IF OK
 HLT 62 ;WRONG TEST! PC MUST HAVE FOULED UP.
 ;THE ERROR NUMBER IS 62

00300 643
 00400 644
 00500 645
 00600 646
 00700 647
 00800 648
 00900 649
 01000 650 002426 104400
 01100 651 002430 004567 013552
 01200 652 002434 100252 125252
 01300 653 002440 000425 052525
 01400 654 002444 000217
 01500 655 002446 016456 000340
 01600 656 002452 012704 000510
 01700 657
 01800 658 002456 000240
 01900 659 002460 075004
 02000 660
 02100 661 002462 004767 013552
 02200 662 002466 010467 175742
 02300 663 002472 122767 000200 175732
 02400 664 002500 001402
 02500 665 002502 104000
 02600 666 002504 000056
 02700 667
 02800 668 002506 022767 000514 175720
 02900 669 002514 001402
 03000 670 002516 104000
 03100 671 002520 000057
 03200 672
 03300 673 002522 022767 000200 175706
 03400 674 002530 001402
 03500 675 002532 104002
 03600 676 002534 000060
 03700 677
 03800 678 002536 005767 175676
 03900 679 002542 001402
 04000 680 002544 104002
 04100 681 002546 000061
 04200 682
 04300 683 002550 122767 000012 175626
 04400 684 002556 001402
 04500 685 002560 104000
 04600 686 002562 000062
 04700 687
 04800 688
 04900

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 18
 00200 CVKACC.P11 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0026

00300 689
 00400 690
 00500 691
 00600 692
 00700 693
 00800 694
 00900 695
 01000 696 002564 104400
 01100 697 002566 004567 013242
 01200 698 002572 000252 125252
 01300 699 002576 100425 052525
 01400 700 002602 000307
 01500 701 002604 016456 000340
 01600 702
 01700 703 002610 000240
 01800 704 002612 075006
 01900 705
 02000 706 002614 004767 013254
 02100 707 002620 022706 000600
 02200 708 002624 001405
 02300 709 002626 012706 000600
 02400 710 002632 104000
 02500 711 002634 000063
 02600 712 002636 000421
 02700 713
 02800 714 002640 122767 000210 175564
 02900 715 002646 001402
 03000 716 002650 104000
 03100 717 002652 000064
 03200 718
 03300 719 002654 022767 100200 175554
 03400 720 002662 001402
 03500 721 002664 104002
 03600 722 002666 000065
 03700 723
 03800 724 002670 005767 175544
 03900 725 002674 001402
 04000 726 002676 104002
 04100 727 002700 000066
 04200 728
 04300 729 002702 122767 000013 175474
 04400 730 002710 001402
 04500 731 002712 104000
 04600 732 002714 000067
 04700 733
 04800 734
 04900

 :TEST 13: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 : 100425,052525 + 000252,125252 = 100200,000000
 : PS = 210, STACK POINTER = SP
 :*****

SCOPE
 TST13: JSR R5, PUSH4 :PUSH 4 WORDS ONTO STACK, SET PRIORI
 .WORD 000252,125252 :SECOND OPERAND ON TOP
 .WORD 100425,052525 :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 TSA13 :BRANCH IF OK
 MOV #BEGIN, SP :RESTORE STACK POINTER
 HLT :STACK POINTER FOULED UP
 63 :THE ERROR NUMBER IS 63
 BR END13 :SKIP REST OF TEST

TSA13: CMPB #210, \$PSW :CHECK PS (EXCEPT T BIT)
 BEQ .+6 :BRANCH IF OK
 HLT :PS NOT EQUAL TO 210
 64 :THE ERROR NUMBER IS 64

CMP #100200,ANS1 :CHECK FIRST HALF OF ANSWER
 BEQ .+6 :BRANCH IF OK
 HLT+2 :ANS1 NOT EQUAL TO 100200
 65 :THE ERROR NUMBER IS 65

TST ANS2 :CHECK SECOND HALF OF ANSWER
 BEQ .+6 :BRANCH IF OK
 HLT+2 :ANS2 NOT EQUAL TO 000000
 66 :THE ERROR NUMBER IS 66

END13: CMPB #13, \$TESTN :CHECK THE TEST NUMBER
 BEQ .+6 :BRANCH IF OK
 HLT :WRONG TEST! PC MUST HAVE FOULED UP.
 67 :THE ERROR NUMBER IS 67.

(VKA000 MAR 11 30A(1052) 21-AUG-78 15:28 PAGE 19
 (VKA000.P1 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0027

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

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

SCOPE
 TST14: JSR R5 PUSH4 :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

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 20
 CVKACC.P1 16-AUG-78 08:41 FADD TEST SECTION

SEQ 0028

```

00100 781
00200 782
00300 783
00400 784
00500 785
00600 786
00700 787
00800 788
00900 789
01000 790
01100 791
01200 792
01300 793
01400 794
01500 795
01600 796
01700 797
01800 798
01900 799
02000 800
02100 801
02200 802
02300 803
02400 804
02500 805
02600 806
02700 807
02800 808
02900 809
03000 810
03100 811
03200 812
03300 813
03400 814
03500 815
03600 816
03700 817
03800 818
03900 819
04000 820
04100 821
04200 822
04300 823
04400 824
04500 825
04600 826
04700
04800
04900

003052 104400
003054 004567 013126
003060 177652 125252
003064 177452 125252
003070 000357
003072 016456 000340
003076 012704 000510
003102 000240
003104 075004
003106 004767 013126
003112 010467 175316
003116 122767 000210 175306
003124 001402
003126 104000
003130 000075
003132 022767 000514 175274
003140 001402
003142 104000
003144 000076
003146 022767 177777 175262
003154 001402
003156 104002
003160 000077
003162 022767 177777 175250
003170 001402
003172 104002
003174 000100
003176 122767 000015 175200
003204 001402
003206 104000
003210 000101

```

 :TEST 15: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 : 177452,125252 + 177652,125252 = 177777,177777
 : PS = 210, STACK POINTER R4
 :*****

TST15: SCOPE
 JSR R5, PUSH4 ;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
 75 ;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
 76 ;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
 77 ;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
 100 ;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.
 101 ;THE ERROR NUMBER IS 101

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

SEQ 0029

```

00300      827
00400      828
00500      829
00600      830
00700      831
00800      832
00900      833
01000      834
01100      835
01200      836
01300      837
01400      838
01500      839
01600      840
01700      841
01800      842
01900      843
02000      844
02100      845
02200      846
02300      847
02400      848
02500      849
02600      850
02700      851
02800      852
02900      853
03000      854
03100      855
03200      856
03300      857
03400      858
03500      859
03600      860
03700      861
03800      862
03900      863
04000      864
04100      865
04200      866
04300      867
04400      868
04500      869
04600      870
04700      871
04800      872
04900      873
05000      874
05100      875
05200      876
05300      877
05400      878
05500      879
05600      880
05700      881
05800      882
05900      883
  
```

003212 104400
 003214 004567 013140
 003220 003244
 003222 104000 104000
 003226 004000 105004
 003232 000144
 003234 016456 000340
 003240 000240
 003242 075007
 003244 104000
 003246 104000
 003250 004000
 003252 105004
 003254 004767 013130
 003260 105767 175146
 003264 001402
 003266 104000
 003270 000102
 003272 022767 104000 175136
 003300 001402
 003302 104002
 003304 000103
 003306 022767 104000 175124
 003314 001402
 003316 104002
 003320 000104
 003322 022767 000401 175112
 003330 001402
 003332 104004
 003334 000105
 003336 005767 175102
 003342 001402
 003344 104004
 003346 000106
 003350 122767 000016 175026
 003356 001402
 003360 104000
 003362 000107

```

*****
:TEST 16: FADD (LSI-11 FLOATING ADD INSTRUCTION)
: 004000,105004 + 104000,104000 000401,000000
: PS = 000, STACK POINTER PC
*****

TST16: SCOPE
JSR R5, PUSH7 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
;TOP OF STACK
;SECOND OPERAND ON TOP
;FIRST OPERAND ON BOTTOM
;PROCESSOR PRIORITY LEVEL
;FIS TRAP VECTOR

NOP
FADD PC ;FLOATING ADD ON FOLLOWING 4 WORDS
STK16: 104000 ;SHOULD CONTAIN 104000
104000 ;SHOULD CONTAIN 104000
004000 ;BEFORE FADD, 004000; AFTER, 000401
105004 ;BEFORE FADD, 105004; AFTER, 000000

JSR PC, POP7 ;POP THE ANSWER
TSTB $PSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 000
102 ;THE ERROR NUMBER IS 102

CMP #104000,ANS1 ;CHECK FIRST HALF OF INPUT DATA (STK16)
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 104000
103 ;THE ERROR NUMBER IS 103

CMP #104000,ANS2 ;CHECK SECOND HALF OF INPUT DATA (STK16+2)
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 104000
104 ;THE ERROR NUMBER IS 104

CMP #000401,ANS3 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+4 ;ANS3 NOT EQUAL TO 000401
105 ;THE ERROR NUMBER IS 105

TST ANS4 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+4 ;ANS4 NOT EQUAL TO 000000
106 ;THE ERROR NUMBER IS 106

END16: CMPB #16, $TESTN ;CHECK THE TEST NUMBER
BEQ .+6 ;BRANCH IF OK
HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
107 ;THE ERROR NUMBER IS 107
  
```

CVKACC MAR 11 30A(1052) 21-AUG-78 15:28 PAGE 22
 CVKACC.P 11 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW

SEQ 0030

```

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,      PUSH
        .WORD    000377,177777
        .WORD    100200,000000
        .WORD    157
        .WORD    ISR17, 000
        MOV      #STACK0,R3
        :PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
        :SECOND OPERAND ON TOP
        :FIRST OPERAND ON BOTTOM
        :PROCESSOR PRIORITY LEVEL
        :FIS TRAP VECTOR
        :SET UP R3 AS STACK POINTER

        NOP
        FADD     R3
        :FLOATING ADD ON THE R3 STACK

RTA17:  JSR      PC,      POPR
        MOV      R3,      $SP
        HLT+2
        110
        BR
        END17

ISR17:  JSR      PC,      POPR
        MOV      R3,      $SP
        TSTB     $PSW
        BEQ      .+6
        HLT
        111
        :POP THE 'ANSWER'
        :SAVE STACK POINTER (R3)
        :FIS TRAP DIDN'T OCCURE!
        :THE ERROR NUMBER IS 110

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

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

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

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

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

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


CVKACC MARV11 30A(1052) 21-AUG-78 15:28 PAGE 24
 CVKACC.P11 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW

SEQ 0032

00100 953
 00200 954
 00300 955
 00400 956
 00500 957
 00600 958
 00700 959
 00800 960 003616 104400
 00900 961 003620 004567 012210
 01000 962 003624 100377 177777
 01100 963 003630 000200 000000
 01200 964 003634 000257
 01300 965 003636 003664 000340
 01400 966
 01500 967 003642 000240
 01600 968 003644 075006
 01700 969
 01800 970 003646 004767 012222
 01900 971 003652 104002
 02000 972 003654 000122
 02100 973 003656 012706 000600
 02200 974 003662 000463
 02300 975
 02400 976 003664 004767 012236
 02500 977 003670 022706 000600
 02600 978 003674 001405
 02700 979 003676 012706 000600
 02800 980 003702 104000
 02900 981 003704 000123
 03000 982 003706 000451
 03100 983
 03200 984 003710 122767 000340 174514
 03300 985 003716 001402
 03400 986 003720 104000
 03500 987 003722 000124
 03600 988
 03700 989 003724 022767 003646 174504
 03800 990 003732 001402
 03900 991 003734 104001
 04000 992 003736 000125
 04100 993
 04200 994 003740 022767 000212 174472
 04300 995 003746 001402
 04400 996 003750 104002
 04500 997 003752 000126
 04600 998
 04700 999 003754 022767 100377 174460
 04800 1000 003762 001402
 04900 1001 003764 104004
 05000 1002 003766 000127
 05100 1003
 05200 1004 003770 022767 177777 174440
 05300 1005 003776 001402
 05400 1006 004000 104004
 05500 1007 004002 000130
 05600 1008

 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 (VKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 26

00200 (VKACC.P11 16-AUG-78 08:41

TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0034

```

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 000141

```

```

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

TST21: SCOPE
JSR R5, PUSHR ;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

```

00100 (VKACC MACV11 30A(1052) 21-AUG-78 15:28 PAGE 27

00200 (VKACC.P11 16-AUG-78 08:41

TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0035

00300 1081 004240 022767 177452 174200

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

00400 1082 004246 001402

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

00500 1083 004250 104006

00600 1084 004252 000143

00700 1085

00800 1086 004254 022767 125253 174166

00900 1087 004262 001402

01000 1088 004264 104006

01100 1089 004266 000144

01200 1090

01300 1091 004270 122767 000021 174106

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

01400 1092 004276 001402

01500 1093 004300 104000

01600 1094 004302 000145

01700 1095

01800 1096

01900

00100 (VKACC, MA'Y11 30A(1052) 21-AUG-78 15:28 PAGE 28
 00200 (VKACC, P'1 16-AUG-78 08:41 TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0036

 TEST 22: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 077652,125253 + 077452,125252 ==> OVERFLOW
 PS(ON STACK) = 002, STACK POINTER SP

TST22: SCOPE
 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 147 ;STACK POINTER FOULED UP
 BR END22 ;THE ERROR NUMBER IS 147
 ;SKIP REST OF TEST

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

00300 1097
 00400 1098
 00500 1099
 00600 1100
 00700 1101
 00800 1102
 00900 1103
 01000 1104
 01100 1105 004304 104400
 01200 1106 004306 004567 011522
 01300 1107 004312 077452 125252
 01400 1108 004316 077652 125253
 01500 1109 004322 000003
 01600 1110 004324 004352 000344
 01700 1111
 01800 1112 004330 000240
 01900 1113 004332 075006
 02000 1114
 02100 1115 004334 004767 011534
 02200 1116 004340 104002
 02300 1117 004342 000146
 02400 1118 004344 012706 000600
 02500 1119 004350 000464
 02600 1120
 02700 1121 004352 004767 011550
 02800 1122 004356 022706 000600
 02900 1123 004362 001405
 03000 1124 004364 012706 000600
 03100 1125 004370 104000
 03200 1126 004372 000147
 03300 1127 004374 000452
 03400 1128
 03500 1129 004376 122767 000344 174026
 03600 1130 004404 001402
 03700 1131 004406 104000
 03800 1132 004410 000150
 03900 1133
 04000 1134 004412 022767 004334 174026
 04100 1135 004420 001402
 04200 1136 004422 104001
 04300 1137 004424 000151
 04400 1138
 04500 1139 004426 022767 000002 174004
 04600 1140 004434 001402
 04700 1141 004436 104002
 04800 1142 004440 000152
 04900 1143
 05000 1144 004442 022767 077452 173772
 05100 1145 004450 001402
 05200 1146 004452 104004
 05300 1147 004454 000153
 05400 1148
 05500 1149 004456 022767 125252 173760
 05600 1150 004464 001402
 05700 1151 004466 104004
 05800 1152 004470 000154
 05900 1153

00100	CVKACC	MACY1	30A(1052)	21-AUG-78	15:28	PAGE 29		
00200	CVKACC	P11	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							

00100 CVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 30
 00200 CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

SEQ 0038

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

```

*****
: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      160              ;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      161              ;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    162              ;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    163              ;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      164              ;WRONG TEST! PC MUST HAVE FOULED UP.
        164                      ;THE ERROR NUMBER IS 164

```


00100 CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 32
 00200 CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

00300 1261
 00400 1262
 00500 1263
 00600 1264
 00700 1265
 00800 1266
 00900 1267
 01000 1268 005032 104400
 01100 1269 005034 004567 010774
 01200 1270 005040 100177 177777
 01300 1271 005044 002460 123456
 01400 1272 005050 000015
 01500 1273 005052 016456 000340
 01600 1274
 01700 1275 005056 000240
 01800 1276 005060 075016
 01900 1277
 02000 1278 005062 004767 011006
 02100 1279 005066 022706 000600
 02200 1280 005072 001405
 02300 1281 005074 012706 000600
 02400 1282
 02500 1283
 02600 1284
 02700 1285
 02800 1286
 02900 1287
 03000 1288 005100 104000
 03100 1289 005102 000172
 03200 1290 005104 000422
 03300 1291
 03400 1292 005106 122767 000000 173316
 03500 1293 005114 001402
 03600 1294 005116 104000
 03700 1295 005120 000173
 03800 1296
 03900 1297 005122 022767 002460 173306
 04000 1298 005130 001402
 04100 1299 005132 104002
 04200 1300 005134 000174
 04300 1301
 04400 1302 005136 022767 123456 173274
 04500 1303 005144 001402
 04600 1304 005146 104002
 04700 1305 005150 000175
 04800 1306
 04900 1307 005152 122767 000025 173224
 05000 1308 005160 001402
 05100 1309 005162 104000
 05200 1310 005164 000176
 05300 1311
 05400 1312
 05500 1313

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

TST25: SCOPE
 JSR R5, PUSH5 ;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
 172
 BR END25 ;STACK POINTER FOULED UP
 ;THE ERROR NUMBER IS 172
 ;SKIP REST OF TEST

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

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

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

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

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 33
 00200 CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

SEQ 0041

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 177 ;PS NOT EQUAL TO 200
 ;THE ERROR NUMBER IS 177

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

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

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

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

00100 CVKACC MAC(Y11 30A(1052) 21-AUG-78 15:28 PAGE 34
 00200 CVKACC.P1 16-AUG-78 08:41 FSUB TEST SECTION

00300 1359
 00400 1360
 00500 1361
 00600 1362
 00700 1363
 00800 1364
 00900 1365
 01000 1366 005324 104400
 01100 1367 005326 004567 010502
 01200 1368 005332 077652 125252
 01300 1369 005336 177452 125252
 01400 1370 005342 000357
 01500 1371 005344 016456 000340
 01600 1372
 01700 1373 005350 000240
 01800 1374 005352 075016
 01900 1375
 02000 1376 005354 004767 010514
 02100 1377 005360 022706 000600
 02200 1378 005364 001405
 02300 1379 005366 012706 000600
 02400 1380
 02500 1381
 02600 1382
 02700 1383
 02800 1384
 02900 1385
 03000 1386 005372 104000
 03100 1387 005374 000204
 03200 1388 005376 000422
 03300 1389
 03400 1390 005400 122767 000210 173024
 03500 1391 005406 001402
 03600 1392 005410 104000
 03700 1393 005412 000205
 03800 1394
 03900 1395 005414 022767 177777 173014
 04000 1396 005422 001402
 04100 1397 005424 104002
 04200 1398 005426 000206
 04300 1399
 04400 1400 005430 022767 177777 173002
 04500 1401 005436 001402
 04600 1402 005440 104002
 04700 1403 005442 000207
 04800 1404
 04900 1405 005444 122767 000027 172732
 05000 1406 005452 001402
 05100 1407 005454 104000
 05200 1408 005456 000210
 05300 1409
 05400 1410
 05500 1411

 TEST 27: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
 177452,125252 - 077652,125252 = 177777,177777
 PS = 210, STACK POINTER SP

TST27: SCOPE
 JSR R5, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 .WORD 077652,125252 ;SECOND OPERAND ON TOP
 .WORD 177452,125252 ;FIRST OPERAND ON BOTTOM
 .WORD 357 ;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 TSA27 ;BRANCH IF OK
 MOV #BEGIN, SP ;RESTORE STACK POINTER

HLT
 204
 BR END27 ;STACK POINTER FOULED UP
 ;THE ERROR NUMBER IS 204
 ;SKIP REST OF TEST

TSA27: CMPB #210, \$PSW ;CHECK PS (EXCEPT T BIT)
 BEQ .+6 ;BRANCH IF OK
 HLT 205 ;PS NOT EQUAL TO 210
 205 ;THE ERROR NUMBER IS 205

CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 ;ANS1 NOT EQUAL TO 177777
 206 ;THE ERROR NUMBER IS 206

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 ;ANS2 NOT EQUAL TO 177777
 207 ;THE ERROR NUMBER IS 207

END27: CMPB #27, \$TESTN ;CHECK THE TEST NUMBER
 BEQ .+6 ;BRANCH IF OK
 HLT ;WRONG TEST, PC MUST HAVE FOULED UP.
 210 ;THE ERROR NUMBER IS 210

CVKACC	MA(Y11	30A(1052)	21-AUG-78	1
CVKACC.P11			16-AUG-78 08:41	
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,      PUSH4      ;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

        NOP
        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      .                ;PS NOT EQUAL TO 010
        211                ;THE ERROR NUMBER IS 211

        CMP      #STACK4,$SP        ;CHECK THE STACK POINTER (R3)
        BEQ      .+6               ;BRANCH IF OK
        HLT      .                ;STACK POINTER (R3) NOT EQUAL TO #STACK4
        212                ;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
        213                ;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
        214                ;THE ERROR NUMBER IS 214

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

```

```

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

TST31:  SCOPE
        JSR      R5,      PUSH4      ;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      216              ;PS NOT EQUAL TO 210
        ;THE ERROR NUMBER IS 216

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

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

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

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

```

00300 1503
 00400 1504
 00500 1505
 00600 1506
 00700 1507
 00800 1508
 00900 1509
 01000 1510
 01100 1511 005760 104400
 01200 1512 005762 004567 010220
 01300 1513 005766 143325 052525
 01400 1514 005772 135152 125252
 01500 1515 005776 000357
 01600 1516 006000 016456 000340
 01700 1517 006004 012705 000510
 01800 1518
 01900 1519 006010 000240
 02000 1520 006012 075015
 02100 1521 006014 004767 010220
 02200 1522 006020 010567 172410
 02300 1523 006024 122767 000200 172400
 02400 1524 006032 001402
 02500 1525 006034 104000
 02600 1526 006036 000223
 02700 1527
 02800 1528 006040 022767 000514 172366
 02900 1529 006046 001402
 03000 1530 006050 104000
 03100 1531 006052 000224
 03200 1532
 03300 1533 006054 022767 043325 172354
 03400 1534 006062 001402
 03500 1535 006064 104002
 03600 1536 006066 000225
 03700 1537
 03800 1538 006070 022767 052525 172342
 03900 1539 006076 001402
 04000 1540 006100 104002
 04100 1541 006102 000226
 04200 1542
 04300 1543 006104 122767 000032 172272
 04400 1544 006112 001402
 04500 1545 006114 104000
 04600 1546 006116 000227
 04700 1547
 04800 1548
 04900

```

*****
: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
  
```

(VKACC MA(V'1 30A(1052) 21-AUG-78 15:28 PAGE 38
(VKACC.P'1 16-AUG-78 08:41 FSUB TEST SECTION

Figure 1

TIME	DATA	TIME	DATA	TIME	DATA	TIME	DATA
00100	(VKACC MA(V'1	30A(1052)	21-AUG-78	15			
00200	(VKACC.P'1	16-AUG-78	08:41				
00300							
00400	1549						
00500	1550						
00600	1551						
00700	1552						
00800	1553						
00900	1554						
01000	1555						
01100	1556	006120	104400				
01200	1557	006122	004567	010060			
01300	1558	006126	035152	125252			
01400	1559	006132	043125	052525			
01500	1560	006136	000040				
01600	1561	006140	016456	000340			
01700	1562	006144	012702	000510			
01800	1563						
01900	1564	006150	000240				
02000	1565	006152	075012				
02100	1566						
02200	1567	006154	004767	010060			
02300	1568	006160	010267	172250			
02400	1569	006164	105767	172242			
02500	1570	006170	001402				
02600	1571	006172	104000				
02700	1572	006174	000230				
02800	1573						
02900	1574	006176	022767	000514	172230		
03000	1575	006204	001402				
03100	1576	006206	104000				
03200	1577	006210	000231				
03300	1578						
03400	1579	006212	022767	043125	172216		
03500	1580	006220	001402				
03600	1581	006222	104002				
03700	1582	006224	000232				
03800	1583						
03900	1584	006226	022767	052524	172204		
04000	1585	006234	001402				
04100	1586	006236	104002				
04200	1587	006240	000233				
04300	1588						
04400	1589	006242	122767	000033	172134		
04500	1590	006250	001402				
04600	1591	006252	104000				
04700	1592	006254	000234				
04800	1593						
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,      PUSH4      ;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 CVKACC MACV 11 30A(1052) 21-AUG-78 15:28 PAGE 39
00200 CVKACC.P 11 16-AUG-78 08:41 FSUB TEST SECTION

SEQ 0047

CVKACC	MACY11	304(1052)	21-AUG-78	1
CVKACC.P11		16-AUG-78	08:41	
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
*****

```

```

SCOPE
TST34: JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
        .WORD    000000,0C0000      ;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    RO      ;FLOATING SUBTRACT ON THE RO STACK

```

```

JSR      PC,      POPR      ;POP THE ANSWER
MOV      RO,      $SP      ;SAVE 'STACK POINTER'
CMPB     #204,     $PSW     ;CHECK PS (EXCEPT T BIT)
BFG      .+6        ;BRANCH IF OK
HLT
235      ;PS NOT EQUAL TO 204
          ;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    R34, $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 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 10
 00200 CVKACC.P11 16-AUG-78 08:41 FSUB TEST SECTION

SEQ 0048

00300 1641
 00400 1642
 00500 1643
 00600 1644
 00700 1645
 00800 1646
 00900 1647
 01000 1648
 01100 1649 006412 104400
 01200 1650 006414 004567 007566
 01300 1651 006420 177777 177777
 01400 1652 006424 000000 000000
 01500 1653 006430 000100
 01600 1654 006432 016456 000340
 01700 1655 006436 012702 000510
 01800 1656
 01900 1657 006442 000240
 02000 1658 006444 075012
 02100 1659 006446 004767 007566
 02200 1660 006452 010267 171756
 02300 1661 006456 105767 171750
 02400 1662 006462 001402
 02500 1663 006464 104000
 02600 1664 006466 000242
 02700 1665
 02800 1666 006470 022767 000514 171736
 02900 1667 006476 001402
 03000 1668 006500 104000
 03100 1669 006502 000243
 03200 1670
 03300 1671 006504 022767 077777 171724
 03400 1672 006512 001402
 03500 1673 006514 104002
 03600 1674 006516 000244
 03700 1675
 03800 1676 006520 022767 177777 171712
 03900 1677 006526 001402
 04000 1678 006530 104002
 04100 1679 006532 000245
 04200 1680
 04300 1681 006534 122767 000035 171642
 04400 1682 006542 001402
 04500 1683 006544 104000
 04600 1684 006546 000246
 04700 1685
 04800 1686
 04900 1687

 :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

CVK100 (VKACC, MARV1 30A(1052) 21-AUG-78 15:28 PAGE 41
 00200 (VKACC, P11 16-AUG-78 08:41 FSUB TEST SECTION

SEQ 0049

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 006616 104000
 03100 1715 006620 000247
 03200 1716 006622 000422
 03300 1717
 03400 1718 006624 122767 000210 171600
 03500 1719 006632 001402
 03600 1720 006634 104000
 03700 1721 006636 000250
 03800 1722
 03900 1723 006640 022767 177777 171570
 04000 1724 006646 001402
 04100 1725 006650 104002
 04200 1726 006652 000251
 04300 1727
 04400 1728 006654 022767 177777 171556
 04500 1729 006662 001402
 04600 1730 006664 104002
 04700 1731 006666 000252
 04800 1732
 04900 1733 006670 122767 000036 171506
 05000 1734 006676 001402
 05100 1735 006700 104000
 05200 1736 006702 000253
 05300 1737
 05400 1738
 05500 1739

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

TST36: SCOPE
 JSR R5, PUSH5 ; 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
 247
 BR END36 ; STACK POINTER FOULED UP
 ; THE ERROR NUMBER IS 247
 ; SKIP REST OF TEST

TSA36: CMPB #210, \$PSW ; CHECK PS (EXCEPT T BIT)
 BEQ .+6 ; BRANCH IF OK
 HLT 250 ; PS NOT EQUAL TO 210
 ; 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 253 ; WRONG TEST! PC MUST HAVE FOULED UP.
 ; 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

SEQ 0050

```

00300 1739
00400 1740
00500 1741
00600 1742
00700 1743
00800 1744
00900 1745
01000 1746
01100 1747 006704 104400
01200 1748 006706 004567 007122
01300 1749 006712 000252 125253
01400 1750 006716 000425 052525
01500 1751 006722 000257
01600 1752 006724 006752 000340
01700 1753
01800 1754 006730 000240
01900 1755 006732 075016
02000 1756
02100 1757 006734 004767 007134
02200 1758 006740 104002
02300 1759 006742 000254
02400 1760 006744 012706 000600
02500 1761 006750 000464
02600 1762
02700 1763 006752 004767 007150
02800 1764 006756 022706 000600
02900 1765 006762 001405
03000 1766 006764 012706 000600
03100 1767 006770 104000
03200 1768 006772 000255
03300 1769 006774 000452
03400 1770
03500 1771 006776 122767 000340 171426
03600 1772 007004 001402
03700 1773 007006 104000
03800 1774 007010 000256
03900 1775
04000 1776 007012 022767 006734 171416
04100 1777 007020 001402
04200 1778 007022 104001
04300 1779 007024 000257
04400 1780
04500 1781 007026 022767 000212 171404
04600 1782 007034 001402
04700 1783 007036 104002
04800 1784 007040 000260
04900 1785
05000 1786 007042 022767 000252 171372
05100 1787 007050 001402
05200 1788 007052 104004
05300 1789 007054 000261
05400 1790
05500 1791 007056 022767 125253 171360
05600 1792 007064 001402
05700 1793 007066 104004
05800 1794 007070 000262
05900 1795

```

 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

NOF
 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

PC	CVKACC	MACY11	30A(1052)	21-AUG-78	15:28	PAGE 43	TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW	SEQ 0051
00100	CVKACC.P11	16-AUG-78	08:41					
00200								
00300								
00400	1795	007072	022767	000425	171346		CMP #000425,ANS5 ;CHECK DATA FROM STACK	
00500	1796	007100	001402				BEQ .+6 ;BRANCH IF OK	
00600	1797	007102	104006				HLT+6 ;DATA ON STACK (000425) CHANGED	
00700	1798	007104	000263				263 ;THE ERROR NUMBER IS 263	
00800	1799							
00900	1800	007106	022767	052525	171334		CMP #052525,ANS6 ;CHECK DATA FROM STACK	
01000	1801	007114	001402				BEQ .+6 ;BRANCH IF OK	
01100	1802	007116	104006				HLT+6 ;DATA ON STACK (052525) CHANGED	
01200	1803	007120	000264				264 ;THE ERROR NUMBER IS 264	
01300	1804							
01400	1805	007122	122767	000037	171254	END37:	CMPB #37, \$TESTN ;CHECK THE TEST NUMBER	
01500	1806	007130	001402				BEQ .+6 ;BRANCH IF OK	
01600	1807	007132	104000				HLT ;WRONG TEST. PC MUST HAVE FOULED UP.	
01700	1808	007134	000265				265 ;THE ERROR NUMBER IS 265	
01800	1809							
01900	1810							

(0100 (VKACC MAC(Y11 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 1811
00400 1812
00500 1813
00600 1814
00700 1815
00800 1816
00900 1817
01000 1818
01100 1819 007136 104400
01200 1820 007140 004567 007042
01300 1821 007144 177452 125252
01400 1822 007150 077652 125253
01500 1823 007154 000015
01600 1824 007156 007210 000344
01700 1825 007162 012703 000510
01800 1826
01900 1827 007166 000240
02000 1828 007170 075013
02100 1829
02200 1830 007172 004767 007042
02300 1831 007176 010367 171232
02400 1832 007202 104002
02500 1833 007204 000266
02600 1834 007206 000464
02700 1835
02800 1836 007210 004767 007054
02900 1837 007214 010367 171214
03000 1838 007220 122767 000344 171204
03100 1839 007226 001402
03200 1840 007230 104000
03300 1841 007232 000267
03400 1842
03500 1843 007234 022767 000510 171172
03600 1844 007242 001402
03700 1845 007244 104000
03800 1846 007246 000270
03900 1847
04000 1848 007250 022767 007172 171160
04100 1849 007256 001402
04200 1850 007260 104001
04300 1851 007262 000271
04400 1852
04500 1853 007264 022767 000002 171146
04600 1854 007272 001402
04700 1855 007274 104002
04800 1856 007276 000272
04900 1857
05000 1858 007300 022767 177452 171134
05100 1859 007306 001402
05200 1860 007310 104004
05300 1861 007312 000273
05400 1862
05500 1863 007314 022767 125252 171122
05600 1864 007322 001402
05700 1865 007324 104004
05800 1866 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 MAR Y11 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 TES PC MUST HAVE FOULED UP.
01700      1880 007372 000277                      277             ;THE ERROR MBER IS 277
01800      1881
01900      1882

```

CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 46
CVKACC.P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0054

00400	1883			
00500	1884			
00600	1885			
00700	1886			
00800	1887			
00900	1888			
01000	1889			
01100	1890	007374	104400	
01200	1891	007376	004567	006604
01300	1892	007402	000000	000000
01400	1893	007406	000000	000000
01500	1894	007412	000111	
01600	1895	007414	016456	000340
01700	1896	007420	012704	000510
01800	1897			
01900	1898	007424	000240	
02000	1899	007426	075024	
02100	1900			
02200	1901	007430	004767	006604
02300	1902	007434	010467	170774
02400	1903	007440	122767	000004 170764
02500	1904	007446	001402	
02600	1905	007450	104000	
02700	1906	007452	000300	
02800	1907			
02900	1908	007454	022767	000514 170752
03000	1909	007462	001402	
03100	1910	007464	104000	
03200	1911	007466	000301	
03300	1912			
03400	1913	007470	005767	170742
03500	1914	007474	001402	
03600	1915	007476	104002	
03700	1916	007500	000302	
03800	1917			
03900	1918	007502	005767	170732
04000	1919	007506	001402	
04100	1920	007510	104002	
04200	1921	007512	000303	
04300	1922			
04400	1923	007514	122767	000041 170662
04500	1924	007522	001402	
04600	1925	007524	104000	
04700	1926	007526	000304	
04800	1927			
04900	1928			

```

*****
:TEST 41:      FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
:      000000,000000 * 000000,000000 = 000000,000000
:      PS = 004,      STACK POINTER = R4
:*****

TST41:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
        .WORD    000000,000000      ;SECOND OPERAND ON TOP
        .WORD    000000,000000      ;FIRST OPERAND ON BOTOM
        .WORD    111                ;PROCESSOR PRIORITY LEVEL
        .WORD    TRAPER, 340         ;FIS TRAP VECTOR
        MOV      #STACK0,R4         ;SET UP STACK POINTER

        NOP
        FMUL     R4                  ;FLOATING MULTIPLY ON THE R4 STACK

        JSR      PC,      POPR       ;POP THE ANSWER
        MOV      R4,      $SP        ;SAVE 'STACK POINTER'
        CMPB     #004,      $PSW     ;CHECK PS (EXCEPT T BIT)
        BEQ      .+6                ;BRANCH IF OK
        HLT      300                ;PS NOT EQUAL TO 004
        ;THE ERROR NUMBER IS 300

        CMP      #STACK4,$SP        ;CHECK THE STACK POINTER (R4)
        BEQ      .+6                ;BRANCH IF OK
        HLT      301                ;STACK POINTER (R4) NOT EQUAL TO #STACK4
        ;THE ERROR NUMBER IS 301

        TST      ANS1               ;CHECK FIRST HALF OF ANSWER
        BEQ      .+6                ;BRANCH IF OK
        HLT+2    302                ;ANS1 NOT EQUAL TO 000000
        ;THE ERROR NUMBER IS 302

        TST      ANS2               ;CHECK SECOND HALF OF ANSWER
        BEQ      .+6                ;BRANCH IF OK
        HLT+2    303                ;ANS2 NOT EQUAL TO 000000
        ;THE ERROR NUMBER IS 303

END41:  CMPB     #41,      $TESTN    ;CHECK THE TEST NUMBER
        BEQ      .+6                ;BRANCH IF OK
        HLT      304                ;WRONG TEST. PC MUST HAVE FOULED UP.
        ;THE ERROR NUMBER IS 304

```

00100 CVKACC MA(V11 30A(1052) 21-AUG-78 15:28 PAGE 47
00200 CVKACC.P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0055

00400	1929				
00500	1930				
00600	1931				
00700	1932				
00800	1933				
00900	1934				
01000	1935				
01100	1936	007530	104400		
01200	1937	007532	004567	006450	
01300	1938	007536	052345	123456	
01400	1939	007542	140200	000000	
01500	1940	007546	000343		
01600	1941	007550	016456	000340	
01700	1942	007554	012702	000510	
01800	1943				
01900	1944	007560	000240		
02000	1945	007562	075022		
02100	1946				
02200	1947	007564	004767	006450	
02300	1948	007570	010267	170640	
02400	1949	007574	122767	000210	170630
02500	1950	007602	001402		
02600	1951	007604	104000		
02700	1952	007606	000305		
02800	1953				
02900	1954	007610	022767	000514	170616
03000	1955	007616	001402		
03100	1956	007620	104000		
03200	1957	007622	000306		
03300	1958				
03400	1959	007624	022767	152345	170604
03500	1960	007632	001402		
03600	1961	007634	104002		
03700	1962	007636	000307		
03800	1963				
03900	1964	007640	022767	123456	170572
04000	1965	007646	001402		
04100	1966	007650	104002		
04200	1967	007652	000310		
04300	1968				
04400	1969	007654	122767	000042	170522
04500	1970	007662	001402		
04600	1971	007664	104000		
04700	1972	007666	000311		
04800	1973				
04900	1974				

```

*****
:TEST 42:      FMUL (LSI-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

```

CVKACC MAR11 30A(1052) 21-AUG-78 15:28 PAGE 48
 CVKACC.P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0056

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

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

 :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 312 ;PS NOT EQUAL TO 004
 ;THE ERROR NUMBER IS 312
 CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R5)
 BEQ .+6 ;BRANCH IF OK
 HLT 313 ;STACK POINTER (R5) NOT EQUAL TO #STACK4
 ;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
 ;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
 ;THE ERROR NUMBER IS 315
 END43: CMPB #43, \$TESTN ;CHECK THE TEST NUMBER
 BEQ .+6 ;BRANCH IF OK
 HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
 316 ;THE ERROR NUMBER IS 316

00100 CVKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 49
 00200 CVKACC.P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0057

```

00300 2021
00400 2022
00500 2023
00600 2024
00700 2025
00800 2026
00900 2027
01000 2028
01100 2029
01200 2030
01300 2031
01400 2032
01500 2033
01600 2034
01700 2035
01800 2036
01900 2037
02000 2038
02100 2039
02200 2040
02300 2041
02400 2042
02500 2043
02600 2044
02700 2045
02800 2046
02900 2047
03000 2048
03100 2049
03200 2050
03300 2051
03400 2052
03500 2053
03600 2054
03700 2055
03800 2056
03900 2057
04000 2058
04100 2059
04200 2060
04300 2061
04400 2062
04500 2063
04600 2064
04700 2065
04800 2066
04900 2067
  
```

010024 104400
 010026 004567 006154
 010032 000052 125252
 010036 161616 161616
 010042 000217
 010044 016456 000340
 010050 012703 000510
 010054 000240
 010056 075023
 010060 004767 006154
 010064 010367 170344
 010070 122767 000204 170334
 010076 001402
 010100 104000
 010102 000317
 010104 022767 000514 170322
 010112 001402
 010114 104000
 010116 000320
 010120 005767 170312
 010124 001402
 010126 104002
 010130 000321
 010132 005767 170302
 010136 001402
 010140 104002
 010142 000322
 010144 122767 000044 170232
 010152 001402
 010154 104000
 010156 000323

```

*****
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 ;PS NOT EQUAL TO 204
        317 ;THE ERROR NUMBER IS 317

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

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

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

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

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 50
 CVKACC.P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0058

00100
 00200
 00300
 00400 2067
 00500 2068
 00600 2069
 00700 2070
 00800 2071
 00900 2072
 01000 2073
 01100 2074 010160 104400
 01200 2075 010162 004567 005646
 01300 2076 010166 041500 000000
 01400 2077 010172 176452 125252
 01500 2078 010176 000357
 01600 2079 010200 016456 000340
 01700 2080
 01800 2081 010204 000240
 01900 2082 010206 075026
 02000 2083
 02100 2084 010210 004767 005660
 02200 2085 010214 022706 000600
 02300 2086 010220 00405
 02400 2087 010222 012706 000600
 02500 2088 010226 104000
 02600 2089 010230 000324
 02700 2090 010232 000422
 02800 2091
 02900 2092 010234 122767 000210 170170
 03000 2093 010242 001402
 03100 2094 010244 104000
 03200 2095 010246 000325
 03300 2096
 03400 2097 010250 022767 177777 170160
 03500 2098 010256 001402
 03600 2099 010260 104002
 03700 2100 010262 000326
 03800 2101
 03900 2102 010264 022767 177777 170146
 04000 2103 010272 001402
 04100 2104 010274 104002
 04200 2105 010276 000327
 04300 2106
 04400 2107 010300 122767 000045 170076
 04500 2108 010306 001402
 04600 2109 010310 104000
 04700 2110 010312 000330
 04800
 04900

 :TEST 45: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
 : 176452,125252 * 041500,000000 = 177777,177777
 : PS = 210, STACK POINTER SP
 :*****

TST45: SCOPE
 JSR R5, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
 .WORD 041500,000000 ;SECOND OPERAND ON TOP
 .WORD 176452,125252 ;FIRST OPERAND ON BOTTOM
 .WORD 357 ;PROCESSOR PRIORITY LEVEL
 .WORD TRAPER, 340 ;FIS TRAP VECTOR

NOP
 FMUL SP ;FLOATING MULTIPLY ON THE STACK

JSR PC, POP5 ;POP THE ANSWER
 CMP #BEGIN, SP ;CHECK THE STACK POINTER
 BEQ TSA45 ;BRANCH IF OK
 MOV #BEGIN, SP ;RESTORE STACK POINTER
 HLT ;STACK POINTER FOULED UP
 324 ;THE ERROR NUMBER IS 324
 BR END45 ;SKIP REST OF TEST

TSA45: CMPB #210, \$PSW ;CHECK PS (EXCEPT T BIT)
 BEQ .+6 ;BRANCH IF OK
 HLT ;PS NOT EQUAL TO 210
 325 ;THE ERROR NUMBER IS 325

CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 ;ANS1 NOT EQUAL TO 177777
 326 ;THE ERROR NUMBER IS 326

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 ;ANS2 NOT EQUAL TO 177777
 327 ;THE ERROR NUMBER IS 327

END45: CMPB #45, \$TESTN ;CHECK THE TEST NUMBER
 BEQ .+6 ;BRANCH IF OK
 HLT ;WRONG TEST. PC MUST HAVE FOULED UP.
 330 ;THE ERROR NUMBER IS 330

00100 CVKAF MA Y11 30A(1052) 21-AUG-78 15:28 PAGE 51
 00200 (VKAF, P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0059

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 2127
 01900 2128 010344 000240
 02000 2129 010346 075021
 02100 2130
 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 2137
 02900 2138 010374 022767 000514 170032
 03000 2139 010402 001402
 03100 2140 010404 104000
 03200 2141 010406 000332
 03300 2142
 03400 2143 010410 022767 000200 170020
 03500 2144 010416 001402
 03600 2145 010420 104002
 03700 2146 010422 000333
 03800 2147
 03900 2148 010424 005767 170010
 04000 2149 010430 001402
 04100 2150 010432 104002
 04200 2151 010434 000334
 04300 2152
 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
 :*****
 :ST46: SCOPE
 JSR R5, PUSH4 :PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
 .WORD 114100,000001 :SECOND OPERAND ON TOP
 .WORD 124252,125252 :FIRST OPERAND ON BOTTOM
 .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 331 :PS NOT EQUAL TO 200
 :THE ERROR NUMBER IS 331
 CMP #STACK4,\$SP :CHECK THE STACK POINTER (R1)
 BEQ .+6 :BRANCH IF OK
 HLT 332 :STACK POINTER (R1) NOT EQUAL TO #STACK4
 :THE ERROR NUMBER IS 332
 CMP #000200,ANS1 :CHECK FIRST HALF OF ANSWER
 BEQ .+6 :BRANCH IF OK
 HLT+2 333 :ANS1 NOT EQUAL TO 000200
 :THE ERROR NUMBER IS 333
 TST ANS2 :CHECK SECOND HALF OF ANSWER
 BEQ .+6 :BRANCH IF OK
 HLT+2 334 :ANS2 NOT EQUAL TO 000000
 :THE ERROR NUMBER IS 334
 END46: CMPB #46, \$TESTN :CHECK THE TEST NUMBER
 BEQ .+6 :BRANCH IF OK
 HLT 335 :WRONG TEST: PC MUST HAVE FOULED JP.
 :THE ERROR NUMBER IS 335

00100 (VKA) MAR 11 30A(1052) 21-AUG-78 15:28 PAGE 52
 00200 (VKA) P11 16-AUG-78 08:41 FMUL TEST SECTION

SEQ 0060

00300 2159
 00400 2160
 00500 2161
 00600 2162
 00700 2163
 00800 2164
 00900 2165
 01000 2166
 01100 2167 010452 104400
 01200 2168 010454 004567 005700
 01300 2169 010460 010504
 01400 2170 010462 104000 104000
 01500 2171 010466 104000 105004
 01600 2172 010472 000252
 01700 2173 010474 016456 000340
 01800 2174
 01900 2175 010500 000240
 02000 2176 010502 075017
 02100 2177 010504 104000
 02200 2178 010506 104000
 02300 2179 010510 104000
 02400 2180 010512 105004
 02500 2181
 02600 2182 010514 004767 005670
 02700 2183 010520 122767 000210 167704
 02800 2184 010526 001402
 02900 2185 010530 104000
 03000 2186 010532 000336
 03100 2187
 03200 2188 010534 022767 104000 167674
 03300 2189 010542 001402
 03400 2190 010544 104002
 03500 2191 010546 000337
 03600 2192
 03700 2193 010550 022767 104000 167662
 03800 2194 010556 001402
 03900 2195 010560 104002
 04000 2196 010562 000340
 04100 2197
 04200 2198 010564 022767 100401 167650
 04300 2199 010572 001402
 04400 2200 010574 104004
 04500 2201 010576 000341
 04600 2202
 04700 2203 010600 005767 167640
 04800 2204 010604 001402
 04900 2205 010606 104004
 05000 2206 010610 000342
 05100 2207
 05200 2208 010612 122767 000047 167564
 05300 2209 010620 001402
 05400 2210 010622 104000
 05500 2211 010624 000343
 05600
 05700

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

TEST47: 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 336 ; PS NOT EQUAL TO 210
 ; THE ERROR NUMBER IS 336

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

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

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

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

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

00300 2213
 00400 2214
 00500 2215
 00600 2216
 00700 2217
 00800 2218
 00900 2219
 01000 2220 010626 104400
 01100 2221 010630 004567 005524
 01200 2222 010634 010660
 01300 2223 010636 104000 104000
 01400 2224 010642 134600 073601
 01500 2225 010646 000246
 01600 2226 010650 016456 000340
 01700 2227
 01800 2228 010654 000240
 01900 2229 010656 075027
 02000 2230 010660 104000
 02100 2231 010662 104000
 02200 2232 010664 134600
 02300 2233 010666 073601
 02400 2234
 02500 2235 010670 004767 005514
 02600 2236 010674 122767 000200 167530
 02700 2237 010702 001402
 02800 2238 010704 104000
 02900 2239 010706 000344
 03000 2240
 03100 2241 010710 022767 104000 167520
 03200 2242 010716 001402
 03300 2243 010720 104002
 03400 2244 010722 000345
 03500 2245
 03600 2246 010724 022767 104000 167506
 03700 2247 010732 001402
 03800 2248 010734 104002
 03900 2249 010736 000346
 04000 2250
 04100 2251 010740 022767 000401 167474
 04200 2252 010746 001402
 04300 2253 010750 104004
 04400 2254 010752 000347
 04500 2255
 04600 2256 010754 005767 167464
 04700 2257 010760 001402
 04800 2258 010762 104004
 04900 2259 010764 000350
 05000 2260
 05100 2261 010766 122767 000050 167410
 05200 2262 010774 001402
 05300 2263 010776 104000
 05400 2264 011000 000351
 05500 2265
 05600 2266
 05700 2267
 05800 2268
 05900 2269

 :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

STK50: NOP
 FMUL PC ;FLOATING MULTIPLY ON FOLLOWING 4 WORDS
 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 344 ;PS NOT EQUAL TO 200
 ;THE ERROR NUMBER IS 344

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

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

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

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

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

00100 (VKACC MARV11 30AK1052) 21-AUG-78 15:28 PAGE 54
 00200 (VKACC.P11 16-AUG-78 08:41) TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW

SEQ 0062

```

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 (LSI-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, POPR ;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 #RTA51,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 56
 00200 CVKACC 011 16-AUG-78 08:41 TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW

SEQ 0064

```

00300 2339
00400 2340
00500 2341
00600 2342
00700 2343
00800 2344
00900 2345
01000 2346
01100 2347
01200 2348
01300 2349
01400 2350
01500 2351
01600 2352
01700 2353
01800 2354
01900 2355
02000 2356
02100 2357
02200 2358
02300 2359
02400 2360
02500 2361
02600 2362
02700 2363
02800 2364
02900 2365
03000 2366
03100 2367
03200 2368
03300 2369
03400 2370
03500 2371
03600 2372
03700 2373
03800 2374
03900 2375
04000 2376
04100 2377
04200 2378
04300 2379
04400 2380
04500 2381
04600 2382
04700 2383
04800 2384
04900 2385
05000 2386
05100 2387
05200 2388
05300 2389
05400 2390
05500 2391
05600 2392
05700 2393
05800 2394
05900 2395

011236 104400
011240 004567 004570
011244 041500 000001
011250 076452 125252
011254 000105
011256 011304 000357

011262 000240
011264 075026

011266 004767 004602
011272 104002
011274 000364
011276 012706 000600
011302 000464

011304 004767 004616
011310 022706 000600
011314 001405
011316 012706 000600
011322 104000
011324 000365
011326 000452

011330 122767 000357 167074
011336 001402
011340 104000
011342 000366

011344 022767 011266 167064
011352 001402
011354 104001
011356 000367

011360 022767 000002 167052
011366 001402
011370 104002
011372 000370

011374 022767 041500 167040
011402 001402
011404 104004
011406 000371

011410 022767 000001 167026
011416 001402
011420 104004
011422 000372
  
```

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

TST52: SCOPE
 JSR R5, PUSHS ;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 ;FIS TRAP DIDN'T OCCURE!
 364 ;THE ERROR NUMBER IS 364
 MOV #BEGIN, SP ;RESTORE THE STACK POINTER
 BR END52

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
 365 ;THE ERROR NUMBER IS 365
 BR END52 ;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
 366 ;THE ERROR NUMBER IS 366

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

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

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

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

PC	CVKACC	MACY11	30A(1052)	21-AUG-78	15:28	PAGE 57	TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW	SEG
00200	CVKACC.P11			16-AUG-78	08:41			
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 ;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 ;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 ;WRONG TEST. PC MUST HAVE FOILED UP.	
01700	2408	011466	000375				375 ;THE ERROR NUMBER IS 375	
01800	2409							
01900	2410							

(X)100 CVKACC MAR Y11 30A(1052) 21-AUG-78 15:28 PAGE 58
 00200 CVKACC.P 16-AUG-78 08:41 FDIV TEST SECTION

SEQ 0066

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 376 ;PS NOT EQUAL TO 000
 ;THE ERROR NUMBER IS 376
 CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R0)
 BEQ .+6 ;BRANCH IF OK
 HLT 377 ;STACK POINTER (R0) NOT EQUAL TO #STACK4
 ;THE ERROR NUMBER IS 377
 CMP #077777,ANS1 ;CHECK FIRST HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 400 ;ANS1 NOT EQUAL TO 077777
 ;THE ERROR NUMBER IS 400
 CMP #177776,ANS2 ;CHECK SECOND HALF OF ANSWER
 BEQ .+6 ;BRANCH IF OK
 HLT+2 401 ;ANS2 NOT EQUAL TO 177776
 ;THE ERROR NUMBER IS 401
 END53: CMPB #53, \$TESTN ;CHECK THE TEST NUMBER
 BEQ .+6 ;BRANCH IF OK
 HLT 402 ;WRONG TEST. PC MUST HAVE FOULED UP.
 ;THE ERROR NUMBER IS 402

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 59
 00200 CVKACC.P11 16-AUG-78 08:41 FDIV TEST SECTION

SEQ 0067

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

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

TST54: SCOPE
 JSR R5, PUSH5 ;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

00100 CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 60
 00200 CVKACC.P11 16-AUG-78 08:41 FDIV TEST SECTION

SEQ 0068

```

00300 2503
00400 2504
00500 2505
00600 2506
00700 2507
00800 2508
00900 2509
01000 2510
01100 2511 011762 104400
01200 2512 011764 004567 004216
01300 2513 011770 065252 125252
01400 2514 011774 125252 125252
01500 2515 012000 000217
01600 2516 012002 016456 000340
01700 2517 012006 012702 000510
01800 2518
01900 2519 012012 000240
02000 2520 012014 075032
02100 2521
02200 2522 012016 004767 004216
02300 2523 012022 010267 166406
02400 2524 012026 122767 000210 166376
02500 2525 012034 001402
02600 2526 012036 104000
02700 2527 012040 000410
02800 2528
02900 2529 012042 022767 000514 166364
03000 2530 012050 001402
03100 2531 012052 104000
03200 2532 012054 000411
03300 2533
03400 2534 012056 022767 100200 166352
03500 2535 012064 001402
03600 2536 012066 104002
03700 2537 012070 000412
03800 2538
03900 2539 012072 005767 166342
04000 2540 012076 001402
04100 2541 012100 104002
04200 2542 012102 000413
04300 2543
04400 2544 012104 122767 000055 166272
04500 2545 012112 001402
04600 2546 012114 104000
04700 2547 012116 000414
04800 2548
04900 2549

```

 :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 CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 61
00200 CVKACC.P 16-AUG-78 08:41 FDIV TEST SECTION

SEO 0069

00400	2549			
00500	2550			
00600	2551			
00700	2552			
00800	2553			
00900	2554			
01000	2555			
01100	2556	012120	104400	
01200	2557	012122	004567	004060
01300	2558	012126	140670	123456
01400	2559	012132	000000	000000
01500	2560	012136	000105	
01600	2561	012140	016456	000340
01700	2562	012144	012703	000510
01800	2563			
01900	2564	012150	000240	
02000	2565	012152	075033	
02100	2566			
02200	2567	012154	004767	004060
02300	2568	012160	010367	166250
02400	2569	012164	122767	000004 166240
02500	2570	012172	001402	
02600	2571	012174	104000	
02700	2572	012176	000415	
02800	2573			
02900	2574	012200	022767	000514 166226
03000	2575	012206	001402	
03100	2576	012210	104000	
03200	2577	012212	000416	
03300	2578			
03400	2579	012214	005767	166216
03500	2580	012220	001402	
03600	2581	012222	104002	
03700	2582	012224	000417	
03800	2583			
03900	2584	012226	005767	166206
04000	2585	012232	001402	
04100	2586	012234	104002	
04200	2587	012236	000420	
04300	2588			
04400	2589	012240	122767	000056 166136
04500	2590	012246	001402	
04600	2591	012250	104000	
04700	2592	012252	000421	
04800	2593			
04900	2594			

```

*****
:TEST 56:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
:      000000,000000 / 140670,123456 = 000000,000000
:      PS 004,      STACK POINTER = R3
:*****

TST56:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
        .WORD    140670,123456      ;SECOND OPERAND ON TOP
        .WORD    000000,000000      ;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

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 62
 00200 CVKACC.P11 16-AUG-78 08:41 FDIV TEST SECTION

SEQ 0070

```

00300 2595
00400 2596
00500 2597
00600 2598
00700 2599
00800 2600
00900 2601
01000 2602
01100 2603
01200 2604
01300 2605
01400 2606
01500 2607
01600 2608
01700 2609
01800 2610
01900 2611
02000 2612
02100 2613
02200 2614
02300 2615
02400 2616
02500 2617
02600 2618
02700 2619
02800 2620
02900 2621
03000 2622
03100 2623
03200 2624
03300 2625
03400 2626
03500 2627
03600 2628
03700 2629
03800 2630
03900 2631
04000 2632
04100 2633
04200 2634
04300 2635
04400 2636
04500 2637
04600 2638
04700 2639
04800 2640
04900 2641
05000 2642
05100 2643
05200 2644
05300 2645
05400 2646
05500 2647
05600 2648
05700 2649
05800 2650
05900 2651
  
```

012254 104400
 012256 004567 004076
 012262 012306
 012264 104000 104000
 012270 102500 146000
 012274 000357
 012276 016456 000340
 012302 000240
 012304 075037
 012306 104000
 012310 104000
 012312 102500
 012314 146000
 012316 004767 004066
 012322 122767 000200 166102
 012330 001402
 012332 104000
 012334 000422
 012336 022767 104000 166072
 012344 001402
 012346 104002
 012350 000423
 012352 022767 104000 166060
 012360 001402
 012362 104002
 012364 000424
 012366 022767 036700 166046
 012374 001402
 012376 104004
 012400 000425
 012402 005767 166036
 012406 001402
 012410 104004
 012412 000426
 012414 122767 000357 165762
 012422 001402
 012424 104000
 012426 000427

```

*****
: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
;TOP OF STACK
;SECOND OPERAND ON TOP
;FIRST OPERAND ON BOTTOM
;PROCESSOR PRIORITY LEVEL
;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 ;ANS1 NOT EQUAL TO 104000
423 ;THE ERROR NUMBER IS 423

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

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

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

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

(C100 (VKA(C MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 63

(VKA(C.P 11 16-AUG-78 08:41

TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW

SEU 0071

```

00100      2651
00200      2652
00300      2653
00400      2654
00500      2655
00600      2656
00700      2657
00800      2658
00900      2659
01000      2660
01100      2661
01200      2662
01300      2663
01400      2664
01500      2665
01600      2666
01700      2667
01800      2668
01900      2669
02000      2670
02100      2671
02200      2672
02300      2673
02400      2674
02500      2675
02600      2676
02700      2677
02800      2678
02900      2679
03000      2680
03100      2681
03200      2682
03300      2683
03400      2684
03500      2685
03600      2686
03700      2687
03800      2688
03900      2689
04000      2690
04100      2691
04200      2692
04300      2693
04400      2694
04500      2695
04600      2696
04700      2697
04800      2698
04900      2699
05000      2700
05100      2701
05200      2702
05300      2703
05400      2704
05500      2705
05600      2706
05700
05800
05900

      012430 104400
      012432 004567 003550
      012436 065252 125252
      012442 025252 125251
      012446 000015
      012450 012502 000300
      012454 012701 000510

      012460 000240
      012462 075031

      012464 004767 003550
      012470 010167 165740
      012474 104002
      012476 000430
      012500 000464

      012502 004767 003562
      012506 010167 165722
      012512 122767 000300 165712
      012520 001402
      012522 104000
      012524 000431

      012526 022767 000510 165700
      012534 001402
      012536 104000
      012540 000432

      012542 022767 012464 165666
      012550 001402
      012552 104001
      012554 000433

      012556 022767 000012 165654
      012564 001402
      012566 104002
      012570 000434

      012572 022767 065252 165642
      012600 001402
      012602 104004
      012604 000435

      012606 022767 125252 165630
      012614 001402
      012616 104004
      012620 000436

      012622 022767 025252 165616
      012630 001402

```

```

;TEST 60:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
;      025252,125251 / 065252,125252 ==> UNDERFLOW
;      PS(ON STACK) = 012,      STACK POINTER = R1
;*****

TST60:  SCOPE
        JSR      R5,      PUSH
        .WORD    065252,125252 ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
        .WORD    025252,125251 ;SECOND OPERAND ON TOP
        .WORD    015          ;FIRST OPERAND ON BOTTOM
        .WORD    ISR60, 300    ;PROCESSOR PRIORITY LEVEL
        MOV      #STACK0,R1    ;FIS TRAP VECTOR
                                ;SET UP R1 AS STACK POINTER

        NOP
        FDIV     R1            ;FLOATING DIVIDE ON THE R1 STACK

RTA60:  JSR      PC,      POPR    ;POP THE 'ANSWER'
        MOV      R1,      $SP    ;SAVE STACK POINTER (R1)
        HLT+2    430            ;FIS TRAP DIDN'T OCCURE!
        BR       END60          ;THE ERROR NUMBER IS 430

ISR60:  JSR      PC,      POPR    ;POP ALL DATA OFF THE STACKS
        MOV      R1,      $SP    ;SAVE STACK POINTER (R1)
        CMPB     #300,      $PSW  ;CHECK PS AFTER FIS TRAP
        BEQ      .+6            ;BRANCH IF OK
                                ;PS AFTER FIS TRAP NOT EQUAL TO 300
                                ;THE ERROR NUMBER IS 431

        CMP      #STACK0,$SP    ;CHECK THE STACK POINTER (R1)
        BEQ      .+6            ;BRANCH IF OK
                                ;STACK POINTER (R1) NOT EQUAL TO #STACK0
                                ;THE ERROR NUMBER IS 432

        CMP      #RTA60,ANS1    ;CHECK FIS TRAP RETURN ADDRESS
        BEQ      .+6            ;BRANCH IF OK
                                ;FIS TRAP AT WRONG ADDRESS
                                ;THE ERROR NUMBER IS 433

        CMP      #012,ANS2      ;CHECK PS BEFORE FIS TRAP
        BEQ      .+6            ;BRANCH IF OK
                                ;PS AT FIS TRAP TIME NOT 012
                                ;THE ERROR NUMBER IS 434

        CMP      #065252,ANS3    ;CHECK DATA FROM THE STACK
        BEQ      .+6            ;BRANCH IF OK
                                ;DATA ON STACK (065252) CHANGED
                                ;THE ERROR NUMBER IS 435

        CMP      #125252,ANS4    ;CHECK DATA FROM STACK
        BEQ      .+6            ;BRANCH IF OK
                                ;DATA ON STACK (125252) CHANGED
                                ;THE ERROR NUMBER IS 436

        CMP      #025252,ANS5    ;CHECK DATA FROM STACK
        BEQ      .+6            ;BRANCH IF OK

```

Address	Instruction	Comments
00100	CVKACC MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 64	
00200	CVKACC.P11 16-AUG-78 08:41	
00300		
00400	2707 012632 104006,	HLT+6 :DATA ON STACK (025252) CHANGED
00500	2708 012634 000437	437 :THE ERROR NUMBER IS 437
00600	2709	
00700	2710 012636 022767 125251 165604	CMP #125251,ANS6 :CHECK DATA FROM STACK
00800	2711 012644 001402	BEQ .+6 :BRANCH IF OK
00900	2712 012646 104006	HLT+6 :DATA ON STACK (125251) CHANGED
01000	2713 012650 000440	440 :THE ERROR NUMBER IS 440
01100	2714	
01200	2715 012652 122767 000060 165524	END60: CMPB #60, \$TESTN :CHECK THE TEST NUMBER
01300	2716 012660 001402	BEQ .+6 :BRANCH IF OK
01400	2717 012662 104000	HLT :WRONG TEST. PC MUST HAVE FOULED UP.
01500	2718 012664 000441	441 :THE ERROR NUMBER IS 441
01600	2719	
01700	2720	

```

*****
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    442              ;FIS TRAP DIDN'T OCCURE.
        BR       END61            ;THE ERROR NUMBER IS 442

```

```

ISR61:  JSR      PC,      POPR      ;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      443              ;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      444              ;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    445              ;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    446              ;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    447              ;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    450              ;DATA ON STACK (125252) CHANGED
        450                      ;THE ERROR NUMBER IS 450

```

```

00100 2721
00200 2722
00300 2723
00400 2724
00500 2725
00600 2726
00700 2727
00800 2728 012666 104400
00900 2729 012670 004567 003312
01000 2730 012674 127652 125252
01100 2731 012700 067452 125252
01200 2732 012704 000242
01300 2733 012706 012740 000357
01400 2734 012712 012704 000510
01500 2735
01600 2736 012716 000240
01700 2737 012720 075034
01800 2738
01900 2739 012722 004767 003312
02000 2740 012726 010467 165502
02100 2741 012732 104002
02200 2742 012734 000442
02300 2743 012736 000464
02400 2744
02500 2745 012740 004767 003324
02600 2746 012744 010467 165464
02700 2747 012750 122767 000357 165454
02800 2748 012756 001402
02900 2749 012760 104000
03000 2750 012762 000443
03100 2751
03200 2752 012764 022767 000510 165442
03300 2753 012772 001402
03400 2754 012774 104000
03500 2755 012776 000444
03600 2756
03700 2757 013000 022767 012722 165430
03800 2758 013006 001402
03900 2759 013010 104001
04000 2760 013012 000445
04100 2761
04200 2762 013014 022767 000202 165416
04300 2763 013022 001402
04400 2764 013024 104002
04500 2765 013026 000446
04600 2766
04700 2767 013030 022767 127652 165404
04800 2768 013036 001402
04900 2769 013040 104004
05000 2770 013042 000447
05100 2771
05200 2772 013044 022767 125252 165332
05300 2773 013052 001402
05400 2774 013054 104004
05500 2775 013056 000450
05600
05700
05800
05900

```

00300						CMP	#067452,ANS5	:CHECK DATA FROM STACK
00400	2777	013060	022767	067452	165360	BEG	.+6	:BRANCH IF OK
00500	2778	013066	001402			HLT+6		:DATA ON STACK (067452) CHANGED
00600	2779	013070	104006			451		:THE ERROR NUMBER IS 451
00700	2780	013072	000451					
00800	2781							
00900	2782	013074	022767	125252	165346	CMP	#125252,ANS6	:CHECK DATA FROM STACK
01000	2783	013102	001402			BEG	.+6	:BRANCH IF OK
01100	2784	013104	104006			HLT+6		:DATA ON STACK (125252) CHANGED
01200	2785	013106	000452			452		:THE ERROR NUMBER IS 452
01300	2786							
01400	2787	013110	122767	000061	165266	END61:	CMPB #61, STESTN	:CHECK THE TEST NUMBER
01500	2788	013116	001402			BEG	.+6	:BRANCH IF OK
01600	2789	013120	104000			HLT		:WRONG TEST. PC MUST HAVE FOULED UP.
01700	2790	013122	000453			453		:THE ERROR NUMBER IS 453
01800	2791							
01900	2792							

CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 67
 CVKACC.P11 16-AUG-78 08:41

TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO

SEQ 0075

```

00100 2793
00200 2794
00300 2795
00400 2796
00500 2797
00600 2798
00700 2799
00800 2800
00900 2801
01000 2802
01100 2803
01200 2804
01300 2805
01400 2806
01500 2807
01600 2808
01700 2809
01800 2810
01900 2811
02000 2812
02100 2813
02200 2814
02300 2815
02400 2816
02500 2817
02600 2818
02700 2819
02800 2820
02900 2821
03000 2822
03100 2823
03200 2824
03300 2825
03400 2826
03500 2827
03600 2828
03700 2829
03800 2830
03900 2831
04000 2832
04100 2833
04200 2834
04300 2835
04400 2836
04500 2837
04600 2838
04700 2839
04800 2840
04900 2841
05000 2842
05100 2843
05200 2844
05300 2845
05400 2846
05500 2847
05600 2848
05700
05800
05900

013124 104400
013126 004567 003054
013132 100125 125252
013136 052525 052525
013142 000047
013144 013176 000113
013150 012705 000510

013154 000240
013156 075035

013160 004767 003054
013164 010567 165244
013170 104002
013172 000454
013174 000464

013176 004767 003066
013202 010567 165226
013206 122747 000113 165216
013214 001402
013216 104000
013220 000455

013222 022767 000510 165204
013230 001402
013232 104000
013234 000456

013236 022767 013160 165172
013244 001402
013246 104001
013250 000457

013252 022767 000013 165160
013260 001402
013262 104002
013264 000460

013266 022767 100125 165146
013274 001402
013276 104004
013300 000461

013302 022767 125252 165134
013310 001402
013312 104004
013314 000462
  
```

 :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, PUSH4 :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 :FIS TRAP DIDN'T OCCURE!
 454 :THE ERROR NUMBER IS 454
 BR END62

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 :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 :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 :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 :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 :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 :DATA ON STACK (125252) CHANGED
 462 :THE ERROR NUMBER IS 462

```
0C100 CVKACC MACY'1 30A(1052) 21-AUG-78 15:28 PAGE 68          SEC  
00200 CVKACC.P'1    16-AUG-78 08:41      TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO  
UO300  
00400       2849   013316   022767   052525   165122           CMP     #052525,ANS5        ;CHECK DATA FROM STACK  
00500       2850   C13324   001402           BEQ     .+6              ;BRANCH IF OK  
00600       2851   013326   104006           HLT+6             ;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             ;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                ;WRONG TEST! PC MUST HAVE FOULED UP.  
01700       2862   013360   000465           465            ;THE ERROR NUMBER IS 465  
01800       2863  
01900       2864
```

00100 (VKACC MAC Y11 30A(1052) 21-AUG-78 15:28 PAGE 69
 00200 (VKACC.P11 16-AUG-78 08:41 TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO

SEQ 0077

```

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

```

 :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 71
 00200 CVKACC.P11 16-AUG-78 08:41 TEST OF ALL FIS AT ONCE

SEQ 0079

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

 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 106400...C
 NOP
 F SUB R4 ;135252,125252-040616,016161=140616,017434
 F DIV R4 ;045670,123456/140616,017434=145246,047065
 F MUL R4 ;032107,065432*145246,047065=137201,106137
 F ADD 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,\$SP ;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,\$SP ;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 PAGE 72
CVK4CC.P11 16-AUG-78 08:41 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

CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 73
 CVKACC.P11 16-AUG-78 08:41 ADDRESS ERROR TEST

SEQ 0081

```

00100 3005
00200 3006
00300 3007
00400 3008
00500 3009
00600 3010
00700 3011 014032 104400
00800 3012 014034 012737 014124 000004
00900 3013 014042 012737 000340 000006
01000 3014 014050 004567 002132
01100 3015 014054 070707 016161
01200 3016 014060 146314 143434
01300 3017 014064 000143
01400 3018 014066 016456 000340
01500 3019 014072
01600 3020 014072 106427
01700 3021 014076 012702 177777
01800 3022
01900 3023 014102 000240
02000 3024 014104 075002
02100 3025
02200 3026 014106 004767 002126
02300 3027 014112 010267 164316
02400 3028 014116 104002
02500 3029 014120 000505
02600 3030 014122 000464
02700 3031
02800 3032 014124 004767 002140
02900 3033 014130 010267 16 300
03000 3034 014134 122767 000340 164270
03100 3035 014142 001402
03200 3036 014144 104000
03300 3037 014146 000506
03400 3038
03500 3039 014150 022767 177777 164256
03600 3040 014156 001402
03700 3041 014160 104000
03800 3042 014162 000507
03900 3043
04000 3044 014164 022767 014106 164244
04100 3045 014172 001402
04200 3046 014174 104001
04300 3047 014176 000510
04400 3048
04500 3049 014200 022767 000151 164232
04600 3050 014206 001402
04700 3051 014210 104002
04800 3052 014212 000511
04900 3053
05000 3054 014214 022767 070707 164220
05100 3055 014222 001402
05200 3056 014224 104004
05300 3057 014226 000512
05400 3058
05500 3059 014230 022767 016161 164206
05600 3060 014236 001402

```

 :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
;SECOND OPERAND ON TOP
;WORD 070707,016161 ;FIRST OPERAND ON BOTTOM
;WORD 146314,143434 ;PROCESSOR PRIORITY LEVEL
;WORD 143 ;FIS TRAP VECTOR
;WORD TRAPER, 340 ;SET PROCESSOR STATUS
MTPS #143
;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, POPR ;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
;PS AFTER TRAP NOT EQUAL TO 340
HLT 506 ;THE ERROR NUMBER IS 506

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

```

Address	Op Code	Op	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380
---------	---------	----	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

00100 CVKACC MARY11 30A(1052) 21-AUG-78 15.28 PAGE 75
 00200 CVKACC.P11 16-AUG-78 08:41 ADDRESS ERROR TEST

SEQ 0083

```

00400 3077
00500 3078
00600 3079
00700 3080
00800 3081
00900 3082
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

```

```

*****
: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      END66            ;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
        .WORD   520              ;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
        .WORD   521              ;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
        .WORD   522              ;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
        .WORD   523              ;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.
        .WORD   524              ;THE ERROR NUMBER IS 524

```

00100 CVKACC MARY1 30A(1052) 21-AUG-78 15:28 PAGE 76
 00200 CVKACC.P1 16-AUG-78 08:41 ADDRESS ERROR TEST

SEQ 0084

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

MOV #6, @#4 ;RESTORE TIME-OUT VECTOR
 CLR @#6
 MOV #3, TIMES ;REDUCE NUMBER OF ITERATIONS

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

 TST67: SCOPE
 BITB #40, @#SENVN
 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, @#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
 1\$: MOV #.+6, LADS ;RESET LOOP ADDRESS
 JSR R5, PUSH ;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, @#TPS ;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: CLRB @#TPS ;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\$: TSTB @#TPS ;LOOP HERE UNTILL DONE BIT COMES ON
 BPL 2\$
 MOVB #15, @#TPB ;TYPE ANOTHER 'CR'
 MOV #100, @#TPS ;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

(0100 CVKACL MA(Y11 30A(1052) 21-AUG-78 15:28 PAGE 77
 00200 CVKACC.P11 16-AUG-78 08:41 INTERRUPT ABORT TEST SECTION

SEQ 0085

00300	3187	014756	001407			BEQ	6\$		:BRANCH IF OK
00400	3188	014760	010667	163450		MOV	SP,	\$SP	:SAVE FOR TYPING
00500	3189	014764	012706	000574		MOV	#BEGIN-4,	SP	:RESTORE THE STACK POINTER
00600	3190	014770	104000			HLT			:STACK POINTER FOULED UP
00700	3191	014772	000525			525			:THE ERROR NUMBER IS 525
00800	3192	014774	000457			BR	END67		:SKIP REST OF TEST
00900	3193								
01000	3194	014776	010167	163432		MOV	R1,	\$SP	:SAVE STACK POINTER
01100	3195	015002	122767	000344	163422	CMPB	#344,	\$PSW	:CHECK PS AFTER INTERRUPT
01200	3196	015010	001402			BEQ	.*6		:BRANCH IF OK
01300	3197	015012	104000			HLT			:PS AFTER INTERRUPT NOT EQUAL TO LVLA
01400	3198	015014	000526			526			:THE ERROR NUMBER IS 526
01500	3199								
01600	3200	015016	022767	000510	163410	CMP	#STACK0,	\$SP	:CHECK THE STACK POINTER (R1)
01700	3201	015024	001402			BEQ	.*6		:BRANCH IF OK
01800	3202	015026	104000			HLT			:STACK POINTER (R1) NOT EQUAL TO #STACK0
01900	3203	015030	000527			527			:THE ERROR NUMBER IS 527
02000	3204								
02100	3205	015032	022767	014640	163376	CMP	#RTA67,	ANS1	:CHECK FIS TRAP RETURN ADDRESS
02200	3206	015040	001402			BEQ	.*6		:BRANCH IF OK
02300	3207	015042	104001			HLT+1			:FIS TRAP AT WRONG ADDRESS
02400	3208	015044	000530			530			:THE ERROR NUMBER IS 530
02500	3209								
02600	3210								
02700	3211	015046	022767	000177	163366	CMP	#000177,	ANS3	:CHECK DATA FROM THE STACK
02800	3212	015054	001402			BEQ	.*6		:BRANCH IF OK
02900	3213	015056	104004			HLT+4			:DATA ON STACK (000177) CHANGED
03000	3214	015060	000531			531			:THE ERROR NUMBER IS 531
03100	3215								
03200	3216	015062	022767	134543	163354	CMP	#134543,	ANS4	:CHECK DATA FROM STACK
03300	3217	015070	001402			BEQ	.*6		:BRANCH IF OK
03400	3218	015072	104004			HLT+4			:DATA ON STACK (134543) CHANGED
03500	3219	015074	000532			532			:THE ERROR NUMBER IS 532
03600	3220								
03700	3221	015076	022767	035700	163342	CMP	#035700,	ANS5	:CHECK DATA FROM STACK
03800	3222	015104	001402			BEQ	.*6		:BRANCH IF OK
03900	3223	015106	104006			HLT+6			:DATA ON STACK (035700) CHANGED
04000	3224	015110	000533			533			:THE ERROR NUMBER IS 533
04100	3225								
04200	3226	015112	022767	143235	163330	CMP	#143235,	ANS6	:CHECK DATA FROM STACK
04300	3227	015120	001402			BEQ	.*6		:BRANCH IF OK
04400	3228	015122	104006			HLT+6			:DATA ON STACK (143235) CHANGED
04500	3229	015124	000534			534			:THE ERROR NUMBER IS 534
04600	3230								
04700	3231	015126	005367	163402		DEC	TEMP		:STAY IN THE LOOP FOR 30 TIMES
04800	3232	015132	001261			BNE	1\$		
04900	3233								
05000	3234	015134	022626			CMP	(SP)+,	(SP)+	:RESTORE STACK POINTER TO 500
05100	3235	015136	122767	000067	163240	CMPB	#67,	\$TESTN	:CHECK THE TEST NUMBER
05200	3236	015144	001402			BEQ	.*6		:BRANCH IF OK
05300	3237	015146	104000			HLT			:WRONG TEST. PC MUST HAVE FOULED UP.
05400	3238	015150	000535			535			:THE ERROR NUMBER IS 535
05500	3239	015152				MTPS	#340		
05600	3240	015152	106427			.WORD	*06400...		
05700	3241								
05800	3242								

END67:

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

SEQ 0086

00300 3242
 00400 3243
 00500 3244
 00600 3245
 00700 3246
 00800 3247
 00900 3248
 01000 3249 015156 104400
 01100 3250 015160 132737 000040 000421
 01200 3251 015166 001176
 01300 3252 015170 013704 000546
 01400 3253 015174 012724 015276
 01500 3254 015200 012714 000340
 01600 3255 015204 032737 004000 000422
 01700 3256 015212 001005
 01800 3257 015214 000004 000473
 01900 3258
 02000 3259 015220 012767 015226 163242
 02100 3260 015226 004567 000754
 02200 3261 015232 040200 000000
 02300 3262 015236 107070 070707
 02400 3263 015242 000100
 02500 3264 015244 016456 000340
 02600 3265 015250 012700 000510
 02700 3266 015254 012767 000060 163252
 02800 3267 015262 112777 000100 163260
 02900 3268
 03000 3269 015270 075020
 03100 3270 015272 024040
 03200 3271 015274 000775
 03300 3272
 03400 3273 015276 105077 163246
 03500 3274 015302 022716 015270
 03600 3275 015306 001424
 03700 3276 015310 022766 015270 000004
 03800 3277 015316 001423
 03900 3278 015320 005337 000534
 04000 3279 015324 001516
 04100 3280 015326 112777 000015 163216
 04200 3281 015334 105777 163210
 04300 3282 015340 100375
 04400 3283 015342 112777 000015 163202
 04500 3284 015350 012777 000100 163172
 04600 3285 015356 000002
 04700 3286
 04800 3287 015360 004767 000704
 04900 3288 015364 000403
 05000 3289
 05100 3290 015366 022626
 05200 3291 015370 004767 000700
 05300 3292 015374 005746
 05400 3293 015376 012746 015270
 05500 3294 015402 022706 000574
 05600 3295 015406 001407
 05700 3296 015410 010667 163020
 05800 3297 015414 012706 000574
 05900 3298

 TEST 70: TEST THAT FIS ABORTS PROPERLY WHEN INTERRUPTED
 107070,070707 * 040200,000000 107070,070707
 PS - .PS, STACK POINTER R0

TST70: SCOPE
 BITB #40, @SENVM
 BNE END70+2 ;EXIT THIS TEST IF BIT 5 OF SENVM IS HIGH
 MOV @TTYOUT, R4
 MOV #ISR70, (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
 ;RESET LOOP ADDRESS
 1\$: MOV #.+6, LAD\$
 JSR R5, PUSH\$;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
 .WORD 040200,000000 ;SECOND OPERAND ON TOP
 .WORD 107070,070707 ;FIRST OPERAND ON BOTTOM
 .WORD 100 ;PROCESSOR PRIORITY LEVEL
 .WORD TRAPER, 340 ;FIS TRAP VECTOR
 MOV #STACK0, R0 ;SET UP STACK POINTER
 MOV #60, TEMP ;INITIALIZE COUNTER FOR FIS INTERRUPTS
 MOVB #100, @STPS ;SET TTY INTERRUPT ENABLE

RTA70: FMUL R0
 CMP -(R0), -(R0) ;FLOATING MULTIPLY ON THE STACK
 BR RTA70 ;RESET THE STACK POINTER FOR NEXT PASS
 ;REPEAT UNTIL INTERRUPTED

ISR70: CLRB @STPS
 CMP #RTA70, (SP) ;CLEAR THE INTERRUPT ENABLE
 BEQ 3\$;CHECK IF INTERRUPT AT FIS INSTR.
 CMP #RTA70, 4(SP) ;BRANCH IF IT DID
 BEQ 4\$;CHECK FOR INTERRUPT WITH T-BIT SET
 DEC @TEMP ;BRANCH IF IT DID
 BEQ END70 ;DID INTERRUPTS OCCUR TEMP # OF TIMES
 ;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'
 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
 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

(C)100	(VKACC)	MAR(Y'1)	30A(1052)	21-AUG-78	15:28	PAGE 79
00200	(VKACC,P'		16-AUG-78	08:41		INTERUPT ABORT TEST SECTION
00300						
00400	3298	015420	104000			:STACK POINTER FOULED UP
00500	3299	015422	000536			:THE ERROR NUMBER IS 536
00600	3300	015424	000456			:SKIP REST OF TEST
00700	3301					
00800	3302	015426	010067	163002	6S:	:SAVE STACK POINTER
00900	3303	015432	122767	000344	162772	:CHECK PS AFTER INTERUPT
01000	3304	015440	001402			:BRANCH IF OK
01100	3305	015442	104000			:PS AFTER INTERUPT NOT EQUAL TO LVLA
01200	3306	015444	000537			:THE ERROR NUMBER IS 537
01300	3307					
01400	3308	015446	022767	000510	162760	:CHECK THE STACK POINTER (R0)
01500	3309	015454	001402			:BRANCH IF OK
01600	3310	015456	104000			:STACK POINTER (R0) NOT EQUAL TO #STACKO
01700	3311	015460	000540			:THE ERROR NUMBER IS 540
01800	3312					
01900	3313	015462	022767	015270	162746	:CHECK FIS TRAP RETURN ADDRESS
02000	3314	015470	001402			:BRANCH IF OK
02100	3315	015472	104001			:FIS TRAP AT WRONG ADDRESS
02200	3316	015474	000541			:THE ERROR NUMBER IS 541
02300	3317					
02400	3318					
02500	3319	015476	022767	040200	162736	:CHECK DATA FROM THE STACK
02600	3320	015504	001402			:BRANCH IF OK
02700	3321	015506	104004			:DATA ON STACK (040200) CHANGED
02800	3322	015510	000542			:THE ERROR NUMBER IS 542
02900	3323					
03000	3324	015512	005767	162726		:CHECK DATA FROM STACK
03100	3325	015516	001402			:BRANCH IF OK
03200	3326	015520	104004			:DATA ON STACK (000000) CHANGED
03300	3327	015522	000543			:THE ERROR NUMBER IS 543
03400	3328					
03500	3329	015524	022767	107070	162714	:CHECK DATA FROM STACK
03600	3330	015532	001402			:BRANCH IF OK
03700	3331	015534	104006			:DATA ON STACK (107070) CHANGED
03800	3332	015536	000544			:THE ERROR NUMBER IS 544
03900	3333					
04000	3334	015540	022767	070707	162702	:CHECK DATA FROM STACK
04100	3335	015546	001402			:BRANCH IF OK
04200	3336	015550	104006			:DATA ON STACK (070707) CHANGED
04300	3337	015552	000545			:THE ERROR NUMBER IS 545
04400	3338					
04500	3339	015554	005367	162754		:STAY IN THE LOOP FOR 30 TIMES
04600	3340	015560	001262			
04700	3341					
04800	3342	015562	022626			:RESTORE STACK POINTER TO 500
04900	3343	015564	122767	000070	162612	:CHECK THE TEST NUMBER
05000	3344	015572	001402			:BRANCH IF OK
05100	3345	015574	104000			:WRONG TEST - PC MUST HAVE FOULED UP.
05200	3346	015576	000546			:THE ERROR NUMBER IS 546
05300	3347	015600				
05400	3348	015600	106427			
05500	3349					

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	CLR	(R4)	
00700	3353					53300			

CC100 CVKACC MARY11 30A(1052) 21-AUG-78 15:28 PAGE 81
 00200 CVKACC.P11 16-AUG-78 08:41 INTERRUPT ABORT TEST SECTION

SEQ 0089

```

00300
00400 3354
00500 3355
00600 3356
00700 3357
00800 3358
00900 3359
01000 3360
01100 3361
01200 3362
01300 3363
01400 3364
01500 3365 015620
01600 3366 015620 104400
01700 3367 015622 005267 162560
01800 3368 015626 042767 100000 162552
01900 3369 015634 005327
02000 3370 015636 000001
02100 3371 015640 003015
02200 3372 015642 012737
02300 3373 015644 000001
02400 3374 015646 015636
02500 3375 015650 000004 015700
02600 3376 015654
02700 3377
02800 3378 015654 013700 000042
02900 3379 015660 001405
03000 3380 015662 000005
03100 3381 015664 004710
03200 3382 015666 000240
03300 3383 015670 000240
03400 3384 015672 000240
03500 3385 015674
03600 3386 015674 000137 000600
03700 3387 015700 005015 047105 020104
03800 3388 015706 040520 051523
03900 3389 015712 377 377 000
04000 3390 015716
04100 3391
04200 3392 015644 000001 53600 54100

```

```

*****
.SBTTL  END OF PASS ROUTINE

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

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

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

$DOAGN: JMP @#BEGIN
  ;;RETURN
$ENDMG: .ASCII <15><12>/END PASS/
$ENULL: .BYTE -1,-1,0
  .EVEN
  ;;NULL CHARACTER STRING

ENDCT:

```

CVKACC MAY 11 30A(1052) 21-AUG-78 15:28 PAGE 82
 CVKACC.P11 16-AUG-78 08:41 END OF PASS ROUTINE

SEQ 0046

```

00100 3393 ;*****
00200 3394
00300 3395
00400 3396
00500 3397 015716 032737 000400 000422
00600 3398 015724 001404
00700 3399 015726 123767 000422 162450
00800 3400 015734 001431
00900 3401 015736 032737 040000 000422
01000 3402 015744 001023
01100 3403 015746 032737 004000 000422
01200 3404 015754 001412
01300 3405 015756 105767 162521
01400 3406 015762 001404
01500 3407 015764 126767 162546 162511
01600 3408 015772 001010
01700 3409 015774 112767 000001 62501
01800 3410 016002 105267 162376
01900 3411 016006 011667 162456
02000 3412 016012 000002
02100 3413
02200 3414 016014 105267 162463
02300 3415 016020 005767 162444
02400 3416 016024 001766
02500 3417 016026 016716 162436
02600 3418 016032 000002
02700 3419
02800
02900
03000
  
```

.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
  
```

00100 CVKACC MAC11 30A(1052) 21-AUG-78 15:28 PAGE 83
 00200 CVKACC.P11 16-AUG-78 08:41 PUSH AND POP SUBROUTINES

SEQ 0091

```

00300
00400 3420 55000
00500 3421 55100 ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
00600 3422 55200
00700 3423 016034 005726 55300 PUSHES: TST (SP)+ ;POP STACK BY 1
00800 3424 016036 062705 000010 55400 ADD #10, R5 ;POINT TO END OF DATA
00900 3425 016042 014546 55500 MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
01000 3426 016044 014546 55600 MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
01100 3427 016046 014546 55700 MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
01200 3428 016050 014546 55800 MOV -(R5), -(SP) ;PUSH DATA ONTO THE STACK
01300 3429 016052 062705 000010 55900 ADD #10, R5 ;POINT TO END OF DATA
01400 3430 016056 56300 MTPS (R5)+ ;SET THE PROCESSOR STATUS
01500 3431 016056 106425 .WORD 106400...C
01600 3432 016060 005205 56700 INC R5
01700 3433 016062 012577 162376 56800 MOV (R5)+, @FISVEC ;SET UP FIS ERROR TRAP VECTOR
01800 3434 016066 012577 162374 56900 MOV (R5)+, @FISLVL ;TRAP STATUS
01900 3435 016072 000115 57000 JMP (R5) ;RETURN
02000 3436 57100
02100 3437 57200
02200 3438 57300 ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
02300 3439 57400 ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
02400 3440 57500
02500 3441 016074 57900 POPS: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
02600 3442 016074 106767 .WORD 106700...C
02700 3443 016100 042767 000020 162324 58300 BIC #20, $PSW ;CLEAR T-BIT
02800 3444 016106 012604 58400 MOV (SP)+, R4 ;SAVE RTS ADDRESS
02900 3445 016110 012667 162322 58500 MOV (SP)+, ANS1 ;SAVE THE ANSWER
03000 3446 016114 012667 162320 58600 MOV (SP)+, ANS2
03100 3447 016120 010667 162310 58700 MOV SP, $SP ;SAVE THE STACK POINTER
03200 3448 016124 000114 58800 JMP (R4) ;RETURN
03300 3449 58900
03400 3450 59000
03500 3451 59100 ;SUBROUTINE TO POP 6 WORDS OFF THE STACK.
03600 3452 59200 ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP.
03700 3453 59300 ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
03800 3454 59400 ;ALSO SAVES THE PS AND STACK POINTER.
03900 3455 59500
04000 3456 016126 59900 POPES: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
04100 3457 016126 106767 .WORD 106700...C
04200 3458 016132 012604 60300 MOV (SP)+, R4 ;SAVE RTS ADDRESS
04300 3459 016134 012667 162276 60400 MOV (SP)+, ANS1 ;SAVE RTI ADDRESS
04400 3460 016140 011667 162274 60500 MOV (SP), ANS2 ;SAVE RTI STATUS
04500 3461 016144 042767 000020 162266 60600 BIC #20, ANS2 ;CLEAR THE T-BIT
04600 3462 016152 012746 016160 60700 MOV #1$, -(SP)
04700 3463 016156 000002 60800 RTI ;RESTORE THE PROCESSOR STATUS
04800 3464 016160 012667 162256 60900 1$: MOV (SP)+, ANS3 ;SAVE DATA
04900 3465 016164 012667 162254 61000 MOV (SP)+, ANS4
05000 3466 016170 012667 162252 61100 MOV (SP)+, ANS5
05100 3467 016174 012667 162250 61200 MOV (SP)+, ANS6
05200 3468 016200 010667 162230 61300 MOV SP, $SP ;SAVE SP
05300 3469 016204 000114 61400 JMP (R4) ;RTS
05400 3470 61500
05500 3471 61600 ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
05600 3472 61700
05700 3473 016206 012704 000510 61800 PUSH: MOV #STACK0,R4 ;SET R4 TO STACK
05800 3474 016212 012524 61900 MOV (R5)+, (R4)+ ;PUT DATA ON STACK
05900 3475 016214 012524 62000 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

```

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 76700 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

```

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 86
 00200 CVKACC.P11 16-AUG-78 08:41 PUSH AND POP SUBROUTINES

SEQ 0094

```

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 $ICNT
03200 3580 016624 000167 177164 JMP <IT$ ;LOOP ON TEST UNTIL NO ERRORS
03300 3581
03400 3582

```

72300

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 87
 00200 CVKACC.P11 16-AUG-78 08:41 HLT ROUTINE (ERROR TYPEOUT)

SEQ 0095

00300	3583				72500			
00400	3584							
00500	3585				72700			
00600	3586				72800			
00700	3587				72900			
00800	3588	016630	117767	161574	73000			
00900	3589	016636	062767	000002	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 @HLTAD\$,TYPcnt ;TYPE COUNT IS LOW BYTE OF HLT
 ADD #2,TYPcnt ;TYPE COUNT - X+2
 MOV #HLTAD\$,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 TYPcnt ;CHECK FOR DONE
 BPL ERR1\$;BRANCH IF NOT DONE
 RTS PC

PC	CVKACC	MAC	Y11	30A(1052)	21-AUG-78	15:28	PAGE 88	OCTAL WORD & ADDRESS	TYPED
00100	CVKACC	P11	16-AUG-78	08:41					
00200									
00300									
00400	3599	016672	112767	000001	161644			PRINTR: MOV	#1, .PR
00500	3600	016700	000402					BR	.+6
00600	3601	016702	005067	161636				PRINTS: CLR	.PR
00700	3602	016706	112767	177772	161631			MOVB	#-6, .PR+1
00800	3603	016714	010446					MOV	R4, -(6)
00900	3604	016716	012704	017020				MOV	#.PRBUF, R4
01000	3605	016722	105014					CLRB	(4)
01100	3606	016724	000405					BR	.PRF
01200	3607	016726	105014					.PRL: CLRB	(4)
01300	3608	016730	006105					ROL	TTY
01400	3609	016732	106114					ROLB	(4)
01500	3610	016734	006105					ROL	TTY
01600	3611	016736	106114					ROLB	(4)
01700	3612	016740	006105					.PRF: ROL	TTY
01800	3613	016742	106114					ROLB	(4)
01900	3614	016744	105714					TSTB	(4)
02000	3615	016746	001402					BEQ	.+6
02100	3616	016750	105267	161570				INCB	.PR
02200	3617	016754	105767	161564				TSTB	.PR
02300	3618	016760	001402					BEQ	.+6
02400	3619	016762	152724	000060				BISB	#'0, (4)+
02500	3620	016766	105267	161553				INCB	.PR+
02600	3621	016772	001355					BNE	.PRL
02700	3622	016774	022704	017020				CMP	#.PRBUF, R4
02800	3623	017000	001002					BNE	.+6
02900	3624	017002	112724	000060				MOVB	#'0, (4)+
03000	3625	017006	105014					CLRB	(4)
03100	3626	017010	000004	017020				TYPE	.PRBUF
03200	3627	017014	012604					MOV	(6)+, R4
03300	3628	017016	000207					RTS	PC
03400	3629								
03500	3630	017020	000004					.PRBUF: .BLKW	4

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 89
 00200 CVKACC.P11 16-AUG-78 08:41 OCTAL WORD & ADDRESS TYPE

SEQ 0097

:*****

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

00100 (VKACC MA 30A(1052) 21-AUG-78 15:28 PAGE 90
 00200 (VKACC.P 16-AUG-78 08:41 ASCIZ TYPE OUT ROUTINE

SEQ 0098

```

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 $TYPE: BITB #40,@$SENV ;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 ASSCIZ 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$: ADD #2,(SP) ;ADJUST THE RETURN PC
02100 3688 017222 000002 76400 RTI ;AND RETURN
02200 3689 76500
02300 3690 017224 105777 161320 76600 4$: TSTB @2$PS ;IS THE PRINTER AVAILABLE?
02400 3691 017230 100375 76700 BP 4$ ;IF NOT THEN LOOP HERE
02500 3692 017232 112677 161314 76800 MOVB (SP)+,@2$PS ;OUT PUT THE CHARACTER
02600 3693 017236 000763 76900 BR 2$ ;AND GO BACK
02700 3694 017240 005015 053104 040513 77000 $TITLE: .ASCIZ <15><12>/DVKACC - LSI-11 FIS INSTRUCTION TEST/
02800 3695 017246 041503 026440 046040
02900 3696 017254 044523 030455 020061
03000 3697 017262 044506 020123 044440
03100 3698 017270 051516 051124 041525
03200 3699 017276 044524 047117 052040
03300 3700 017304 051505 000124
03400 3701
03500 3702 017310 012706 000600 77100 .EVEN
03600 3703 017314 132737 000001 000420 77200 NOOP: MOV #BEGIN,SP ;THAT WAS THE HEADING FOR THE DIAGNOSTIC
03700 3704 017322 001011 77300 BITB #1,@$SENV ;INITIALIZE STACK POINTER TO 600
03800 3705 017324 132737 000040 000421 77400 BNE 22$ ;ARE WE UNDER APT
03900 3706 017332 001005 77450 BITB #40,@$SENV ;IF SO THEN DO NOT TYPE HEADING
04000 3707 017334 012737 017170 000020 77460 BNE 22$ ;HAVE THE CONSOLE OUTPUTS BEEN SUPPRESSED
04100 3708 017342 000004 017240 77500 MOV #20,$TYPE ;IF SO THEN DO NOT PRINT HEADING
04200 3709 017346 012767 000001 161162 77600 TYPE $TITLE ;SET UP VECTOR 20 TO PRINT HEADING
04300 3710 017354 012700 000410 77700 22$: MOV #1,TIMES ;TYPE HEADING 'VKACC -LSI-11 ...'
04400 3711 017360 005040 77800 MOV #2,$DEVCT,R0 ;# OF ITERATIONS IN THE FIRST PASS
04500 3712 017362 022700 000400 77900 2$: CLR -(R0) ;PREPARE TO INITIALIZE THE PROGRAM
04600 3713 017366 001374 78000 CMP #SMALL,R0
04700 3714 017370 000167 160610 78100 BNE 2$
04800 3715 000001 78200 JMP RESTRT ;START THE PROGRAM
78300 .END

```

```

CVKACC MACY11
CVKACC.P11

A 015716
ABASE - 000000
ACDW1 000000
ACDW2 = 000000
ACPUOP 000000
ADDW0 - 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11- 000000
ADDW12 000000
ADDW13- 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 - 000000
ADDW4 000000
ADDW5 - 000000
ADDW6 - 000000
ADDW7 = 000000
ADDW8 - 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM = 000000
AENV 000000
AENVM = 000000
AFATAL= 000000
AMADR1- 000000
AMADR2= 000000
AMADR3 000000
AMADR4- 000000
AMAMS1 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4 000000
AMSGAD- 000000
AMSGLG- 000000
AMSGTY- 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3- 000000
AMTYP4- 000000
ANS1 000436

ANS2 000440

ANS3 000442

ANS4 000444

ANS5 000446

```

3392#	3393
100	
100	
100	
100	118
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	109
100	
100	114
100	115
100	106
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	
100	111
100	112
100	105
100	
100	
100	
100	
156#	157
765	811
1487	1533
2143	2187
2901	2976
158#	159
770	816
1492	1538
2148	2192
2906	2977
160#	161
2633	2695
162#	870
2700	2772
163#	937

3393

118

109

'14
'15
'06

111
112
105

157
811
1533
2187
2976
159
816
1538
2192
2977
161
2695
870
2772
937

259
855
1579
2241
2990
264
860
1584
2246
2995
865
2767
932
2844
1009

305
917
1625
2303
3044
310
922
1030
2308
3049
927
2839
1004
2022
1081

351
989
1671
2375
3117
356
994
1676
2380
3122
999
2917
1076
3059
1153

397
1061
1723
2441
3205
402
1065
1728
2446
3446
1071
3054
1148
3216
1795

443
1133
1775
2487
3313
448
1138
1780
2492
3460
1143
3211
1790
3324
1867

489
1199
1847
2533
3445
494
1204
1852
2538
3461
1785
3319
1862
3465
2323

535
1245
1913
2579
3459
540
1250
1918
2584
3492
1857
3464
2202
3511
2394

58
1297
1959
2623
3492
586
1302
1964
2628
3506
2197
3510
2256
354
270

62
134
200
268
350
63
134
201
269
350
225
354
231
277

67
139
205
275
353
67
140
205
276
354
231
239
284

71
144
209
282

72
144
210
283
238
263
292

00100 CVKACC MACV11 30A(1052) 21-AUG-78 15:28 PAGE 93
 00200 CVKACC.P11 16-AUG-78 08:41 CROSS REFERENCE TABLE -- USER SYMBOLS
 00300
 00400
 00500 ANS6 000450 3064 3221 3329 3466* 3512*
 00600 164# 942 1014 1086 1158 1800 1872 2328 2400 2710 2782 2854 293
 00700 APASS = 000000 3069 3226 3334 3467* 3513*
 00800 APRIOR = 000000 100 108
 00900 ASWREG = 000000 100 116
 01000 ATESTN = 000000 100 107
 01100 AUNIT = 000000 100 110
 01200 AUSWR = 000000 100 117
 01300 AVECT1 = 000000 100
 01400 AVECT2 = 000000 100
 01500 BEGIN 000600 201 208# 431 433 477 479 707 709 753 755 973 977 97
 01600 1117 1121 1123 1279 1281 1377 1379 1705 1707 1759 1763 1765 208
 01700 2087 2359 2363 2365 2475 2477 2885 2889 2891 3186 3189 3294 329
 01800 3386 3702
 01900 ENDCT 015644 3392#
 02000 END1 001040 269#
 02100 END10 002254 591#
 02200 END11 002412 637#
 02300 END12 002550 683#
 02400 END13 002702 712 729#
 02500 END14 003036 758 775#
 02600 END15 003176 821#
 02700 END16 003350 875#
 02800 END17 003602 903 947#
 02900 END2 001200 315#
 03000 END20 004032 974 982 1019#
 03100 END21 004270 1047 1091#
 03200 END22 004522 1118 1126 1163#
 03300 END23 004662 1209#
 03400 END24 005016 1255#
 03500 END25 005152 1290 1307#
 03600 END26 005310 1353#
 03700 END27 005444 1388 1405#
 03800 END3 001334 361#
 03900 END30 005604 1451#
 04000 END31 005744 1497#
 04100 END32 006104 1543#
 04200 END33 006242 1589#
 04300 END34 006376 1635#
 04400 END35 006534 1681#
 04500 END36 006670 1716 1733#
 04600 END37 007122 1760 1768 1805#
 04700 END4 001470 407#
 04800 END40 007360 1833 1877#
 04900 END41 007514 1923#
 05000 END42 007654 1969#
 05100 END43 010010 2015#
 05200 END44 010144 2061#
 05300 END45 010300 2090 2107#
 05400 END46 010436 2153#
 05500 END47 010612 2207#
 05600 END5 001622 436 453#
 05700 END50 010766 2261#
 05800 END51 011222 2289 2333#
 05900 END52 011454 2360 2368 2405#

SEQ 0100

[illegible]

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

00100 CVKACC MACY11 30A(1052) 21-AUG-78 15:28 PAGE 97
 00200 CVKACC.P11 16-AUG-78 08:41 CROSS REFERENCE TABLE -- USER SYMBOLS
 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 TTYOUT 000546 191# 3144 3252 3351#
 05900 TYPCNT 000540 188# 3588# 3589# 3595#

SEQ 0104

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	SCPUOP	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	166
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	1846#	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

00100 CVKACC MAY11 30A(1052) 21-AUG-78 15:28 PAGE 99

00200 CVKACC.P11 16-AUG-78 08:41

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0106

00300			2434	2435#	2439	2440#	2444	2445#	2449	2450#	2454	2455#	2479	2480#	248
00400			2486#	2490	2491#	2495	2496#	2500	2501#	2526	2527#	2531	2532#	2536	253
00500			2541	2542#	2546	2547#	2572	2573#	2577	2578#	2582	2583#	2587	2588#	259
00600			2593#	2621	2622#	2626	2627#	2631	2632#	2636	2637#	2641	2642#	2646	264
00700			2670	2671#	2678	2679#	2683	2684#	2688	2689#	2693	2694#	2698	2699#	270
00800			2704#	2708	2709#	2713	2714#	2718	2719#	2742	2743#	2750	2751#	2755	275
00900			2760	2761#	2765	2766#	2770	2771#	2775	2776#	2780	2781#	2785	2786#	279
01000			2791#	2814	2815#	2822	2823#	2827	2828#	2832	2833#	2837	2838#	2842	284
01100			2847	2848#	2852	2853#	2857	2858#	2862	2863#	2884	2885#	2893	2894#	289
01200			2900#	2904	2905#	2915	2916#	2920	2921#	2925	2926#	2930	2931#	2935	293
01300			2940	2941#	2982	2983#	2987	2988#	2993	2994#	2998	2999#	3003	3004#	302
01400			3030#	3037	3038#	3042	3043#	3047	3048#	3052	3053#	3057	3058#	3062	306
01500			3067	3068#	3072	3073#	3076	3077#	3102	3103#	3110	3111#	3115	3116#	312
01600			3121#	3125	3126#	3130	3131#	3191	3192#	3198	3199#	3203	3204#	3208	320
01700			3214	3215#	3219	3220#	3224	3225#	3229	3230#	3238	3239#	3299	3300#	330
01800			3307#	3311	3312#	3316	3317#	3322	3323#	3327	3328#	3332	3333#	3337	333
01900			3346	3347#	3549	3550#									
02000			106#	3561*											
02100	\$FATAL	000402													
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	\$TESTN	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

		CROSS REFERENCE TABLE --	USER SYMBOLS								SEQ 0107					
00100																
00200																
00300																
00400			1497	1543	1589	1635	1681	1733	1805	1877	1923	1969	2015	2061		
00500			2153	2207	2261	2333	2405	2451	2497	2543	2589	2643	2715	2787		
00600			2937	3000	3073	3127	3235	3343	3399	3410*				210		
00700	\$IN	000001	14#													
00800	\$TPB	000552	193#	221	3172*	3175*	3280*	3283*	3692*							
00900	\$TPS	000550	192#	3159*	3165*	3173	3176*	3267*	3273*	3281	3284*	3690				
01000	\$TSTM	000434	145#													
01100	\$TITLE	017240	3694#	3708												
01200	\$TYPE	017170	210	3679#	3707											
01300	\$UNIT	000412	110#													
01400	\$UNITM	000440	147#													
01500	\$USWR	000424	117#													
01600	.	017374	81#	82	87	88#	90#	92#	96#	132	133#	135#	137#	149#		
01700			153#	155#	157#	159#	161#	199#	206#	250	255	260	265	270		
01800			301	306	311	316	342	347	352	357	362	388	393	398		
01900			408	439	444	449	454	485	490	495	500	526	531	536		
02000			546	572	577	582	587	592	618	623	628	633	638	664		
02100			674	679	684	715	720	725	730	761	766	771	776	802		
02200			812	817	822	851	856	861	866	871	876	908	913	918		
02300			928	933	938	943	948	985	990	995	1000	1005	1010	1015		
02400			1052	1057	1062	1067	1072	1077	1082	1087	1092	1129	1134	1139		
02500			1149	1154	1159	1164	1190	1195	1200	1205	1210	1236	1241	1246		
02600			1256	1293	1298	1303	1308	1334	1339	1344	1349	1354	1391	1396		
02700			1406	1432	1437	1442	1447	1452	1478	1483	1488	1493	1498	1524		
02800			1534	1539	1544	1570	1575	1580	1585	1590	1616	1621	1626	1631		
02900			1662	1667	1672	1677	1682	1719	1724	1729	1734	1771	1776	1781		
03000			1791	1796	1801	1806	1838	1843	1848	1853	1858	1863	1868	1873		
03100			1904	1909	1914	1919	1924	1950	1955	1960	1965	1970	1996	2001		
03200			2011	2016	2042	2047	2052	2057	2062	2093	2098	2103	2108	2134		
03300			2144	2149	2154	2183	2188	2193	2198	2203	2208	2237	2242	2247		
03400			2257	2262	2294	2299	2304	2309	2314	2319	2324	2329	2334	2371		
03500			2381	2386	2391	2396	2401	2406	2432	2437	2442	2447	2452	2483		
03600			2493	249												

COMMON
DUMP
ENDCOM
ESCAPE
MFPS
MTPS
MUL
NEWTS
POP
PRINT
PUSH
REPORT
SDUMP
SETUP
SKIP
SLASH
STARS
TYPBIN
TYPDEC
TYPNAM
TYPNUM
TYPOCS
TYPOCT
TYPTXT
\$ADDER
\$ADDES
\$ADDR
\$ADDS
\$ADRER
\$ALL
\$ALL2
\$DIVER
\$DIVES
\$DIVR
\$DIVS
\$FATL

[illegible]

3564
2277
2964
4
4
4
88
95
22
41
300
294
296
264
286
241
245
25
39
52
66
80
92
102
114
125
140
154
168
180
192
206
220
232
244
258
270
282

359
309
301
365
363
7
102
109
27
45
307
272
250
25
39
53
67
80
93
104
114
128
143
157
171
183
195
208
221
233
244
258
270
282

344
309

8

32
68

279
254

26
40
53
67
81
93
105
115
129
143
157
172
184
195
209
223
233
245
259
271
283

345
323

9

36
73

26
40
54
68
81
94
105
115
130
144
158
172
184
196
210
224
235
247
262
277
288

2
4
5
6
8
9
10
11
13
14
15
17
18
19
21
22
23
24
26
27
28

350
347

1.

5

2
4
5
7
8
9
10
11
13
14
15
17
18
19
21
22
23
24
26
27
28

35
34

1

5

3
4
5
7
8
9
10
11
13
14
16
17
18
19
21
22
23
24
26
27
28

35

1

6

3
4
5
7
8
8
10
11
13
14
16
17
18
20
21
22
23
25
26
27
28

33

3

1
1
1
1
1
1
2
2
2
2
2
2

3

[illegible]

00100	CVKACC	MACY11	30A(1052)	21-AUG-78	15:28	PAGE 103	CROSS REFERENCE TABLE -- MACRO NAMES										SEQ 0109	
00200	CVKACC	P11	16-AUG-78	08:41														
00300																		
00400			2920	2925	2930	2935	2940	2982	2987	2993	2998	3003	3029	3037	3042	3047	305	
00500			3057	3062	3067	3072	3076	3102	3110	3115	3120	3125	3130	3191	3198	3203	320	
00600			3214	3219	3224	3229	3238	3299	3306	3311	3316	3322	3327	3332	3337	3346	354	
00700	\$FINT	194#	3134	3242														
00800	\$FIS7	194#	827	2159	2213	2595												
00900	\$HLT	27#	3552															
01000	\$MULER	194#	2267															
01100	\$MULES	194#	2339															
01200	\$MULR	194#	1883	1929	1975	2021	2113											
01300	\$MULS	194#	2067															
01400	\$OCTAL	28#	3599															
01500	\$SCOPE	26#	3393															
01600	\$SUBER	194#	1811															
01700	\$SUBES	194#	1739															
01800	\$SUBR	194#	1169	1215	1313	1411	1457	1503	1549	1595	1641							
01900	\$SUBS	194#	1261	1359	1687													
02000	\$SWDO	22#	30															
02100	\$SESCA	1#																
02200	\$SFATL	18#																
02300		364	252	257	262	267	272	298	303	308	313	318	344	349	354	35		
02400		502	390	395	400	405	410	435	441	446	451	456	481	487	492	49		
02500		640	528	533	538	543	548	574	579	584	589	594	620	625	630	63		
02600		778	666	671	676	681	686	711	717	722	727	732	757	763	768	77		
02700		920	804	809	814	819	824	853	858	863	868	873	878	902	910	91		
02800		1017	925	930	935	940	945	950	972	981	987	992	997	1002	1007	101		
02900		1136	1022	1046	1054	1059	1064	1069	1074	1079	1084	1089	1094	1116	1125	113		
03000		1253	1141	1146	1151	1156	1161	1166	1192	1197	1202	1207	1212	1238	1243	124		
03100		1403	1258	1289	1295	1300	1305	1310	1336	1341	1346	1351	1356	1387	1393	139		
03200		1541	1408	1434	1439	1444	1449	1454	1480	1485	1490	1495	1500	1526	1531	153		
03300		1679	1546	1572	1577	1582	1587	1592	1618	1623	1628	1633	1638	1664	1669	167		
03400		1803	1684	1715	1721	1726	173											

CVK100 (VKAC MA V 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 .SDB20 1#
 00500 .SDB20 1#
 00600 .SDI 1#
 00700 .SEOP 1# 4# 3354
 00800 .SEKR 1#
 00900 .SEKR 1#
 01000 .SMUL 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 .SSB20 1#
 01900 .SSB20 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