

LSI11

FIS DIAGNOSTIC
MD-11-DVKAC-A

EP-DVKAC-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

OCT 1976
digital
MADE IN USA

The microfiche card contains 48 frames of diagnostic data. The first 47 frames are organized into a grid of 8 rows and 6 columns. Each frame contains a specific diagnostic test or system status report. The data is printed in a small, dense font, typical of microfiche format. The 48th frame is blank.

SVKACA MACY11 27 732 24-AUG-76 15:37
 SVKACA.SAC TABLE OF CONTENTS

SEQ 0001

5290	SWITCH OPTIONS AND ASSIGNMENTS
5330	ACT11 WORKS
5334	VECTOR A-D, STACKS, ANSWER AREA, AND SETUP ROUTINE
5339	APT MACY D-X-ETABLE
5340	APT A-D-ENTER BLOCK
7106	STARTING OF THE PROGRAM
7144	FADD TEST SECTION
7161	TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW
7166	TEST FLOATING ADD INSTRUCTION WITH OVERFLOW
7171	FSUB TEST SECTION
7183	TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW
7187	TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW
7191	FML TEST SECTION
7204	TEST FLOATING MUL. INSTRUCTION WITH UNDERFLOW
7208	TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW
7212	FDIV TEST SECTION
7221	TEST FLOATING DIV. INSTRUCTION WITH UNDERFLOW
7225	TEST FLOATING DIV. INSTRUCTION WITH OVERFLOW
7229	TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO
7235	TEST OF ALL FIS AT ONCE
7240	ADDRESS ERROR TEST
7248	INTERUPT ABORT TEST SECTION
7258	END OF PASS ROUTINE
7268	SCOPE ROUTINE
7270	PUSH AND POP SUBROUTINES
7445	HLT ROUTINE (ERROR TYPEOUT)
7451	USER ERROR ROUTINE
7463	OCTAL WORD & ADDRESS TYPER
7465	POWER DOWN AND UP ROUTINES
7467	ASCIZ TYPE OUT ROUTINE

C01

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52
DVKACA.SRC

SEQ 0002

5046

5048
5049
5055
5056
5057
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
000001
160000

```

ABS
MCALL SEOP,HEADER,SACT11,.SAPTBL5,.SAPTHDR,.SETUP,STARS
MCALL PUSH,POP,.SPOWER
TITLE DVKACA
*COPYRIGHT (C) AUGUST 1975
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY PERVEZ ZAKI
*
*THIS PROGRAM WAS ASSEMBLED USING THE POP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-B),JULY 11,1975.
*
$TN=1
$SHR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

```

5058
5065
5069
5083
5084
5098
5131
5136
5141
5149
5180
5225
5278
5282
5283
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

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

```

ERROR MESSAGE FORMAT:
ERRNM  ADR  PSW  SP  ANS1 ANS2 ANS3 ANS4 ANS5 ANS6

```

```

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

```

5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-2
 DVKACA.SRC SWITCH OPTIONS AND ASSIGNMENTS

SEQ 0004

5296	104000	HLT=	ENT	
5297	000000	R0=	X0	
5298	000001	R1=	X1	
5299	000002	R2=	X2	
5300	000003	R3=	X3	
5301	000004	R4=	X4	
5302	000005	R5=	X5	
5303	000005	TTY=	X5	
5304	000006	SP=	X6	
5305	000007	PC=	X7	
5306	000024	PWRVEC=	24	
5307	104400	SCOPE=	TRAP	
5308	100000	SW15=	100000	
5309	040000	SW14=	40000	
5310	020000	SW13=	20000	
5311	010000	SW12=	10000	
5312	004000	SW11=	4000	
5313	002000	SW10=	2000	
5314	001000	SW09=	1000	
5315	000400	SW08=	400	
5316	000004	TYPE=	IOT	
5317	000001	N=	1	
5318	000001	SF=	1	
5319		;*****		
5320				
5321	000000	. =	0	;TRAP CATCHER FROM 0 - 776
5329		;*****		
5330				
(1)				
(1)				
(1)				
(1)	001000			
(1)	000046			
(1)	000046			
(1)	015042			
(1)	000052			
(1)	000052			
(1)	000000			
(1)	001000			
5331				
5332				

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

5343		000432			SPSW: . = HLTADS+2	
5344	000432					;PROCESSOR STATUS WORD
5345		000434			SSP: . = SPSW+2	
5346	000434					;STACK POINTER
5347		000436			ANS1: . = SSP+2	
5348	000436					;FIRST ANSWER (SEE CODE)
5349		000440			ANS2: . = ANS1+2	
5350	000440					
5351		000442			ANS3: . = ANS2+2	
5352	000442					
5353		000444			ANS4: . = ANS3+2	
5354	000444	000000			ANS5: 0	
5355	000446	000000			ANS6: 0	
5356	000450	000000				
5357	000452	000000	000000	000000	0,0,0,0	;NON-%6 STACK BUFFER
	000460	000000				
5358	000462	000000			ERRORS: 0	
5359	000464	000244			FISVEC: 244	;FIS TRAP VECTOR ADDRESS
5360	000466	000246			FISLVL: 246	
5361	000470	000000			LADS: 0	
5362	000472	006412	000		RETURN: .ASCIZ <12><15>	;RETURN AND LINEFEED
5363	000475	015	020012	020040	SPACE: .ASCIZ <15><12>"	;RETURN AND 3 SPACES
	000502	000				
5364	000503	000				
5365					SICNT: .BYTE 0	
5366	000504	000007			.EVEN	
5367	000506	000000			SBELL: .WORD 7	;RING A BELL
5368	000510	000000			SAVTPS: 0	;LOC TO SAVE TELEPRINTER STATUS
5369	000512	000000			STACK0: 0	;NON-%6 STACK NORMAL LIMIT
5370	000514	000000			STACK2: 0	
5371	000516	000000			STACK4: 0	
5372	000520	000000	000000	000000	STACK6: 0	
	000526	000000	000000		STACK8: 0,1,0,0,0	;NON-%6 STACK BUFFER
5373	000532	000000				
5374		000511			STAK10: 0	
5375	000534	000000			STACK1 = STACK0+1	
5376	000536	000000			TEMP: 0	
5377	000540	000000			TIME3: 0	
5378	000542	000006			TYPCNT: 0	
5379	000544	000000			YESRT: RTT	;RETURN FROM TRACE TRAP
5380	000546	000064			.PR: 0	;COUNT AND SWITCH
5381	000550	177564			TTYOUT: 64	
5382	000552	177566			STPS: 177564	;TTY PRINTER STATUS REG.
					STPB: 177566	;TTY PRINTER BUFFER REG.

```

7104 ;*****
7105
7106 .SBTTL STARTING OF THE PROGRAM
7107
7108
7109      000200      000001      000330      .=      200
7110 000200 012767 000001 000330      MOV      #1,TIMES      ;NUMBER OF ITERATIONS IN THE FIRST PASS=1
7111 000206 012700 000410      MOV      #SDEVCT,RO      ;PREPARE TO INITIALIZE THE PROGRAM
7112 000212 005040      2$:      CLR      -(RO)
7113 000214 022700 000400      CMP      #SMAIL,RO
7114 000220 001374      BNE      2$
7115
7116 000222 000167 000352      RESTRT: JMP      BEGIN      ;JUMP TO STARTING ADDRESS OF PROGRAM
7117
7118
7119
7120      000600      .=      600
7121
7122 000600 012706 000600      BEGIN: MOV      #BEGIN, SP      ;INITIALIZE STACK POINTER
7123 000604 012737 000542 000014      MOV      #YESRT, @#14      ;SET TRACE TRAP VECTOR
7124 000612 012737 016346 000020      MOV      #STYPE, @#20      ;SET UP VECTOR 20
7125 000620 012737 016206 000024      MOV      #SPWRDN, @#24      ;SERVICE POWER DOWN ROUTINE FOR ANY FUTURE
7126                                     ;POWER DOWN
7127 000626 012700 000030      MOV      #30, RO      ;SET RO TO VECTOR 30
7128 000632 012720 015642      MOV      #HLTS, (0)+      ;SET EMT VECTOR
7129 000636 012720 000340      MOV      #340, (0)+
7130 000642 012720 015074      MOV      #SCOPES, (0)+      ;SET TRAP VECTOR
7131 000646 012710 000340      MOV      #340, (0)
7132 000652 012737 000006 000004 1$: MOV      #6, @#4      ;RESTORE TIME-OUT VECTOR
7133 000660 132737 000001 000420      BITB      #1,@#SENV      ;ARE WE UNDER APT ?
7134 000666 001410      BEQ      2$      ;IF NOT THEN GO TO 2$
7135 000670 012700 000554      MOV      #STPB+2,RO      ;OTHERWISE SET FOR THE OTHER SLU
7136 000674 012740 176566      MOV      #176566,-(RO)
7137 000700 012740 176564      MOV      #176564,-(RO)
7138 000704 012740 000074      MOV      #74,-(RO)
7139 000710 005067 177470      2$:      CLR      $TESTN
7140 000714 005067 177550      CLR      LAOS      ;CLEAR LOOP ADDRESS
7141
7142

```

7147

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 000720 104400
(1) 000722 004567 014436
(1) 000726 000000 000000
(1) 000732 000000 000000
(1) 000736 000000
(1) 000740 015634 000340
(1) 000744 012700 000510
(1)
(1) 000750 000240
(1) 000752 075000
(1)
(1) 000754 004767 014436
(1) 000760 010067 177450
(1) 000764 122767 000004 177440
(1) 000772 001402
(1) 000774 104000
(3) 000776 000001
(1)
(1) 001000 022767 000514 177426
(1) 001006 001402
(1) 001010 104000
(3) 001012 000002
(1)
(1) 001014 005767 177416
(1) 001020 001402
(1) 001022 104002
(3) 001024 000003
(1)
(1) 001026 005767 177406
(1) 001032 001402
(1) 001034 104002
(3) 001036 000004
(1)
(1) 001040 122767 000001 177336 END1:
(1) 001046 001402
(1) 001050 104000
(3) 001052 000005
(1)
(1)

```

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

TST1: SCOPE
JSR RS, PUSH4 ; 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, SSP ; SAVE "STACK POINTER"
CMPB #004, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 004
1 ; THE ERROR NUMBER IS 1

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

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

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

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

7148

```

(1)
(1)
(1)
(1)
(1)
(1)
(1) 001054 104400
(1) 001056 004567 014302
(1) 001062 040200 000000
(1) 001066 040200 000000
(1) 001072 000040
(1) 001074 015634 000340
(1) 001100 012701 000510
(1)
(1) 001104 000240
(1) 001106 075001
(1)
(1) 001110 004767 014302
(1) 001114 010167 177314
(1) 001120 105767 177306
(1) 001124 001402
(1) 001126 104000
(3) 001130 000006
(1)
(1) 001132 022767 000514 177274
(1) 001140 001402
(1) 001142 104000
(3) 001144 000007
(1)
(1) 001146 022767 040400 177262
(1) 001154 001402
(1) 001156 104002
(3) 001160 000010
(1)
(1) 001162 005767 177252
(1) 001166 001402
(1) 001170 104002
(3) 001172 000011
(1)
(1) 001174 122767 000002 177202 END2:
(1) 001202 001402
(1) 001204 104000
(3) 001206 000012
(1)
(1)

```

TEST 2: FADD (LSI-11 FLOATING ADD INSTRUCTION)
040200,000000 + 040200,000000 = 040400,000000
PS = 000, STACK POINTER = R1

```

TST2: SCOPE
JSR RS, PUSH4 ; 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, SSP ; SAVE "STACK POINTER"
TSTB SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 000
6 ; THE ERROR NUMBER IS 6

CMP #STACK4,SSP ; CHECK THE STACK POINTER (R1)
BEQ .+6 ; BRANCH IF OK
HLT ; STACK POINTER (R1) NOT EQUAL TO #STACK4
7 ; THE ERROR NUMBER IS 7

CMP #040400,ANS1 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 040400
10 ; THE ERROR NUMBER IS 10

TST ANS2 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 000000
11 ; THE ERROR NUMBER IS 11

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

```

7149

```

(1)
(1)
(1)
(1)
(1)
(1)
(1) 001210 104400
(1) 001212 004567 014146
(1) 001216 177777 177777
(1) 001222 077777 177777
(1) 001226 000100
(1) 001230 015634 000340
(1) 001234 012702 000510
(1)
(1) 001240 070240
(1) 001242 075002
(1)
(1) 001244 004767 014146
(1) 001250 010267 177160
(1) 001254 122767 000004 177150
(1) 001262 001402
(1) 001264 104000
(3) 001266 000013
(1)
(1) 001270 022767 000514 177136
(1) 001276 001402
(1) 001300 104000
(3) 001302 000014
(1)
(1) 001304 005767 177126
(1) 001310 001402
(1) 001312 104002
(3) 001314 000015
(1)
(1) 001316 005767 177116
(1) 001322 001402
(1) 001324 104002
(3) 001326 000016
(1)
(1) 001330 122767 000003 177046 END3:
(1) 001336 001402
(1) 001340 104000
(3) 001342 000017
(1)
(1)

```

```

*****
TEST 3: FADD (LSI-11 FLOATING ADD INSTRUCTION)
077777,177777 + 177777,177777 = 000000,000000
PS = 004, STACK POINTER = R2
*****

TST3: SCOPE
JSR RS, PUSHR ; PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; FIRST OPERAND ON BOTTOM
; PROCESSOR PRIORITY LEVEL
; FIS TRAP VECTOR
; SET UP STACK POINTER
; FLOATING ADD ON THE R2 STACK

MOV #004, $SP
MOV #004, $PSW
CMPB #004, $PSW
BEQ .+6
HLT 13
; POP THE ANSWER
; SAVE "STACK POINTER"
; CHECK PS (EXCEPT T BIT)
; BRANCH IF OK
; PS NOT EQUAL TO 004
; THE ERROR NUMBER IS 13

CMP #STACK4, $SP
BEQ .+6
HLT 14
; CHECK THE STACK POINTER (R2)
; BRANCH IF OK
; STACK POINTER (R2) NOT EQUAL TO #STACK4
; THE ERROR NUMBER IS 14

TST ANS1
BEQ .+6
HLT+2 15
; CHECK FIRST HALF OF ANSWER
; BRANCH IF OK
; ANS1 NOT EQUAL TO 000000
; THE ERROR NUMBER IS 15

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

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

```



```

7151
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 001476 104400
(1) 001500 004567 013506
(1) 001504 025177 177777
(1) 001510 125200 000000
(1) 001514 000307
(1) 001516 015634 000340
(1)
(1) 001522 000240
(1) 001524 075006
(1)
(1) 001526 004767 013520
(1) 001532 022706 000600
(1) 001536 001405
(1) 001540 012706 000600
(1) 001544 104000
(3) 001546 000025
(1) 001550 000421
(1)
(1) 001552 122767 000210 176652 TSAS:
(1) 001560 001402
(1) 001562 104000
(3) 001564 000026
(1)
(1) 001566 022767 117200 176642
(1) 001574 001402
(1) 001576 104002
(3) 001600 000027
(1)
(1) 001602 005767 176632
(1) 001606 001402
(1) 001610 104002
(3) 001612 000030
(1)
(1) 001614 122767 000005 176562 ENDS:
(1) 001622 001402
(1) 001624 104000
(3) 001626 000031
(1)
(1)

```

 TEST 5: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 125200,000000 + 025177,177777 = 117200,000000
 PS = 210, STACK POINTER = SP

SCOPE
 TSTS: JSR RS, PUSH5 ; PUSH 4 WORDS ONTO STACK, SET PRIORITY
 .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 TSAS ; 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
 CMPB #210, SPSW ; CHECK PS (EXCEPT T BIT)
 BEQ .+6 ; BRANCH IF OK
 HLT ; PS NOT EQUAL TO 210
 26 ; THE ERROR NUMBER IS 26
 CMP #117200,ANS1 ; CHECK FIRST HALF OF ANSWER
 BEQ .+6 ; BRANCH IF OK
 HLT+2 ; ANS1 NOT EQUAL TO 117200
 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
 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

7152

```

(1)
(1)
(1)
(1)
(1)
(1)
(1) 001630 104400
(1) 001632 004567 013526
(1) 001636 100125 052525
(1) 001642 135753 024642
(1) 001646 000347
(1) 001650 015634 000340
(1) 001654 012705 000510
(1)
(1) 001660 000240
(1) 001662 075005
(1)
(1) 001664 004767 013526
(1) 001670 010567 176540
(1) 001674 122767 000210 176530
(1) 001702 001402
(1) 001704 104000
(3) 001706 000032
(1)
(1) 001710 022767 000514 176516
(1) 001716 001402
(1) 001720 104000
(3) 001722 000033
(1)
(1) 001724 022767 135753 176504
(1) 001732 001402
(1) 001734 104002
(3) 001736 000034
(1)
(1) 001740 022767 024642 176472
(1) 001746 001402
(1) 001750 104002
(3) 001752 000035
(1)
(1) 001754 122767 000006 176422 END6:
(1) 001762 001402
(1) 001764 104000
(3) 001766 000036
(1)
(1)

```

```

*****
TEST 6:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
              135753,024642 + 100125,052525 = 135753,024642
              PS = 210,      STACK POINTER = R5
*****

TST6:  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,      $SP      ; SAVE "STACK POINTER"
      CMPB     #210,      $PSW    ; CHECK PS (EXCEPT T BIT)
      BEQ      .+6               ; BRANCH IF OK
      HLT      .+6               ; PS NOT EQUAL TO 210
      32                      ; THE ERROR NUMBER IS 32

      CMP      #STACK4,$SP       ; CHECK THE STACK POINTER (R5)
      BEQ      .+6               ; BRANCH IF OK
      HLT      .+6               ; STACK POINTER (R5) NOT EQUAL TO #STACK4
      33                      ; THE ERROR NUMBER IS 33

      CMP      #135753,ANS1      ; CHECK FIRST HALF OF ANSWER
      BEQ      .+6               ; BRANCH IF OK
      HLT+2    .+6               ; ANS1 NOT EQUAL TO 135753
      34                      ; THE ERROR NUMBER IS 34

      CMP      #024642,ANS2      ; CHECK SECOND HALF OF ANSWER
      BEQ      .+6               ; BRANCH IF OK
      HLT+2    .+6               ; ANS2 NOT EQUAL TO 024642
      35                      ; THE ERROR NUMBER IS 35

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

```


7155

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 002266 104400
(1) 002270 004567 013070
(1) 002274 100252 125252
(1) 002300 000425 052525
(1) 002304 000217
(1) 002306 015634 000340
(1) 002312 012704 000510
(1)
(1) 002316 000240
(1) 002320 075004
(1)
(1) 002322 004767 013070
(1) 002326 010467 176102
(1) 002332 122767 000200 176072
(1) 002340 001402
(1) 002342 104000
(3) 002344 000051
(1)
(1) 002346 022767 000514 176060
(1) 002354 001402
(1) 002356 104000
(3) 002360 000052
(1)
(1) 002362 022767 000200 176046
(1) 002370 001402
(1) 002372 104002
(3) 002374 000053
(1)
(1) 002376 005767 176036
(1) 002402 001402
(1) 002404 104002
(3) 002406 000054
(1)
(1) 002410 122767 000011 175766 END11:
(1) 002416 001402
(1) 002420 104000
(3) 002422 000055
(1)
(1)

```

```

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

TST11:  SCOPE
        JSR      RS,      PUSHR      ; PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
        .WORD    100252,125252      ; SECOND OPERAND ON TOP
        .WORD    000425,052525      ; FIRST OPERAND ON BOTTOM
        .WORD    217                ; PROCESSOR PRIORITY LEVEL
        .WORD    TRAPER,340         ; FIS TRAP VECTOR
        MOV      #STACK0,R4         ; SET UP STACK POINTER

        NOP
        FADD     R4                 ; FLOATING ADD ON THE R4 STACK

        JSR      PC,      POPR      ; POP THE ANSWER
        MOV      R4,      $SP      ; SAVE "STACK POINTER"
        CMPB     #200,      $PSW    ; CHECK PS (EXCEPT T BIT)
        BEQ      .+6               ; BRANCH IF OK
        HLT      .                ; PS NOT EQUAL TO 200
        51                      ; THE ERROR NUMBER IS 51

        CMP      #STACK4,$SP        ; CHECK THE STACK POINTER (R4)
        BEQ      .+6               ; BRANCH IF OK
        HLT      .                ; STACK POINTER (R4) NOT EQUAL TO #STACK4
        52                      ; THE ERROR NUMBER IS 52

        CMP      #000200,ANS1       ; CHECK FIRST HALF OF ANSWER
        BEQ      .+6               ; BRANCH IF OK
        HLT+2    .                ; ANS1 NOT EQUAL TO 000200
        53                      ; THE ERROR NUMBER IS 53

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

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

```

7156

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 002424 104400
(1) 002426 004567 012560
(1) 002432 000252 125252
(1) 002436 100425 052525
(1) 002442 000307
(1) 002444 015634 000340
(1)
(1) 002450 000240
(1) 002452 075006
(1)
(1) 002454 004767 012572
(1) 002460 022706 000600
(1) 002464 001405
(1) 002466 012706 000600
(1) 002472 104000
(3) 002474 000056
(1) 002476 000421
(1)
(1) 002500 122767 000210 175724 TSA12:
(1) 002506 001402
(1) 002510 104000
(3) 002512 000057
(1)
(1) 002514 022767 100200 175714
(1) 002522 001402
(1) 002524 104002
(3) 002526 000060
(1)
(1) 002530 005767 175704
(1) 002534 001402
(1) 002536 104002
(3) 002540 000061
(1)
(1) 002542 122767 000012 175634 END12:
(1) 002550 001402
(1) 002552 104000
(3) 002554 000062
(1)
(1)

```

 TEST 12: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 100425,052525 + 000252,125252 = 100200,000000
 PS = 210, STACK POINTER = SP

SCOPE
 TST12: JSR RS
 .WORD 000252,125252
 .WORD 100425,052525
 .WORD 307
 .WORD TRAPER,340
 PUSH5 : PUSH 4 WORDS ONTO STACK, SET PRIORITY
 : SECOND OPERAND ON TOP
 : FIRST OPERAND ON BOTTOM
 : PROCESSOR PRIORITY LEVEL
 : 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 TSA12 : BRANCH IF OK
 MOV #BEGIN, SP : RESTORE STACK POINTER
 HLT : STACK POINTER FOULED UP
 S6 : THE ERROR NUMBER IS 56
 BR END12 : SKIP REST OF TEST
 CMPB #210, SPSW : CHECK PS (EXCEPT T BIT)
 BEQ .+6 : BRANCH IF OK
 HLT : PS NOT EQUAL TO 210
 S7 : THE ERROR NUMBER IS 57
 CMP #100200,ANS1 : CHECK FIRST HALF OF ANSWER
 BEQ .+6 : BRANCH IF OK
 HLT+2 : ANS1 NOT EQUAL TO 100200
 B0 : THE ERROR NUMBER IS 60
 TST ANS2 : CHECK SECOND HALF OF ANSWER
 BEQ .+6 : BRANCH IF OK
 HLT+2 : ANS2 NOT EQUAL TO 000000
 B1 : THE ERROR NUMBER IS 61
 CMPB #12, \$TESTN : CHECK THE TEST NUMBER
 BEQ .+6 : BRANCH IF OK
 HLT : WRONG TEST! PC MUST HAVE FOULED UP.
 B2 : THE ERROR NUMBER IS 62

7157

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 002556 104400
(1) 002557 012426
(1) 002558 077452 125252
(1) 002570 077652 125252
(1) 002574 000257
(1) 002576 015634 000340
(1)
(1) 002602 000240
(1) 002604 075006
(1)
(1) 002606 004767 012440
(1) 002612 022706 000600
(1) 002616 001405
(1) 002620 012706 000600
(1) 002624 104000
(3) 002626 000063
(1) 002630 000422
(1)
(1) 002632 122767 000200 175572 TSA13:
(1) 002640 001402
(1) 002642 104000
(3) 002644 000064
(1)
(1) 002646 022767 077777 175562
(1) 002654 001402
(1) 002656 104002
(3) 002660 000065
(1)
(1) 002662 022767 177777 175550
(1) 002670 001402
(1) 002672 104002
(3) 002674 000066
(1)
(1) 002676 122767 000013 175500 END13:
(1) 002704 001402
(1) 002706 104000
(3) 002710 000067
(1)
(1)

```

 TEST 13: FADD (LSI-11 FLOATING ADD INSTRUCTION)
 077652,125252 + 077452,125252 = 077777,177777
 PS = 200, STACK POINTER = SP

SCOPE
 TST13: JSR RS PUSH4 : PUSH 4 WORDS ONTO STACK, SET PRIORITY
 .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 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

CMPB #200, SPSW ; CHECK PS (EXCEPT T BIT)
 BEQ .+6 ; BRANCH IF OK
 HLT ; PS NOT EQUAL TO 200
 64 ; THE ERROR NUMBER IS 64

CMP #077777,ANS1 ; CHECK FIRST HALF OF ANSWER
 BEQ .+6 ; BRANCH IF OK
 HLT+2 ; ANS1 NOT EQUAL TO 077777
 65 ; THE ERROR NUMBER IS 65

CMP #177777,ANS2 ; CHECK SECOND HALF OF ANSWER
 BEQ .+6 ; BRANCH IF OK
 HLT+2 ; ANS2 NOT EQUAL TO 177777
 66 ; THE ERROR NUMBER IS 66

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

7158

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 002712 104400
(1) 002714 004567 012444
(1) 002720 177652 125252
(1) 002724 177452 125252
(1) 002730 000357
(1) 002732 015634 000340
(1) 002736 012704 000510
(1)
(1) 002742 000240
(1) 002744 075004
(1)
(1) 002746 004767 012444
(1) 002752 010467 175456
(1) 002756 122767 000210 175446
(1) 002764 001402
(1) 002766 104000
(3) 002770 000070
(1)
(1) 002772 022767 000514 175434
(1) 003000 001402
(1) 003002 104000
(3) 003004 000071
(1)
(1) 003006 022767 177777 175422
(1) 003014 001402
(1) 003016 104002
(3) 003020 000072
(1)
(1) 003022 022767 177777 175410
(1) 003030 001402
(1) 003032 104002
(3) 003034 000073
(1)
(1) 003036 122767 000014 175340 END14:
(1) 003044 001402
(1) 003046 104000
(3) 003050 000074
(1)
(1)

```

TEST 14: FADD (LSI-11 FLOATING ADD INSTRUCTION)
177452,125252 + 177652,125252 = 177777,177777
PS = 210, STACK POINTER = R4

TST14: SCOPE
JSR R5, FUSHR ; PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; FIRST OPERAND ON BOTTOM
; PROCESSOR PRIORITY LEVEL
; FIS TRAP VECTOR
; SET UP STACK POINTER
; FLOATING ADD ON THE R4 STACK
; POP THE ANSWER
; SAVE "STACK POINTER"
; CHECK PS (EXCEPT T BIT)
; BRANCH IF OK
; PS NOT EQUAL TO 210
; THE ERROR NUMBER IS 70
; CHECK THE STACK POINTER (R4)
; BRANCH IF OK
; STACK POINTER (R4) NOT EQUAL TO #STACK4
; THE ERROR NUMBER IS 71
; CHECK FIRST HALF OF ANSWER
; BRANCH IF OK
; ANS1 NOT EQUAL TO 177777
; THE ERROR NUMBER IS 72
; CHECK SECOND HALF OF ANSWER
; BRANCH IF OK
; ANS2 NOT EQUAL TO 177777
; THE ERROR NUMBER IS 73
; CHECK THE TEST NUMBER
; BRANCH IF OK
; WRONG TEST! PC MUST HAVE FOULED UP.
; THE ERROR NUMBER IS 74

[illegible]

```

*****
TEST 15:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
              004000,105004 + 104000,104000 = 000401,000000
              PS = 000,          STACK POINTER = PC
*****

TST15:  SCOPE
        JSR      R5,      PUSH7    ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
        .WORD    STK15           ;TOP OF STACK
        .WORD    104000,104000   ;SECOND OPERAND ON TOP
        .WORD    004000,105004   ;FIRST OPERAND ON BOTTOM
        .WORD    144             ;PROCESSOR PRIORITY LEVEL
        .WORD    TRAPER,340      ;FIS TRAP VECTOR

        NOP
        FADD     PC              ;FLOATING ADD ON FOLLOWING 4 WORDS
STK15:   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     SPSW            ;CHECK PS (EXCEPT T BIT)
        BEQ      .+6             ;BRANCH IF OK
        HLT      75              ;PS NOT EQUAL TO 000
                                   ;THE ERROR NUMBER IS 75

        CMP      #104000,ANS1    ;CHECK FIRST HALF OF INPUT DATA (STK15)
        BEQ      .+6             ;BRANCH IF OK
        HLT+2    76              ;ANS1 NOT EQUAL TO 104000
                                   ;THE ERROR NUMBER IS 76

        CMP      #104000,ANS2    ;CHECK SECOND HALF OF INPUT DATA (STK15+2)
        BEQ      .+6             ;BRANCH IF OK
        HLT+2    77              ;ANS2 NOT EQUAL TO 104000
                                   ;THE ERROR NUMBER IS 77

        CMP      #000401,ANS3    ;CHECK FIRST HALF OF ANSWER
        BEQ      .+6             ;BRANCH IF OK
        HLT+4    100             ;ANS3 NOT EQUAL TO 000401
                                   ;THE ERROR NUMBER IS 100

        TST      ANS4            ;CHECK SECOND HALF OF ANSWER
        BEQ      .+6             ;BRANCH IF OK
        HLT+4    101             ;ANS4 NOT EQUAL TO 000000
                                   ;THE ERROR NUMBER IS 101

END15:   CMPB     #15,      $TESTN ;CHECK THE TEST NUMBER
        BEQ      .+6             ;BRANCH IF OK
        HLT      102             ;WRONG TEST! PC MUST HAVE FOULED UP.
                                   ;THE ERROR NUMBER IS 102

```

• *****

(1)					: TEST 16:	FADD (LSI-1) FLOATING ADD INSTRUCTION)
(1)					:	100200,000000 + 000377,177777 ==> UNDERFLOW
(1)					:	PS(ON STACK) = 012, STACK POINTER = R3
(1)					:	*****
(1)						
(1)	003224	104400			TST16:	SCOPE
(1)	003226	004567	012132		JSR	R5, PUSHR ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
(1)	003232	000377	177777		.WORD	000377,177777 ;SECOND OPERAND ON TOP
(1)	003236	100200	000000		.WORD	100200,000000 ;FIRST OPERAND ON BOTTOM
(1)	003242	000157			.WORD	157 ;PROCESSOR PRIORITY LEVEL
(1)	003244	003276	000000		.WORD	ISR16, 000 ;FIS TRAP VECTOR
(1)	003250	012703	000510		MOV	#STACK0,R3 ;SET UP R3 AS STACK POINTER
(1)						
(1)	003254	000240			NOP	
(1)	003256	075003			FADD	R3 ;FLOATING ADD ON THE R3 STACK
(1)						
(1)	003260	004767	012132		RTA16:	JSR PC, POPR ;POP THE "ANSWER"
(1)	003264	010367	175144		MOV	R3, \$SP ;SAVE STACK POINTER (R3)
(1)	003270	104002			HLT+2	;FIS TRAP DIDN'T OCCURE!
(3)	003272	000103			103	;THE ERROR NUMBER IS 103
(1)	003274	000462			BR	END16
(1)						
(1)	003276	004767	012144		ISR16:	JSR PC, POPER ;POP ALL DATA OFF THE STACKS
(1)	003302	010367	175126		MOV	R3, \$SP ;SAVE STACK POINTER (R3)
(1)	003306	105767	175120		TSTB	\$PSW ;CHECK PS AFTER FIS TRAP
(1)	003312	001402			BEQ	+6 ;BRANCH IF OK
(1)	003314	104000			HLT	;PS AFTER FIS TRAP NOT EQUAL TO 000
(3)	003316	000104			104	;THE ERROR NUMBER IS 104
(1)						
(1)	003320	022767	000510	175106	CMP	#STACK0,\$SP ;CHECK THE STACK POINTER (R3)
(1)	003326	001402			BEQ	+6 ;BRANCH IF OK
(1)	003330	104000			HLT	;STACK POINTER (R3) NOT EQUAL TO #STACK0
(3)	003332	000105			105	;THE ERROR NUMBER IS 105
(1)						
(1)	003334	022767	003260	175074	CMP	#RTA16, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1)	003342	001402			BEQ	+6 ;BRANCH IF OK
(1)	003344	104001			HLT+1	;FIS TRAP AT WRONG ADDRESS
(3)	003346	000106			106	;THE ERROR NUMBER IS 106
(1)						
(1)	003350	022767	000012	175062	CMP	#012, ANS2 ;CHECK PS BEFORE FIS TRAP
(1)	003356	001402			BEQ	+6 ;BRANCH IF OK
(1)	003360	104002			HLT+2	;PS AT FIS TRAP TIME NOT 012
(3)	003362	000107			107	;THE ERROR NUMBER IS 107
(1)						
(1)	003364	022767	000377	175050	CMP	#000377,ANS3 ;CHECK DATA FROM THE STACK
(1)	003372	001402			BEQ	+6 ;BRANCH IF OK
(1)	003374	104004			HLT+4	;DATA ON STACK (000377) CHANGED
(3)	003376	000110			110	;THE ERROR NUMBER IS 110
(1)						
(1)	003400	022767	177777	175036	CMP	#177777,ANS4 ;CHECK DATA FROM STACK
(1)	003406	001402			BEQ	+6 ;BRANCH IF OK
(1)	003410	104004			HLT+4	;DATA ON STACK (177777) CHANGED
(3)	003412	000111			111	;THE ERROR NUMBER IS 111
(1)						
(1)	003414	022767	100200	175024	CMP	#100200,ANS5 ;CHECK DATA FROM STACK
(1)	003422	001402			BEQ	+6 ;BRANCH IF OK

```
(1) 003424 104006          HLT+6           ;DATA ON STACK (100200) CHANGED  
(3) 003426 000112         112             ;THE ERROR NUMBER IS 112  
(1)  
(1) 003430 005767   175014      TST        ANS6       ;CHECK DATA FROM STACK  
(1) 003434 001402          BEQ        .+6            ;BRANCH IF OK  
(1) 003436 104006          HLT+6           ;DATA ON STACK (000000) CHANGED  
(3) 003440 000113         113             ;THE ERROR NUMBER IS 113  
(1)  
(1) 003442 122767   000016   174734   END16: CMPB     #16,    $TESTN    ;CHECK THE TEST NUMBER  
(1) 003450 001402          BEQ        .+6            ;BRANCH IF OK  
(1) 003452 104000          HLT              ;WRONG TEST! PC MUST HAVE FOULED UP.  
(3) 003454 000114         114             ;THE ERROR NUMBER IS 114  
(1)  
(1)
```

7164

PC	OP	OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-22
 DVKACA.SRC TEST FLOATING ADD INSTRUCTION WITH UNDERFLOW

SEQ 0024

(1)	003644	022767	000200	174574	CMP	#000200,ANS5	;CHECK DATA FROM STACK
(1)	003652	001402			BEQ	+.6	;BRANCH IF OK
(1)	003654	104006			HLT+6		;DATA ON STACK (000200) CHANGED
(3)	003656	000124			124		;THE ERROR NUMBER IS 124
(1)							
(1)	003660	005767	174564		TST	ANS6	;CHECK DATA FROM STACK
(1)	003664	001402			BEQ	+.6	;BRANCH IF OK
(1)	003666	104006			HLT+6		;DATA ON STACK (000000) CHANGED
(3)	003670	000125			125		;THE ERROR NUMBER IS 125
(1)							
(1)	003672	122767	000017	174504	END17:	CMPB	#17, \$TESTN ;CHECK THE TEST NUMBER
(1)	003700	001402			BEQ	+.6	;BRANCH IF OK
(1)	003702	104000			HLT		;WRONG TEST! PC MUST HAVE FOULED UP.
(3)	003704	000126			126		;THE ERROR NUMBER IS 126
(1)							
(1)							

7168

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 003706 104400
(1) 003710 004567 011450
(1) 003714 177652 125252
(1) 003720 177452 125253
(1) 003724 000105
(1) 003726 003760 000252
(1) 003732 012701 000510
(1)
(1) 003736 000240
(1) 003740 075001
(1)
(1) 003742 004767 011450
(1) 003746 010167 174462
(1) 003752 104002
(3) 003754 000127
(1) 003756 000464
(1)
(1) 003760 004767 011462
(1) 003764 010167 174444
(1) 003770 122767 000252 174434
(1) 003776 001402
(1) 004000 104000
(3) 004002 000130
(1)
(1) 004004 022767 000510 174422
(1) 004012 001402
(1) 004014 104000
(3) 004016 000131
(1)
(1) 004020 022767 003742 174410
(1) 004026 001402
(1) 004030 104001
(3) 004032 000132
(1)
(1) 004034 022767 000002 174376
(1) 004042 001402
(1) 004044 104002
(3) 004046 000133
(1)
(1) 004050 022767 177652 174364
(1) 004056 001402
(1) 004060 104004
(3) 004062 000134
(1)
(1) 004064 022767 125252 174352
(1) 004072 001402
(1) 004074 104004
(3) 004076 000135
(1)

*****
TEST 20:      FADD (LSI-11 FLOATING ADD INSTRUCTION)
              177452,125253 + 177652,125252 ==> OVERFLOW
              PS(ON STACK) = 002,      STACK POINTER = R1
*****
SCOPE
TST20:  JSR      R5,      PUSH4      ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
        .WORD    177652,125252      ;SECOND OPERAND ON TOP
        .WORD    177452,125253      ;FIRST OPERAND ON BOTTOM
        .WORD    105                ;PROCESSOR PRIORITY LEVEL
        .WORD    ISR20, 252          ;FIS TRAP VECTOR
        MOV      #STACK0,R1          ;SET UP R1 AS STACK POINTER

        NOP
        FADD     R1                  ;FLOATING ADD ON THE R1 STACK

RTA20:  JSR      PC,      POPR        ;POP THE "ANSWER"
        MOV      R1,      SSP        ;SAVE STACK POINTER (R1)
        HLT+2     127                ;FIS TRAP DIDN'T OCCURE!
        BR        END20              ;THE ERROR NUMBER IS 127

ISR20:  JSR      PC,      POPER       ;POP ALL DATA OFF THE STACKS
        MOV      R1,      SSP        ;SAVE STACK POINTER (R1)
        CMPB     #252,  SPSW        ;CHECK PS AFTER FIS TRAP
        BEQ      .+6                ;BRANCH IF OK
        HLT      130                ;PS AFTER FIS TRAP NOT EQUAL TO 252
        130                          ;THE ERROR NUMBER IS 130

        CMP      #STACK0,SSP        ;CHECK THE STACK POINTER (R1)
        BEQ      .+6                ;BRANCH IF OK
        HLT      131                ;STACK POINTER (R1) NOT EQUAL TO #STACK0
        131                          ;THE ERROR NUMBER IS 131

        CMP      #RTA20, ANS1       ;CHECK FIS TRAP RETURN ADDRESS
        BEQ      .+6                ;BRANCH IF OK
        HLT+1     132                ;FIS TRAP AT WRONG ADDRESS
        132                          ;THE ERROR NUMBER IS 132

        CMP      #002,  ANS2        ;CHECK PS BEFORE FIS TRAP
        BEQ      .+6                ;BRANCH IF OK
        HLT+2     133                ;PS AT FIS TRAP TIME NOT 002
        133                          ;THE ERROR NUMBER IS 133

        CMP      #177652,ANS3       ;CHECK DATA FROM THE STACK
        BEQ      .+6                ;BRANCH IF OK
        HLT+4     134                ;DATA ON STACK (177652) CHANGED
        134                          ;THE ERROR NUMBER IS 134

        CMP      #125252,ANS4       ;CHECK DATA FROM STACK
        BEQ      .+6                ;BRANCH IF OK
        HLT+4     135                ;DATA ON STACK (125252) CHANGED
        135                          ;THE ERROR NUMBER IS 135

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-24
 DVKACA.SRC TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0026

(1)	004100	022767	177452	174340	CMP	#177452,ANS5	;CHECK DATA FROM STACK
(1)	004106	001402			BEQ	.+6	;BRANCH IF OK
(1)	004110	104006			HLT+6		;DATA ON STACK (177452) CHANGED
(3)	004112	000136			136		;THE ERROR NUMBER IS 136
(1)							
(1)	004114	022767	125253	174326	CMP	#125253,ANS6	;CHECK DATA FROM STACK
(1)	004122	001402			BEQ	.+6	;BRANCH IF OK
(1)	004124	104006			HLT+6		;DATA ON STACK (125253) CHANGED
(3)	004126	000137			137		;THE ERROR NUMBER IS 137
(1)							
(1)	004130	122767	000020	174246	END20:	CMPB	#20, \$TESTN
(1)	004136	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	004140	104000			HLT		;BRANCH IF OK
(3)	004142	000140			140		;WRONG TEST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 140
(1)							

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-26
 DVKACA.SRC TEST FLOATING ADD INSTRUCTION WITH OVERFLOW

SEQ 0028

(1)	004332	022767	077652	174106	CMP	#077652,ANS5	;CHECK DATA FROM STACK
(1)	004340	001402			BEQ	.+6	;BRANCH IF OK
(1)	004342	104006			HLT+6		;DATA ON STACK (077652) CHANGED
(3)	004344	000150			150		;THE ERROR NUMBER IS 150
(1)							
(1)	004346	022767	125253	174074	CMP	#125253,ANS6	;CHECK DATA FROM STACK
(1)	004354	001402			BEQ	.+6	;BRANCH IF OK
(1)	004356	104006			HLT+6		;DATA ON STACK (125253) CHANGED
(3)	004360	000151			151		;THE ERROR NUMBER IS 151
(1)							
(1)	004362	122767	000021	174014	END21:	CMPB	#21, \$TESTN
(1)	004370	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	004372	104000			HLT		;BRANCH IF OK
(3)	004374	000152			152		;WRONG TEST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 152
(1)							

(1)				
(1)				
(1)				
(1)				
(1)				
(1)				
(1)	004376	104400		
(1)	004400	004567	010760	
(1)	004404	135352	051107	
(1)	004410	177520	017552	
(1)	004414	000040		
(1)	004416	015634	000340	
(1)	004422	012701	000510	
(1)				
(1)	004426	000240		
(1)	004430	075011		
(1)				
(1)	004432	004767	010760	
(1)	004436	010167	173772	
(1)	004442	122767	000010	173762
(1)	004450	001402		
(1)	004452	104000		
(3)	004454	000153		
(1)				
(1)	004456	022767	000514	173750
(1)	004464	001402		
(1)	004466	104000		
(3)	004470	000154		
(1)				
(1)	004472	022767	177520	173736
(1)	004500	001402		
(1)	004502	104002		
(3)	004504	000155		
(1)				
(1)	004506	022767	017552	173724
(1)	004514	001402		
(1)	004516	104002		
(3)	004520	000156		
(1)				
(1)	004522	122767	000022	173654
(1)	004530	001402		
(1)	004532	104000		
(3)	004534	000157		
(1)				
(1)				

```

*****
TEST 22:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
              177520,017552 - 135352,051107 = 177520,017552
              PS = 010,      STACK POINTER = R1
*****

TST22:  SCOPE
        JSR      RS,      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      .+6              ; PS NOT EQUAL TO 010
        153                    ; THE ERROR NUMBER IS 153

        CMP      #STACK4,$SP      ; CHECK THE STACK POINTER (R1)
        BEQ      .+6              ; BRANCH IF OK
        HLT      .+6              ; STACK POINTER (R1) NOT EQUAL TO #STACK4
        154                    ; THE ERROR NUMBER IS 154

        CMP      #177520,ANS1     ; CHECK FIRST HALF OF ANSWER
        BEQ      .+6              ; BRANCH IF OK
        HLT+2    .+6              ; ANS1 NOT EQUAL TO 177520
        155                    ; THE ERROR NUMBER IS 155

        CMP      #017552,ANS2     ; CHECK SECOND HALF OF ANSWER
        BEQ      .+6              ; BRANCH IF OK
        HLT+2    .+6              ; ANS2 NOT EQUAL TO 017552
        156                    ; THE ERROR NUMBER IS 156

END22:  CMPB     #22,      $TESTN  ; CHECK THE TEST NUMBER
        BEQ      .+6              ; BRANCH IF OK
        HLT      .+6              ; WRONG TEST! PC MUST HAVE FOULED UP.
        157                    ; THE ERROR NUMBER IS 157

```

```

(1) ;*****
(1) TEST 23: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
(1) 125252,125252 - 125252,125253 = 017400,000000
(1) PS = 000, STACK POINTER = R0
(1) ;*****
(1)
(1) 004536 104400
(1) 004540 004567 010620 TST23: JSR RS, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
(1) 004544 125252 125253 ;SECOND OPERAND ON TOP
(1) 004550 125252 125252 ;FIRST OPERAND ON BOTTOM
(1) 004554 000047 ;PROCESSOR PRIORITY LEVEL
(1) 004556 015634 000340 ;FIS TRAP VECTOR
(1) 004562 012700 000510 MOV #STACK0,R0 ;SET UP STACK POINTER
(1)
(1) 004566 000240
(1) 004570 075010 NOP
(1) FSUB R0 ;FLOATING SUBTRACT ON THE R0 STACK
(1)
(1) 004572 004767 010620 JSR PC, POPR ;POP THE ANSWER
(1) 004576 010067 173632 MOV R0, SSP ;SAVE "STACK POINTER"
(1) 004602 105767 173624 TSTB $PSW ;CHECK PS (EXCEPT T BIT)
(1) 004606 001402 BEQ .+6 ;BRANCH IF OK
(1) 004610 104000 HLT ;PS NOT EQUAL TO 000
(3) 004612 000160 160 ;THE ERROR NUMBER IS 160
(1)
(1) 004614 022767 000514 173612 CMP #STACK4,SSP ;CHECK THE STACK POINTER (R0)
(1) 004622 001402 BEQ .+6 ;BRANCH IF OK
(1) 004624 104000 HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK4
(3) 004626 000161 161 ;THE ERROR NUMBER IS 161
(1)
(1) 004630 022767 017400 173600 CMP #017400,ANS1 ;CHECK FIRST HALF OF ANSWER
(1) 004636 001402 BEQ .+6 ;BRANCH IF OK
(1) 004640 104002 HLT+2 ;ANS1 NOT EQUAL TO 017400
(3) 004642 000162 162 ;THE ERROR NUMBER IS 162
(1)
(1) 004644 005767 173570 TST ANS2 ;CHECK SECOND HALF OF ANSWER
(1) 004650 001402 BEQ .+6 ;BRANCH IF OK
(1) 004652 104002 HLT+2 ;ANS2 NOT EQUAL TO 000000
(3) 004654 000163 163 ;THE ERROR NUMBER IS 163
(1)
(1) 004656 122767 000023 173520 END23: CMPB #23, $TESTN ;CHECK THE TEST NUMBER
(1) 004664 001402 BEQ .+6 ;BRANCH IF OK
(1) 004666 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 004670 000164 164 ;THE ERROR NUMBER IS 164
(1)
(1)

```



```
(1) :*****  
(1) TEST 26: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)  
(1) : 177452,125252 - 077652,125252 = 177777,177777  
(1) PS = 210, STACK POINTER = SP  
(1) :*****  
(1)  
(1) 005164 104400 SCOPE  
(1) 005166 004567 010020 TST26: JSR RS, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY  
(1) 005172 077652 125252 .WORD 077652,125252 ;SECOND OPERAND ON TOP  
(1) 005176 177452 125252 .WORD 177452,125252 ;FIRST OPERAND ON BOTTOM  
(1) 005202 000357 .WORD 357 ;PROCESSOR PRIORITY LEVEL  
(1) 005204 015634 000340 .WORD TRAPER, 340 ;FIS TRAP VECTOR  
(1)  
(1) 005210 000240 NOP  
(1) 005212 075016 FSUB SP ;FLOATING SUBTRACT ON THE STACK  
(1)  
(1) 005214 004767 010032 JSR PC, POP5 ;POP THE ANSWER  
(1) 005220 022706 000600 CMP #BEGIN, SP ;CHECK THE STACK POINTER  
(1) 005224 001405 BEQ TSA26 ;BRANCH IF OK  
(1) 005226 012706 000600 MOV #BEGIN, SP ;RESTORE STACK POINTER  
(1) 005232 104000 HLT ;STACK POINTER FOULED UP  
(3) 005234 000177 177 BR ;THE ERROR NUMBER IS 177  
(1) 005236 000422 END26 ;SKIP REST OF TEST  
(1)  
(1) 005240 122767 000210 173164 TSA26: CMPSB #210, SPSW ;CHECK PS (EXCEPT T BIT)  
(1) 005246 001402 BEQ .+6 ;BRANCH IF OK  
(1) 005250 104000 HLT ;PS NOT EQUAL TO 210  
(3) 005252 000200 200 ;THE ERROR NUMBER IS 200  
(1)  
(1) 005254 022767 177777 173154 CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER  
(1) 005262 001402 BEQ .+6 ;BRANCH IF OK  
(1) 005264 104002 HLT+2 ;ANS1 NOT EQUAL TO 177777  
(3) 005266 000201 201 ;THE ERROR NUMBER IS 201  
(1)  
(1) 005270 022767 177777 173142 CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER  
(1) 005276 001402 BEQ .+6 ;BRANCH IF OK  
(1) 005300 104002 HLT+2 ;ANS2 NOT EQUAL TO 177777  
(3) 005302 000202 202 ;THE ERROR NUMBER IS 202  
(1)  
(1) 005304 122767 000026 173072 END26: CMPSB #26, $TESTN ;CHECK THE TEST NUMBER  
(1) 005312 001402 BEQ .+6 ;BRANCH IF OK  
(1) 005314 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.  
(3) 005316 000203 203 ;THE ERROR NUMBER IS 203  
(1)  
(1)
```


7179

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 005460 104400
(1) 005462 004567 007676
(1) 005466 135152 125252
(1) 005472 143325 052525
(1) 005476 000243
(1) 005500 015634 000340
(1) 005504 012700 000510
(1)
(1) 005510 000240
(1) 005512 075010
(1)
(1) 005514 004767 007676
(1) 005520 010067 172710
(1) 005524 122767 000210 172700
(1) 005532 001402
(1) 005534 104000
(3) 005536 000211
(1)
(1) 005540 022767 000514 172666
(1) 005546 001402
(1) 005550 104000
(3) 005552 000212
(1)
(1) 005554 022767 143325 172654
(1) 005562 001402
(1) 005564 104002
(3) 005566 000213
(1)
(1) 005570 022767 052525 172642
(1) 005576 001402
(1) 005600 104002
(3) 005602 000214
(1)
(1) 005604 122767 000030 172572 END30:
(1) 005612 001402
(1) 005614 104000
(3) 005616 000215
(1)
(1)

```

```

*****
TEST 30: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
143325,052525 - 135152,125252 = 143325,052525
PS = 210, STACK POINTER = R0
*****

TST30: 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, SSP ; SAVE "STACK POINTER"
CMPB #210, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 210
211 ; THE ERROR NUMBER IS 211

CMP #STACK4, SSP ; CHECK THE STACK POINTER (R0)
BEQ .+6 ; BRANCH IF OK
HLT ; STACK POINTER (R0) NOT EQUAL TO #STACK4
212 ; THE ERROR NUMBER IS 212

CMP #143325, ANS1 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 143325
213 ; THE ERROR NUMBER IS 213

CMP #052525, ANS2 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 052525
214 ; THE ERROR NUMBER IS 214

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

```

7180

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 005620 104400
(1) 005622 004567 007536
(1) 005626 143325 052525
(1) 005632 135152 125252
(1) 005636 000357
(1) 005640 015634 000340
(1) 005644 012705 000510
(1)
(1) 005650 000240
(1) 005652 075015
(1)
(1) 005654 004767 007536
(1) 005660 010567 172550
(1) 005664 122767 000200 172540
(1) 005672 001402
(1) 005674 104000
(3) 005676 000216
(1)
(1) 005700 022767 000514 172526
(1) 005706 001402
(1) 005710 104000
(3) 005712 000217
(1)
(1) 005714 022767 043325 172514
(1) 005722 001402
(1) 005724 104002
(3) 005726 000220
(1)
(1) 005730 022767 052525 172502
(1) 005736 001402
(1) 005740 104002
(3) 005742 000221
(1)
(1) 005744 122767 000031 172432 END31:
(1) 005752 001402
(1) 005754 104000
(3) 005756 000222
(1)
(1)

```

```

*****
TEST 31: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
          135152,125252 - 143325,052525 = 043325,052525
          PS = 200,          STACK POINTER = R5
*****

TST31:  SCOPE
        JSR      R5,      PUSH4      ; 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      .+6              ; PS NOT EQUAL TO 200
        216                      ; THE ERROR NUMBER IS 216

        CMP      #STACK4,$SP      ; CHECK THE STACK POINTER (R5)
        BEQ      .+6              ; BRANCH IF OK
        HLT      .+6              ; STACK POINTER (R5) NOT EQUAL TO #STACK4
        217                      ; THE ERROR NUMBER IS 217

        CMP      #043325,ANS1     ; CHECK FIRST HALF OF ANSWER
        BEQ      .+6              ; BRANCH IF OK
        HLT+2    .+6              ; ANS1 NOT EQUAL TO 043325
        220                      ; THE ERROR NUMBER IS 220

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

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

```

7181

[illegible]

```

*****
TEST 32:      FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
              043125,052525 - 035152,125252 = 043125,052524
              PS = 000,          STACK POINTER = R2
*****

TST32:  SCOPE
        JSR      R5,      PUSHR      ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
        .WORD    035152,125252      ;SECOND OPERAND ON TOP
        .WORD    043125,052525      ;FIRST OPERAND ON BOTTOM
        .WORD    040                ;PROCESSOR PRIORITY LEVEL
        .WORD    TRAPER, 340        ;FIS TRAP VECTOR
        MOV      #STACK0,R2        ;SET UP STACK POINTER

        NOP
        FSUB     R2                ;FLOATING SUBTRACT ON THE R2 STACK

        JSR      PC,      POPR      ;POP THE ANSWER
        MOV      R2,      SSP      ;SAVE "STACK POINTER"
        TSTB     $PSW            ;CHECK PS (EXCEPT T BIT)
        BEQ      .+6             ;BRANCH IF OK
        HLT      .+6             ;PS NOT EQUAL TO 000
        223                    ;THE ERROR NUMBER IS 223

        CMP      #STACK4,$SSP     ;CHECK THE STACK POINTER (R2)
        BEQ      .+6             ;BRANCH IF OK
        HLT      .+6             ;STACK POINTER (R2) NOT EQUAL TO #STACK4
        224                    ;THE ERROR NUMBER IS 224

        CMP      #043125,ANS1     ;CHECK FIRST HALF OF ANSWER
        BEQ      .+6             ;BRANCH IF OK
        HLT+2    .+6             ;ANS1 NOT EQUAL TO 043125
        225                    ;THE ERROR NUMBER IS 225

        CMP      #052524,ANS2     ;CHECK SECOND HALF OF ANSWER
        BEQ      .+6             ;BRANCH IF OK
        HLT+2    .+6             ;ANS2 NOT EQUAL TO 052524
        226                    ;THE ERROR NUMBER IS 226

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

```

M03

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-36
 DVKACA.SRC TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW

SEQ 0038

7185

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 006116 104400
(1) 006120 004567 007066
(1) 006124 000252 125253
(1) 006130 000425 052525
(1) 006134 000257
(1) 006136 006164 000340
(1)
(1) 006142 000240
(1) 006144 075016
(1)
(1) 006146 004767 007100
(1) 006152 104002
(3) 006154 000230
(1) 006156 012706 000600
(1) 006162 000464
(1)
(1) 006164 004767 007114
(1) 006170 022706 000600
(1) 006174 001405
(1) 006176 012706 000600
(1) 006202 104000
(3) 006204 000231
(1) 006206 000452
(1)
(1) 006210 122767 000340 172214
(1) 006216 001402
(1) 006220 104000
(3) 006222 000232
(1)
(1) 006224 022767 006146 172204
(1) 006232 001402
(1) 006234 104001
(3) 006236 000233
(1)
(1) 006240 022767 000212 172172
(1) 006246 001402
(1) 006250 104002
(3) 006252 000234
(1)
(1) 006254 022767 000252 172160
(1) 006262 001402
(1) 006264 104004
(3) 006266 000235
(1)
(1) 006270 022767 125253 172146
(1) 006276 001402
(1) 006300 104004
(3) 006302 000236
(1)

```

```

*****
TEST 33: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
          000425,052525 - 000252,125253 ==> UNDERFLOW
          PS(ON STACK) = 212, STACK POINTER = SP
*****
TST33:  SCOPE
        JSR    RS,    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  ISR33, 340   ;FIS TRAP VECTOR

        NOP
        FSUB   SP          ;FLOATING SUBTRACT ON THE STACK

RTA33:  JSR    PC,    POPS   ;POP THE "ANSWER"
        HLT+2   ;FIS TRAP DIDN'T OCCURE!
        230     ;THE ERROR NUMBER IS 230
        MOV     #BEGIN, SP ;RESTORE THE STACK POINTER
        BR      END33

ISR33:  JSR    PC,    POPES  ;POP ALL DATA OFF THE STACK
        CMP     #BEGIN, SP ;CHECK THE STACK POINTER
        BEQ     ISA33      ;BRANCH IF OK
        MOV     #BEGIN, SP ;RESTORE THE STACK POINTER
        HLT     ;STACK POINTER FOULED UP
        231     ;THE ERROR NUMBER IS 231
        BR      END33      ;SKIP REST OF TEST

ISA33:  CMPB    #340, SPSW   ;CHECK PS AFTER FIS TRAP
        BEQ     .+6         ;BRANCH IF OK
        HLT     ;PS AFTER FIS TRAP NOT EQUAL TO 340
        232     ;THE ERROR NUMBER IS 232

        CMP     #RTA33, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
        BEQ     .+6         ;BRANCH IF OK
        HLT+1   ;FIS TRAP AT WRONG ADDRESS
        233     ;THE ERROR NUMBER IS 233

        CMP     #212, ANS2   ;CHECK PS BEFORE FIS TRAP
        BEQ     .+6         ;BRANCH IF OK
        HLT+2   ;PS AT FIS TRAP TIME NOT 212
        234     ;THE ERROR NUMBER IS 234

        CMP     #000252,ANS3 ;CHECK DATA FROM THE STACK
        BEQ     .+6         ;BRANCH IF OK
        HLT+4   ;DATA ON STACK (000252) CHANGED
        235     ;THE ERROR NUMBER IS 235

        CMP     #125253,ANS4 ;CHECK DATA FROM STACK
        BEQ     .+6         ;BRANCH IF OK
        HLT+4   ;DATA ON STACK (125253) CHANGED
        236     ;THE ERROR NUMBER IS 236

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-37
 DVKACA.SRC TEST FLOATING SUB. INSTRUCTION WITH UNDERFLOW

SEQ 0039

(1)	006304	022767	000425	172134	CMP	#000425,ANS5	;CHECK DATA FROM STACK
(1)	006312	001402			BEQ	.+6	;BRANCH IF OK
(1)	006314	104006			HLT+6		;DATA ON STACK (000425) CHANGED
(3)	006316	000237			237		;THE ERROR NUMBER IS 237
(1)							
(1)	006320	022767	052525	172122	CMP	#052525,ANS6	;CHECK DATA FROM STACK
(1)	006326	001402			BEQ	.+6	;BRANCH IF OK
(1)	006330	104006			HLT+6		;DATA ON STACK (052525) CHANGED
(3)	006332	000240			240		;THE ERROR NUMBER IS 240
(1)							
(1)	006334	122767	000033	172042	END33:	CMPB	#33, \$TESTN
(1)	006342	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	006344	104000			HLT		;BRANCH IF OK
(3)	006346	000241			241		;WRONG TEST! PC MUST HAVE FOJLED UP.
(1)							;THE ERROR NUMBER IS 241
(1)							

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-38
DVKACA.SRC TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW

7189

(1)					:*****
(1)					:TEST 34: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
(1)					:077652,125253 - 177452,125252 ==> OVERFLOW
(1)					:PS(ON STACK) = 002, STACK POINTER = R3
(1)					:*****
(1)					
(1)	006350	104400			SCOPE
(1)	006352	004567	007006	TST34:	JSR RS, PUSHB ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
(1)	006356	177452	125252	.WORD	177452,125252 ;SECOND OPERAND ON TOP
(1)	006362	077652	125253	.WORD	077652,125253 ;FIRST OPERAND ON BOTTOM
(1)	006366	000015		.WORD	015 ;PROCESSOR PRIORITY LEVEL
(1)	006370	006422	000344	.WORD	ISR34, 344 ;FIS TRAP VECTOR
(1)	006374	012703	000510	MOV	#STACK0,R3 ;SET UP R3 AS STACK POINTER
(1)					
(1)	006400	000240		NOP	
(1)	006402	075013		FSUB	R3 ;FLOATING SUBTRACT ON THE R3 STACK
(1)					
(1)	006404	004767	007006	RTA34:	JSR PC, POPR ;POP THE "ANSWER"
(1)	006410	010367	172020	MOV	R3, SSP ;SAVE STACK POINTER (R3)
(1)	006414	104002		HLT+2	;FIS TRAP DIDN'T OCCURE!
(3)	006416	000242		242	;THE ERROR NUMBER IS 242
(1)	006420	000464		BR	END34
(1)					
(1)	006422	004767	007020	ISR34:	JSR PC, POPER ;POP ALL DATA OFF THE STACKS
(1)	006426	010367	172002	MOV	R3, SSP ;SAVE STACK POINTER (R3)
(1)	006432	122767	000344	CMPB	#344, SPSW ;CHECK PS AFTER FIS TRAP
(1)	006440	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006442	104000		HLT	;PS AFTER FIS TRAP NOT EQUAL TO 344
(3)	006444	000243		243	;THE ERROR NUMBER IS 243
(1)					
(1)	006446	022767	000510	CMP	#STACK0,SSP ;CHECK THE STACK POINTER (R3)
(1)	006454	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006456	104000		HLT	;STACK POINTER (R3) NOT EQUAL TO #STACK0
(3)	006460	000244		244	;THE ERROR NUMBER IS 244
(1)					
(1)	006462	022767	006404	CMP	#RTA34, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1)	006470	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006472	104001		HLT+1	;FIS TRAP AT WRONG ADDRESS
(3)	006474	000245		245	;THE ERROR NUMBER IS 245
(1)					
(1)	006476	022767	000002	CMP	#002, ANS2 ;CHECK PS BEFORE FIS TRAP
(1)	006504	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006506	104002		HLT+2	;PS AT FIS TRAP TIME NOT 002
(3)	006510	000246		246	;THE ERROR NUMBER IS 246
(1)					
(1)	006512	022767	177452	CMP	#177452,ANS3 ;CHECK DATA FROM THE STACK
(1)	006520	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006522	104004		HLT+4	;DATA ON STACK (177452) CHANGED
(3)	006524	000247		247	;THE ERROR NUMBER IS 247
(1)					
(1)	006526	022767	125252	CMP	#125252,ANS4 ;CHECK DATA FROM STACK
(1)	006534	001402		BEQ	.+6 ;BRANCH IF OK
(1)	006536	104004		HLT+4	;DATA ON STACK (125252) CHANGED
(3)	006540	000250		250	;THE ERROR NUMBER IS 250
(1)					

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-39
 DVKACA.SRC TEST FLOATING SUB. INSTRUCTION WITH OVERFLOW

SEQ 0041

(1)	006542	022767	077652	171676	CMP	#077652,ANS5	;CHECK DATA FROM STACK
(1)	006550	001402			BEQ	.+6	;BRANCH IF OK
(1)	006552	104006			HLT+6		;DATA ON STACK (077652) CHANGED
(3)	006554	000251			251		;THE ERROR NUMBER IS 251
(1)							
(1)	006556	022767	125253	171664	CMP	#125253,ANS6	;CHECK DATA FROM STACK
(1)	006564	001402			BEQ	.+6	;BRANCH IF OK
(1)	006566	104006			HLT+6		;DATA ON STACK (125253) CHANGED
(3)	006570	000252			252		;THE ERROR NUMBER IS 252
(1)							
(1)	006572	122767	000034	171604	CMPB	#34, \$TESTN	;CHECK THE TEST NUMBER
(1)	006600	001402			BEQ	.+6	;BRANCH IF OK
(1)	006602	104000			HLT		;WRONG TEST! PC MUST HAVE FOULED UP.
(3)	006604	000253			253		;THE ERROR NUMBER IS 253
(1)							
(1)							

7193

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 006606 104400
(1) 006610 004567 006550
(1) 006614 000000 000000
(1) 006620 000000 000000
(1) 006624 000111
(1) 006626 015634 000340
(1) 006632 012704 000510
(1)
(1) 006636 000240
(1) 006640 075024
(1)
(1) 006642 004767 006550
(1) 006646 010467 171562
(1) 006652 122767 000004 171552
(1) 006660 001402
(1) 006662 104000
(3) 006664 000254
(1)
(1) 006666 022767 000514 171540
(1) 006674 001402
(1) 006676 104000
(3) 006700 000255
(1)
(1) 006702 005767 171530
(1) 006706 001402
(1) 006710 104002
(3) 006712 000256
(1)
(1) 006714 005767 171520
(1) 006720 001402
(1) 006722 104002
(3) 006724 000257
(1)
(1) 006726 122767 000035 171450
(1) 006734 001402
(1) 006736 104000
(3) 006740 000260
(1)
(1)

```

TEST 35: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
000000,000000 * 000000,000000 = 000000,000000
PS = 004, STACK POINTER = R4

TST35: SCOPE
JSR RS, PUSHR ; PUSH 4 WORDS ONTO R4 STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; FIRST OPERAND ON BOTTOM
; PROCESSOR PRIORITY LEVEL
; FIS TRAP VECTOR
; SET UP STACK POINTER
; FLOATING MULTIPLY ON THE R4 STACK
MOV #004, \$SP
MOV #004, \$PSW
CMPB #004, \$PSW
BEQ .+6
HLT 254
; POP THE ANSWER
; SAVE "STACK POINTER"
; CHECK PS (EXCEPT T BIT)
; BRANCH IF OK
; PS NOT EQUAL TO 004
; THE ERROR NUMBER IS 254
CMP #STACK4, \$SP
BEQ .+6
HLT 255
; CHECK THE STACK POINTER (R4)
; BRANCH IF OK
; STACK POINTER (R4) NOT EQUAL TO #STACK4
; THE ERROR NUMBER IS 255
TST ANS1
BEQ .+6
HLT+2 256
; CHECK FIRST HALF OF ANSWER
; BRANCH IF OK
; ANS1 NOT EQUAL TO 000000
; THE ERROR NUMBER IS 256
TST ANS2
BEQ .+6
HLT+2 257
; CHECK SECOND HALF OF ANSWER
; BRANCH IF OK
; ANS2 NOT EQUAL TO 000000
; THE ERROR NUMBER IS 257
CMPB #35, \$TESTN
BEQ .+6
HLT 260
; CHECK THE TEST NUMBER
; BRANCH IF OK
; WRONG TEST! PC MUST HAVE FOULED UP.
; THE ERROR NUMBER IS 260
END35:

7194

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 006742 104400
(1) 006744 004567 006414
(1) 006750 052345 123456
(1) 006754 140200 000000
(1) 006760 000343
(1) 006762 015634 000340
(1) 006766 012702 000510
(1)
(1) 006772 000240
(1) 006774 075022
(1)
(1) 006776 004767 006414
(1) 007002 010267 171426
(1) 007006 122767 000210 171416
(1) 007014 001402
(1) 007016 104000
(3) 007020 000261
(1)
(1) 007022 022767 000514 171404
(1) 007030 001402
(1) 007032 104000
(3) 007034 000262
(1)
(1) 007036 022767 152345 171372
(1) 007044 001402
(1) 007046 104002
(3) 007050 000263
(1)
(1) 007052 022767 123456 171360
(1) 007060 001402
(1) 007062 104002
(3) 007064 000264
(1)
(1) 007066 122767 000036 171310 END36:
(1) 007074 001402
(1) 007076 104000
(3) 007100 000265
(1)
(1)

```

TEST 36: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
140200,000000 * 052345,123456 = 152345,123456
PS = 210, STACK POINTER = R2

TST36: SCOPE
JSR R5, PUSHR ; PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; FIRST OPERAND ON BOTTOM
; PROCESSOR PRIORITY LEVEL
; FIS TRAP VECTOR
; SET UP STACK POINTER
; FLOATING MULTIPLY ON THE R2 STACK
; POP THE ANSWER
; SAVE "STACK POINTER"
; CHECK PS (EXCEPT T BIT)
; BRANCH IF OK
; PS NOT EQUAL TO 210
; THE ERROR NUMBER IS 261
; CHECK THE STACK POINTER (R2)
; BRANCH IF OK
; STACK POINTER (R2) NOT EQUAL TO #STACK4
; THE ERROR NUMBER IS 262
; CHECK FIRST HALF OF ANSWER
; BRANCH IF OK
; ANS1 NOT EQUAL TO 152345
; THE ERROR NUMBER IS 263
; CHECK SECOND HALF OF ANSWER
; BRANCH IF OK
; ANS2 NOT EQUAL TO 123456
; THE ERROR NUMBER IS 264
; CHECK THE TEST NUMBER
; BRANCH IF OK
; WRONG TEST! PC MUST HAVE FOULED UP.
; THE ERROR NUMBER IS 265

7195

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 007102 104400
(1) 007104 004567 006254
(1) 007110 135753 024642
(1) 007114 100125 052525
(1) 007120 000117
(1) 007122 015634 000340
(1) 007126 012705 000510
(1)
(1) 007132 000240
(1) 007134 075025
(1)
(1) 007136 004767 006254
(1) 007142 010567 171266
(1) 007146 122767 000004 171256
(1) 007154 001402
(1) 007156 104000
(3) 007160 000266
(1)
(1) 007162 022767 000514 171244
(1) 007170 001402
(1) 007172 104000
(3) 007174 000267
(1)
(1) 007176 005767 171234
(1) 007202 001402
(1) 007204 104002
(3) 007206 000270
(1)
(1) 007210 005767 171224
(1) 007214 001402
(1) 007216 104002
(3) 007220 000271
(1)
(1) 007222 122767 000037 171154 END37:
(1) 007230 001402
(1) 007232 104000
(3) 007234 000272
(1)
(1)

```

```

*****
TEST 37: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
          100125,052525 * 135753,024642 = 000000,000000
          PS = 004, STACK POINTER = RS
*****
TST37: SCOPE
        JSR RS, PUSHR ; PUSH 4 WORDS ONTO RS 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,RS ; SET UP STACK POINTER

        NOP
        FMUL RS ; FLOATING MULTIPLY ON THE RS STACK

        JSR PC, POPR ; POP THE ANSWER
        MOV RS, SSP ; SAVE "STACK POINTER"
        CMPB #004, SPSW ; CHECK PS (EXCEPT T BIT)
        BEQ .+6 ; BRANCH IF OK
        HLT ; PS NOT EQUAL TO 004
        266 ; THE ERROR NUMBER IS 266

        CMP #STACK4,SSP ; CHECK THE STACK POINTER (RS)
        BEQ .+6 ; BRANCH IF OK
        HLT ; STACK POINTER (RS) NOT EQUAL TO #STACK4
        267 ; THE ERROR NUMBER IS 267

        TST ANS1 ; CHECK FIRST HALF OF ANSWER
        BEQ .+6 ; BRANCH IF OK
        HLT+2 ; ANS1 NOT EQUAL TO 000000
        270 ; THE ERROR NUMBER IS 270

        TST ANS2 ; CHECK SECOND HALF OF ANSWER
        BEQ .+6 ; BRANCH IF OK
        HLT+2 ; ANS2 NOT EQUAL TO 000000
        271 ; THE ERROR NUMBER IS 271

        CMPB #37, $TESTN ; CHECK THE TEST NUMBER
        BEQ .+6 ; BRANCH IF OK
        HLT ; WRONG TEST! PC MUST HAVE FOULED UP.
        272 ; THE ERROR NUMBER IS 272

```

7196

(1)						:*****
(1)						:TEST 40: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
(1)						:161616,161616 * 000052,125252 = 000000,000000
(1)						:PS = 204, STACK POINTER = R3
(1)						:*****
(1)						
(1)	007236	104400				TST40: SCOPE
(1)	007240	004567	006120			JSR RS, PUSHR ;PUSH 4 WORDS ONTO R3 STACK, SET PRIORITY
(1)	007244	000052	125252			.WORD 000052,125252 ;SECOND OPERAND ON TOP
(1)	007250	161616	161616			.WORD 161616,161616 ;FIRST OPERAND ON BOTOM
(1)	007254	000217				.WORD 217 ;PROCESSOR PRIORITY LEVEL
(1)	007256	015634	000340			.WORD TRAPER, 340 ;FIS TRAP VECTOR
(1)	007262	012703	000510			MOV #STACK0,R3 ;SET UP STACK POINTER
(1)						
(1)	007266	000240				NOP
(1)	007270	075023				FMUL R3 ;FLOATING MULTIPLY ON THE R3 STACK
(1)						
(1)	007272	004767	006120			JSR PC, POPR ;POP THE ANSWER
(1)	007276	010367	171132			MOV R3, \$SP ;SAVE "STACK POINTER"
(1)	007302	122767	000204	171122		CMPB #204, \$PSW ;CHECK PS (EXCEPT T BIT)
(1)	007310	001402				BEQ .+6 ;BRANCH IF OK
(1)	007312	104000				HLT ;PS NOT EQUAL TO 204
(3)	007314	000273				273 ;THE ERROR NUMBER IS 273
(1)						
(1)	007316	022767	000514	171110		CMP #STACK4,\$SP ;CHECK THE STACK POINTER (R3)
(1)	007324	001402				BEQ .+6 ;BRANCH IF OK
(1)	007326	104000				HLT ;STACK POINTER (R3) NOT EQUAL TO #STACK4
(3)	007330	000274				274 ;THE ERROR NUMBER IS 274
(1)						
(1)	007332	005767	171100			TST ANS1 ;CHECK FIRST HALF OF ANSWER
(1)	007336	001402				BEQ .+6 ;BRANCH IF OK
(1)	007340	104002				HLT+2 ;ANS1 NOT EQUAL TO 000000
(3)	007342	000275				275 ;THE ERROR NUMBER IS 275
(1)						
(1)	007344	005767	171070			TST ANS2 ;CHECK SECOND HALF OF ANSWER
(1)	007350	001402				BEQ .+6 ;BRANCH IF OK
(1)	007352	104002				HLT+2 ;ANS2 NOT EQUAL TO 000000
(3)	007354	000276				276 ;THE ERPOR NUMBER IS 276
(1)						
(1)	007356	122767	000040	171020	END40:	CMPB #40, \$TESTN ;CHECK THE TEST NUMBER
(1)	007364	001402				BEQ .+6 ;BRANCH IF OK
(1)	007366	104000				HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3)	007370	000277				277 ;THE ERROR NUMBER IS 277
(1)						
(1)						

7197

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 007372 104400
(1) 007374 004567 005612
(1) 007400 041500 000000
(1) 007404 176452 125252
(1) 007410 000357
(1) 007412 015634 000340
(1)
(1) 007416 000240
(1) 007420 075026
(1)
(1) 007422 004767 005624
(1) 007426 022706 000600
(1) 007432 001405
(1) 007434 012706 000600
(1) 007440 104000
(3) 007442 000300
(1) 007444 000422
(1)
(1) 007446 122767 000210 170756 TSA41: CMPB #210, SPSW ;CHECK PS (EXCEPT T BIT)
(1) 007454 001402 BEQ .+6 ;BRANCH IF OK
(1) 007456 104000 HLT ;PS NOT EQUAL TO 210
(3) 007460 000301 301 ;THE ERROR NUMBER IS 301
(1)
(1) 007462 022767 177777 170746 CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
(1) 007470 001402 BEQ .+6 ;BRANCH IF OK
(1) 007472 104002 HLT+2 ;ANS1 NOT EQUAL TO 177777
(3) 007474 000302 302 ;THE ERROR NUMBER IS 302
(1)
(1) 007476 022767 177777 170734 CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
(1) 007504 001402 BEQ .+6 ;BRANCH IF OK
(1) 007506 104002 HLT+2 ;ANS2 NOT EQUAL TO 177777
(3) 007510 000303 303 ;THE ERROR NUMBER IS 303
(1)
(1) 007512 122767 000041 170664 END41: CMPB #41, $TESTN ;CHECK THE TEST NUMBER
(1) 007520 001402 BEQ .+6 ;BRANCH IF OK
(1) 007522 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 007524 000304 304 ;THE ERROR NUMBER IS 304
(1)
(1)

```

```

*****
TEST 41: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
176452,125252 * 041500,000000 = 177777,177777
PS = 210, STACK POINTER = SP
*****

```

```

TST41: SCOPE
JSR RS, 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, POPS ;POP THE ANSWER
CMP #BEGIN, SP ;CHECK THE STACK POINTER
BEQ TSA41 ;BRANCH IF OK
MOV #BEGIN, SP ;RESTORE STACK POINTER
HLT ;STACK POINTER FOULED UP
300 ;THE ERROR NUMBER IS 300
BR END41 ;SKIP REST OF TEST

```

```

CMPB #210, SPSW ;CHECK PS (EXCEPT T BIT)
BEQ .+6 ;BRANCH IF OK
HLT ;PS NOT EQUAL TO 210
301 ;THE ERROR NUMBER IS 301

```

```

CMP #177777,ANS1 ;CHECK FIRST HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS1 NOT EQUAL TO 177777
302 ;THE ERROR NUMBER IS 302

```

```

CMP #177777,ANS2 ;CHECK SECOND HALF OF ANSWER
BEQ .+6 ;BRANCH IF OK
HLT+2 ;ANS2 NOT EQUAL TO 177777
303 ;THE ERROR NUMBER IS 303

```

```

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

```

7198

```

(1)
(1)
(1)
(1)
(1)
(1)
(1) 007526 104400
(1) 007530 004567 005630
(1) 007534 114100 000001
(1) 007540 124252 125252
(1) 007544 000200
(1) 007546 015634 000340
(1) 007552 012701 000510
(1)
(1) 007556 000240
(1) 007560 075021
(1)
(1) 007562 004767 005630
(1) 007566 010167 170642
(1) 007572 122767 000200 170632
(1) 007600 001402
(1) 007602 104000
(3) 007604 000305
(1)
(1) 007606 022767 000514 170620
(1) 007614 001402
(1) 007616 104000
(3) 007620 000306
(1)
(1) 007622 022767 000200 170606
(1) 007630 001402
(1) 007632 104002
(3) 007634 000307
(1)
(1) 007636 005767 170576
(1) 007642 001402
(1) 007644 104002
(3) 007646 000310
(1)
(1) 007650 122767 000042 170526 END42:
(1) 007656 001402
(1) 007660 104000
(3) 007662 000311
(1)
(1)

```

```

*****
TEST 42: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
124252,125252 * 114100,000001 = 000200,000000
PS = 200, STACK POINTER = R1
*****

TST42: SCOPE
JSR RS, PUSH4 ; PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
        .WORD 114100,000001 ; SECOND OPERAND ON TOP
        .WORD 124252,125252 ; FIRST OPERAND ON BOTOM
        .WORD 200 ; PROCESSOR PRIORITY LEVEL
        .WORD TRAPER, 340 ; FIS TRAP VECTOR
MOV #STACK0,R1 ; SET UP STACK POINTER

NOP
FMUL R1 ; FLOATING MULTIPLY ON THE R1 STACK

JSR PC, POPR ; POP THE ANSWER
MOV R1, SSP ; SAVE "STACK POINTER"
CMPB #200, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 200
305 ; THE ERROR NUMBER IS 305

CMP #STACK4,SSP ; CHECK THE STACK POINTER (R1)
BEQ .+6 ; BRANCH IF OK
HLT ; STACK POINTER (R1) NOT EQUAL TO #STACK4
306 ; THE ERROR NUMBER IS 306

CMP #000200,ANS1 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 000200
307 ; THE ERROR NUMBER IS 307

TST ANS2 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 000000
310 ; THE ERROR NUMBER IS 310

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

```

7200

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 007664 104400
(1) 007666 004567 005644
(1) 007672 007716
(1) 007674 104000 104000
(1) 007700 104000 105004
(1) 007704 000252
(1) 007706 015634 000340
(1)
(1) 007712 000240
(1) 007714 075017
(1) 007716 104000
(1) 007720 104000
(1) 007722 104000
(1) 007724 105004
(1)
(1) 007726 004767 005634
(1) 007732 122767 000210 170472
(1) 007740 001402
(1) 007742 104000
(3) 007744 000312
(1)
(1) 007746 022767 104000 170462
(1) 007754 001402
(1) 007756 104002
(3) 007760 000313
(1)
(1) 007762 022767 104000 170450
(1) 007770 001402
(1) 007772 104002
(3) 007774 000314
(1)
(1) 007776 022767 100401 170436
(1) 010004 001402
(1) 010006 104004
(3) 010010 000315
(1)
(1) 010012 005767 170426
(1) 010016 001402
(1) 010020 104004
(3) 010022 000316
(1)
(1) 010024 122767 000043 170352
(1) 010032 001402
(1) 010034 104000
(3) 010036 000317
(1)

```

TEST 43: FSUB (LSI-11 FLOATING SUBTRACT INSTRUCTION)
104000,105004 - 104000,104000 = 100401,000000
PS = 210, STACK POINTER = PC

TST43: JSR RS, 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
WORD STK43
WORD 104000,104000
WORD 104000,105004
WORD 252
WORD TRAPER,340

STK43: 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, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 210
312 ; THE ERROR NUMBER IS 312

CMP #104000,ANS1 ; CHECK FIRST HALF OF INPUT DATA (STK43)
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 104000
313 ; THE ERROR NUMBER IS 313

CMP #104000,ANS2 ; CHECK SECOND HALF OF INPUT DATA (STK43+2)
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 104000
314 ; THE ERROR NUMBER IS 314

CMP #100401,ANS3 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+4 ; ANS3 NOT EQUAL TO 100401
315 ; THE ERROR NUMBER IS 315

TST ANS4 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+4 ; ANS4 NOT EQUAL TO 000000
316 ; THE ERROR NUMBER IS 316

CMPB #43, \$TESTN ; CHECK THE TEST NUMBER
BEQ .+6 ; BRANCH IF OK
HLT ; WRONG TEST! PC MUST HAVE FOULED UP.
317 ; THE ERROR NUMBER IS 317

7202

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 010040 104400
(1) 010042 004567 005470
(1) 010046 010072
(1) 010050 104000 104000
(1) 010054 134600 073601
(1) 010060 000246
(1) 010062 015634 000340
(1)
(1) 010066 000240
(1) 010070 075027
(1) 010072 104000
(1) 010074 104000
(1) 010076 134600
(1) 010100 073601
(1)
(1) 010102 004767 005460
(1) 010106 122767 000200 170316
(1) 010114 001402
(1) 010116 104000
(3) 010120 000320
(1)
(1) 010122 022767 104000 170306
(1) 010130 001402
(1) 010132 104002
(3) 010134 000321
(1)
(1) 010136 022767 104000 170274
(1) 010144 001402
(1) 010146 104002
(3) 010150 000322
(1)
(1) 010152 022767 000401 170262
(1) 010160 001402
(1) 010162 104004
(3) 010164 000323
(1)
(1) 010166 005767 170252
(1) 010172 001402
(1) 010174 104004
(3) 010176 000324
(1)
(1) 010200 122767 000044 170176 END44:
(1) 010206 001402
(1) 010210 104000
(3) 010212 000325

```

TEST 44: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
134600,073601 * 104000,104000 = 000401,000000
PS = 200, STACK POINTER = PC

TST44: SCOPE
JSR RS, PUSH7 ; PUSH 4 WORDS ONTO STACK, SET PRIORITY
WORD STK44 ; 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

STK44: 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, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 200
320 ; THE ERROR NUMBER IS 320

CMP #104000,ANS1 ; CHECK FIRST HALF OF INPUT DATA (STK44)
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 104000
321 ; THE ERROR NUMBER IS 321

CMP #104000,ANS2 ; CHECK SECOND HALF OF INPUT DATA (STK44+2)
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 104000
322 ; THE ERROR NUMBER IS 322

CMP #000401,ANS3 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+4 ; ANS3 NOT EQUAL TO 000401
323 ; THE ERROR NUMBER IS 323

TST ANS4 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+4 ; ANS4 NOT EQUAL TO 000000
324 ; THE ERROR NUMBER IS 324

CMPB #44, \$TESTN ; CHECK THE TEST NUMBER
BEQ .+6 ; BRANCH IF OK
HLT ; WRONG TEST! PC MUST HAVE FOULED UP.
325 ; THE ERROR NUMBER IS 325

7206

(1)

;*****

```

(1) ;TEST 45: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
(1) ; 024252,125252 * 114100,000000 ==> UNDERFLOW
(1) ; PS(ON STACK) = 212, STACK POINTER = R0
(1) ;*****
(1)
(1) 010214 104400 SCOPE
(1) 010216 004567 005142 TST45: JSR R5, PUSHR ;PUSH 4 WORDS ONTO R0 STACK, SET PRIORITY
(1) 010222 114100 000000 .WORD 114100,000000 ;SECOND OPERAND ON TOP
(1) 010226 024252 125252 .WORD 024252,125252 ;FIRST OPERAND ON BOTTOM
(1) 010232 000305 .WORD 305 ;PROCESSOR PRIORITY LEVEL
(1) 010234 010266 000057 .WORD ISR45, 057 ;FIS TRAP VECTOR
(1) 010240 012700 000510 MOV #STACK0,R0 ;SET UP R0 AS STACK POINTER
(1)
(1) 010244 000240 NOP
(1) 010246 075020 FMUL R0 ;FLOATING MULTIPLY ON THE R0 STACK
(1)
(1) 010250 004767 005142 RTA45: JSR PC, POPR ;POP THE "ANSWER"
(1) 010254 010067 170154 MOV R0, $SP ;SAVE STACK POINTER (R0)
(1) 010260 104002 HLT+2 ;FIS TRAP DIDN'T OCCURE!
(3) 010262 000326 326 ;THE ERROR NUMBER IS 326
(1) 010264 000463 BR END45
(1)
(1) 010266 004767 005154 ISR45: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
(1) 010272 010067 170136 MOV R0, $SP ;SAVE STACK POINTER (R0)
(1) 010276 122767 000057 170126 CMPB #057, $PSW ;CHECK PS AFTER FIS TRAP
(1) 010304 001402 BEQ .+6 ;BRANCH IF OK
(1) 010306 104000 HLT ;PS AFTER FIS TRAP NOT EQUAL TO 057
(3) 010310 000327 327 ;THE ERROR NUMBER IS 327
(1)
(1) 010312 022767 000510 170114 CMP #STACK0,$SP ;CHECK THE STACK POINTER (R0)
(1) 010320 001402 BEQ .+6 ;BRANCH IF OK
(1) 010322 104000 HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK0
(3) 010324 000330 330 ;THE ERROR NUMBER IS 330
(1)
(1) 010326 022767 010250 170102 CMP #RTA45, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 010334 001402 BEQ .+6 ;BRANCH IF OK
(1) 010336 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 010340 000331 331 ;THE ERROR NUMBER IS 331
(1)
(1) 010342 022767 000212 170070 CMP #212, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 010350 001402 BEQ .+6 ;BRANCH IF OK
(1) 010352 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 212
(3) 010354 000332 332 ;THE ERROR NUMBER IS 332
(1)
(1) 010356 022767 114100 170056 CMP #114100,ANS3 ;CHECK DATA FROM THE STACK
(1) 010364 001402 BEQ .+6 ;BRANCH IF OK
(1) 010366 104004 HLT+4 ;DATA ON STACK (114100) CHANGED
(3) 010370 000333 333 ;THE ERROR NUMBER IS 333
(1)
(1) 010372 005767 170046 TST ANS4 ;CHECK DATA FROM STACK
(1) 010376 001402 BEQ .+6 ;BRANCH IF OK
(1) 010400 104004 HLT+4 ;DATA ON STACK (000000) CHANGED
(3) 010402 000334 334 ;THE ERROR NUMBER IS 334
(1)
(1) 010404 022767 024252 170034 CMP #024252,ANS5 ;CHECK DATA FROM STACK
(1) 010412 001402 BEQ .+6 ;BRANCH IF OK

```

```
(1) 010414 104006 HLT+6 ;DATA ON STACK (024252) CHANGED
(3) 010416 000335 335 ;THE ERROR NUMBER IS 335
(1)
(1) 010420 022767 125252 170022 CMP #125252,ANS6 ;CHECK DATA FROM STACK
(1) 010426 001402 BEQ .+6 ;BRANCH IF OK
(1) 010430 104006 HLT+6 ;DATA ON STACK (125252) CHANGED
(3) 010432 000336 336 ;THE ERROR NUMBER IS 336
(1)
(1) 010434 122767 000045 167742 END45: CMPB #45, $TESTN ;CHECK THE TEST NUMBER
(1) 010442 001402 BEQ .+6 ;BRANCH IF OK
(1) 010444 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 010446 000337 337 ;THE ERROR NUMBER IS 337
(1)
(1)
```

7210

```

(1) ;*****
(1) ;TEST 46: FMUL (LSI-11 FLOATING MULTIPLY INSTRUCTION)
(1) ; 076452,125252 * 041500,000001 ==> OVERFLOW
(1) ; PS(ON STACK) = 002, STACK POINTER = SP
(1) ;*****
(1)
(1) 010450 104400
(1) 010452 004567 004534 TST46: SCOPE
(1) 010456 041500 000001 JSR R5 PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
(1) 010462 076452 125252 .WORD 041500,000001 ;SECOND OPERAND ON TOP
(1) 010466 000105 .WORD 076452,125252 ;FIRST OPERAND ON BOTTOM
(1) 010470 010516 000357 .WORD 105 ;PROCESSOR PRIORITY LEVEL
(1) .WORD ISR46, 357 ;FIS TRAP VECTOR
(1)
(1) 010474 000240 NOP
(1) 010476 075026 FMUL SP ;FLOATING MULTIPLY ON THE STACK
(1)
(1) 010500 004767 004546 RTA46: JSR PC, POPS ;POP THE "ANSWER"
(1) 010504 104002 HLT+2 ;FIS TRAP DIDN'T OCCURE!
(3) 010506 000340 340 ;THE ERROR NUMBER IS 340
(1) 010510 012706 000600 MOV #BEGIN, SP ;RESTORE THE STACK POINTER
(1) 010514 000464 BR END46
(1)
(1) 010516 004767 004562 ISR46: JSR PC, POPES ;POP ALL DATA OFF THE STACK
(1) 010522 022706 000600 CMP #BEGIN, SP ;CHECK THE STACK POINTER
(1) 010526 001405 BEQ ISA46 ;BRANCH IF OK
(1) 010530 012706 000600 MOV #BEGIN, SP ;RESTORE THE STACK POINTER
(1) 010534 104000 HLT ;STACK POINTER FOULED UP
(3) 010536 000341 341 ;THE ERROR NUMBER IS 341
(1) 010540 000452 BR END46 ;SKIP REST OF TEST
(1)
(1) 010542 122767 000357 167662 ISA46: CMPB #357, SPSW ;CHECK PS AFTER FIS TRAP
(1) 010550 001402 BEQ .+6 ;BRANCH IF OK
(1) 010552 104000 HLT ;PS AFTER FIS TRAP NOT EQUAL TO 357
(3) 010554 000342 342 ;THE ERROR NUMBER IS 342
(1)
(1) 010556 022767 010500 167652 CMP #RTA46, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 010564 001402 BEQ .+6 ;BRANCH IF OK
(1) 010566 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 010570 000343 343 ;THE ERROR NUMBER IS 343
(1)
(1) 010572 022767 000002 167640 CMP #002, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 010600 001402 BEQ .+6 ;BRANCH IF OK
(1) 010602 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 002
(3) 010604 000344 344 ;THE ERROR NUMBER IS 344
(1)
(1) 010606 022767 041500 167626 CMP #041500,ANS3 ;CHECK DATA FROM THE STACK
(1) 010614 001402 BEQ .+6 ;BRANCH IF OK
(1) 010616 104004 HLT+4 ;DATA ON STACK (041500) CHANGED
(3) 010620 000345 345 ;THE ERROR NUMBER IS 345
(1)
(1) 010622 022767 000001 167614 CMP #000001,ANS4 ;CHECK DATA FROM STACK
(1) 010630 001402 BEQ .+6 ;BRANCH IF OK
(1) 010632 104004 HLT+4 ;DATA ON STACK (000001) CHANGED
(3) 010634 000346 346 ;THE ERROR NUMBER IS 346
(1)

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-51
 DVKACA.SRC TEST FLOATING MUL. INSTRUCTION WITH OVERFLOW

SEQ 0053

(1)	010636	022767	076452	167602	CMP	#076452,ANS5	;CHECK DATA FROM STACK
(1)	010644	001402			BEQ	.+6	;BRANCH IF OK
(1)	010646	104006			HLT+6		;DATA ON STACK (076452) CHANGED
(3)	010650	000347			347		;THE ERROR NUMBER IS 347
(1)							
(1)	010652	022767	125252	167570	CMP	#125252,ANS6	;CHECK DATA FROM STACK
(1)	010660	001402			BEQ	.+6	;BRANCH IF OK
(1)	010662	104006			HLT+6		;DATA ON STACK (125252) CHANGED
(3)	010664	000350			350		;THE ERROR NUMBER IS 350
(1)							
(1)	010666	122767	000046	167510	END46:	CMPB	#46, \$TESTN
(1)	010674	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	010676	104000			HLT		;BRANCH IF OK
(3)	010700	000351			351		;WRONG TEST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 351
(1)							

7215

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 011040 104400
(1) 011042 004567 004144
(1) 011046 027652 125253
(1) 011052 167452 125252
(1) 011056 000300
(1) 011060 015634 000340
(1)
(1) 011064 000240
(1) 011066 075036
(1)
(1) 011070 004767 004156
(1) 011074 022706 000600
(1) 011100 001405
(1) 011102 012706 000600
(1) 011106 104000
(3) 011110 000357
(1) 011112 000422
(1)
(1) 011114 122767 000210 167310
(1) 011122 001402
(1) 011124 104000
(3) 011126 000360
(1)
(1) 011130 022767 177777 167300
(1) 011136 001402
(1) 011140 104002
(3) 011142 000361
(1)
(1) 011144 022767 177777 167266
(1) 011152 001402
(1) 011154 104002
(3) 011156 000362
(1)
(1) 011160 122767 000050 167216
(1) 011166 001402
(1) 011170 104000
(3) 011172 000363
(1)
(1)

```

TEST 50: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
167452,125252 / 027652,125253 = 177777,177777
PS = 210, STACK POINTER = SP

TST50: SCOPE
JSR RS, PUSH5 ; PUSH 4 WORDS ONTO STACK, SET PRIORITY
; SECOND OPERAND ON TOP
; FIRST OPERAND ON BOTTOM
; PROCESSOR PRIORITY LEVEL
; 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 TSA50 ; BRANCH IF OK
MOV #BEGIN, SP ; RESTORE STACK POINTER
HLT ; STACK POINTER FOULED UP
357 ; THE ERROR NUMBER IS 357
BR ENDS0 ; SKIP REST OF TEST

TSA50: CMPB #210, SPSW ; CHECK PS (EXCEPT T BIT)
BEQ .+6 ; BRANCH IF OK
HLT ; PS NOT EQUAL TO 210
360 ; THE ERROR NUMBER IS 360

CMP #177777,ANS1 ; CHECK FIRST HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS1 NOT EQUAL TO 177777
361 ; THE ERROR NUMBER IS 361

CMP #177777,ANS2 ; CHECK SECOND HALF OF ANSWER
BEQ .+6 ; BRANCH IF OK
HLT+2 ; ANS2 NOT EQUAL TO 177777
362 ; THE ERROR NUMBER IS 362

ENDS0: CMPB #50, \$TESTN ; CHECK THE TEST NUMBER
BEQ .+6 ; BRANCH IF OK
HLT ; WRONG TEST! PC MUST HAVE FOULED UP.
363 ; THE ERROR NUMBER IS 363

7216

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 011174 104400
(1) 011176 004567 004162
(1) 011202 065252 125252
(1) 011206 125252 125252
(1) 011212 000217
(1) 011214 015634 000340
(1) 011220 012702 000510
(1)
(1) 011224 000240
(1) 011226 075032
(1)
(1) 011230 004767 004162
(1) 011234 010267 167174
(1) 011240 122767 000210 167164
(1) 011246 001402
(1) 011250 104000
(3) 011252 000364
(1)
(1) 011254 022767 000514 167152
(1) 011262 001402
(1) 011264 104000
(3) 011266 000365
(1)
(1) 011270 022767 100200 167140
(1) 011276 001402
(1) 011300 104002
(3) 011302 000366
(1)
(1) 011304 005767 167130
(1) 011310 001402
(1) 011312 104002
(3) 011314 000367
(1)
(1) 011316 122767 000051 167060 END51:
(1) 011324 001402
(1) 011326 104000
(3) 011330 000370
(1)
(1)

```

```

*****
TEST 51:      FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
              125252 125252 / 065252 125252 = 100200,000000
              PS = 210,      STACK POINTER = R2
*****

TST51:  SCOPE
        JSR      RS,      PUSH4      ;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,      SSP      ;SAVE "STACK POINTER"
        CMPB     #210,      SPSW    ;CHECK PS (EXCEPT T BIT)
        BEQ      .+6              ;BRANCH IF OK
        HLT      .                ;PS NOT EQUAL TO 210
        364      ;THE ERROR NUMBER IS 364

        CMP      #STACK4,$SP      ;CHECK THE STACK POINTER (R2)
        BEQ      .+6              ;BRANCH IF OK
        HLT      .                ;STACK POINTER (R2) NOT EQUAL TO #STACK4
        365      ;THE ERROR NUMBER IS 365

        CMP      #100200,ANS1     ;CHECK FIRST HALF OF ANSWER
        BEQ      .+6              ;BRANCH IF OK
        HLT+2    .                ;ANS1 NOT EQUAL TO 100200
        366      ;THE ERROR NUMBER IS 366

        TST      ANS2             ;CHECK SECOND HALF OF ANSWER
        BEQ      .+6              ;BRANCH IF OK
        HLT+2    .                ;ANS2 NOT EQUAL TO 000000
        367      ;THE ERROR NUMBER IS 367

        CMPB     #51,      $TESTN  ;CHECK THE TEST NUMBER
        BEQ      .+6              ;BRANCH IF OK
        HLT      .                ;WRONG TEST! PC MUST HAVE FOULED UP.
        370      ;THE ERROR NUMBER IS 370

```

7217

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 011332 104400
(1) 011334 004567 004024
(1) 011340 140670 123456
(1) 011344 000000 000000
(1) 011350 000105
(1) 011352 015634 000340
(1) 011356 012703 000510
(1)
(1) 011362 000240
(1) 011364 075033
(1)
(1) 011366 004767 004024
(1) 011372 010367 167036
(1) 011376 122767 000004 167026
(1) 011404 001402
(1) 011406 104000
(3) 011410 000371
(1)
(1) 011412 022767 000514 167014
(1) 011420 001402
(1) 011422 104000
(3) 011424 000372
(1)
(1) 011426 005767 167004
(1) 011432 001402
(1) 011434 104002
(3) 011436 000373
(1)
(1) 011440 005767 166774
(1) 011444 001402
(1) 011446 104002
(3) 011450 000374
(1)
(1) 011452 122767 000052 166724
(1) 011460 001402
(1) 011462 104000
(3) 011464 000375
(1)
(1)

```

```

*****
TEST 52: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
          000000,000000 / 140670,123456 = 000000,000000
          PS = 004, STACK POINTER = R3
*****

TST52: SCOPE
        JSR    RS,    PUSH4    ;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,    SSP      ;SAVE "STACK POINTER"
        CMPB   #004,    SPSW   ;CHECK PS (EXCEPT T BIT)
        BEQ    .+6            ;BRANCH IF OK
        HLT    .              ;PS NOT EQUAL TO 004
        371              ;THE ERROR NUMBER IS 371

        CMP    #STACK4,SSP     ;CHECK THE STACK POINTER (R3)
        BEQ    .+6            ;BRANCH IF OK
        HLT    .              ;STACK POINTER (R3) NOT EQUAL TO #STACK4
        372              ;THE ERROR NUMBER IS 372

        TST    ANS1            ;CHECK FIRST HALF OF ANSWER
        BEQ    .+6            ;BRANCH IF OK
        HLT+2  .              ;ANS1 NOT EQUAL TO 000000
        373              ;THE ERROR NUMBER IS 373

        TST    ANS2            ;CHECK SECOND HALF OF ANSWER
        BEQ    .+6            ;BRANCH IF OK
        HLT+2  .              ;ANS2 NOT EQUAL TO 000000
        374              ;THE ERROR NUMBER IS 374

        CMPB   #52,    $TESTN  ;CHECK THE TEST NUMBER
        BEQ    .+6            ;BRANCH IF OK
        HLT    .              ;WRONG TEST! PC MUST HAVE FOULED UP.
        375              ;THE ERROR NUMBER IS 375

```



```

(1) ;TEST 54: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
(1) ; 025252,125251 / 065252,125252 ==> UNDERFLOW
(1) ; PS(ON STACK) = 012, STACK POINTER = R1
(1) ;*****
(1)
(1) 011642 104400 SCOPE
(1) 011644 004567 003514 TST54: JSR RS, PUSH4 ;PUSH 4 WORDS ONTO R1 STACK, SET PRIORITY
(1) 011650 065252 125252 ;WORD 065252,125252 ;SECOND OPERAND ON TOP
(1) 011654 025252 125251 ;WORD 025252,125251 ;FIRST OPERAND ON BOTTOM
(1) 011660 000015 ;WORD 015 ;PROCESSOR PRIORITY LEVEL
(1) 011662 011714 000300 ;WORD ISRS4, 300 ;FIS TRAP VECTOR
(1) 011666 012701 000510 MOV #STACK0,R1 ;SET UP R1 AS STACK POINTER
(1)
(1) 011672 000240 NOP
(1) 011674 075031 FDIV R1 ;FLOATING DIVIDE ON THE R1 STACK
(1)
(1) 011676 004767 003514 RTAS4: JSR PC, POPR ;POP THE "ANSWER"
(1) 011702 010167 166526 MOV R1, $SP ;SAVE STACK POINTER (R1)
(1) 011706 104002 HLT+2 ;FIS TRAP DIDN'T OCCURE!
(3) 011710 000404 404 ;THE ERROR NUMBER IS 404
(1) 011712 000464 BR END54
(1)
(1) 011714 004767 003526 ISRS4: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
(1) 011720 010167 166510 MOV R1, $SP ;SAVE STACK POINTER (R1)
(1) 011724 122767 000300 166500 CMPB #300, $PSW ;CHECK PS AFTER FIS TRAP
(1) 011732 001402 BEQ .+6 ;BRANCH IF OK
(1) 011734 104000 HLT ;PS AFTER FIS TRAP NOT EQUAL TO 300
(3) 011736 000405 405 ;THE ERROR NUMBER IS 405
(1)
(1) 011740 022767 000510 166466 CMP #STACK0,$SP ;CHECK THE STACK POINTER (R1)
(1) 011746 001402 BEQ .+6 ;BRANCH IF OK
(1) 011750 104000 HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK0
(3) 011752 000406 406 ;THE ERROR NUMBER IS 406
(1)
(1) 011754 022767 011676 166454 CMP #RTAS4, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 011762 001402 BEQ .+6 ;BRANCH IF OK
(1) 011764 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 011766 000407 407 ;THE ERROR NUMBER IS 407
(1)
(1) 011770 022767 000012 166442 CMP #012, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 011776 001402 BEQ .+6 ;BRANCH IF OK
(1) 012000 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 012
(3) 012002 000410 410 ;THE ERROR NUMBER IS 410
(1)
(1) 012004 022767 065252 166430 CMP #065252,ANS3 ;CHECK DATA FROM THE STACK
(1) 012012 001402 BEQ .+6 ;BRANCH IF OK
(1) 012014 104004 HLT+4 ;DATA ON STACK (065252) CHANGED
(3) 012016 000411 411 ;THE ERROR NUMBER IS 411
(1)
(1) 012020 022767 125252 166416 CMP #125252,ANS4 ;CHECK DATA FROM STACK
(1) 012026 001402 BEQ .+6 ;BRANCH IF OK
(1) 012030 104004 HLT+4 ;DATA ON STACK (125252) CHANGED
(3) 012032 000412 412 ;THE ERROR NUMBER IS 412
(1)
(1) 012034 022767 025252 166404 CMP #025252,ANS5 ;CHECK DATA FROM STACK
(1) 012042 001402 BEQ .+6 ;BRANCH IF OK

```

(1)	012044	104006			HLT+6		;DATA ON STACK (025252) CHANGED
(3)	012046	000413			413		;THE ERROR NUMBER IS 413
(1)							
(1)	012050	022767	125251	166372	CMP	#125251,ANS6	;CHECK DATA FROM STACK
(1)	012056	001402			BEQ	.+6	;BRANCH IF OK
(1)	012060	104006			HLT+6		;DATA ON STACK (125251) CHANGED
(3)	012062	000414			414		;THE ERROR NUMBER IS 414
(1)							
(1)	012064	122767	000054	166312	ENDS4:	CMPB	;CHECK THE TEST NUMBER
(1)	012072	001402			BEQ	.+6	;BRANCH IF OK
(1)	012074	104000			HLT		;WRONG TEST! PC MUST HAVE FOULED UP.
(3)	012076	000415			415		;THE ERROR NUMBER IS 415
(1)							
(1)							

7227

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 012100 104400
(1) 012102 004567 003256
(1) 012106 127652 125252
(1) 012112 067452 125252
(1) 012116 000242
(1) 012120 012152 000357
(1) 012124 012704 000510
(1)
(1) 012130 000240
(1) 012132 075034
(1)
(1) 012134 004767 003256
(1) 012140 010467 166270
(1) 012144 104002
(3) 012146 000416
(1) 012150 000464
(1)
(1) 012152 004767 003270
(1) 012156 010467 166252
(1) 012162 122767 000357 166242
(1) 012170 001402
(1) 012172 104000
(3) 012174 000417
(1)
(1) 012176 022767 000510 166230
(1) 012204 001402
(1) 012206 104000
(3) 012210 000420
(1)
(1) 012212 022767 012134 166216
(1) 012220 001402
(1) 012222 104001
(3) 012224 000421
(1)
(1) 012226 022767 000202 166204
(1) 012234 001402
(1) 012236 104002
(3) 012240 000422
(1)
(1) 012242 022767 127652 166172
(1) 012250 001402
(1) 012252 104004
(3) 012254 000423
(1)
(1) 012256 022767 125252 166160
(1) 012264 001402
(1) 012266 104004
(3) 012270 000424
(1)

```

 TEST 55: FDIV (LSI-1) FLOATING DIVIDE INSTRUCTION)
 067452,125252 / 127652,125252 ==> OVERFLOW
 PS(ON STACK) = 202, STACK POINTER = R4

TST55: SCOPE
 JSR RS, 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 ISR55, 357 ; FIS TRAP VECTOR
 MOV #STACK0,R4 ; SET UP R4 AS STACK POINTER

NOP
 FDIV R4 ; FLOATING DIVIDE ON THE R4 STACK

RTA55: JSR PC, POPR ; POP THE "ANSWER"
 MOV R4, \$SP ; SAVE STACK POINTER (R4)
 HLT+2 ; FIS TRAP DIDN'T OCCURE!
 416 ; THE ERROR NUMBER IS 416
 BR END55

ISR55: JSR PC, POPER ; POP ALL DATA OFF THE STACKS
 MOV R4, \$SP ; SAVE STACK POINTER (R4)
 CMPB #357, \$PSW ; CHECK PS AFTER FIS TRAP
 BEQ .+6 ; BRANCH IF OK
 HLT ; PS AFTER FIS TRAP NOT EQUAL TO 357
 417 ; THE ERROR NUMBER IS 417

CMP #STACK0,\$SP ; CHECK THE STACK POINTER (R4)
 BEQ .+6 ; BRANCH IF OK
 HLT ; STACK POINTER (R4) NOT EQUAL TO #STACK0
 420 ; THE ERROR NUMBER IS 420

CMP #RTA55, ANS1 ; CHECK FIS TRAP RETURN ADDRESS
 BEQ .+6 ; BRANCH IF OK
 HLT+1 ; FIS TRAP AT WRONG ADDRESS
 421 ; THE ERROR NUMBER IS 421

CMP #202, ANS2 ; CHECK PS BEFORE FIS TRAP
 BEQ .+6 ; BRANCH IF OK
 HLT+2 ; PS AT FIS TRAP TIME NOT 202
 422 ; THE ERROR NUMBER IS 422

CMP #127652,ANS3 ; CHECK DATA FROM THE STACK
 BEQ .+6 ; BRANCH IF OK
 HLT+4 ; DATA ON STACK (127652) CHANGED
 423 ; THE ERROR NUMBER IS 423

CMP #125252,ANS4 ; CHECK DATA FROM STACK
 BEQ .+6 ; BRANCH IF OK
 HLT+4 ; DATA ON STACK (125252) CHANGED
 424 ; THE ERROR NUMBER IS 424

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-60
 DVKACA.SRC TEST FLOATING DIV INSTRUCTION WITH OVERFLOW

SEQ 0062

(1)	012272	022767	067452	166146	CMP	#067452,ANS5	;CHECK DATA FROM STACK
(1)	012300	001402			BEQ	+.6	;BRANCH IF OK
(1)	012302	104006			HLT+6		;DATA ON STACK (067452) CHANGED
(3)	012304	000425			425		;THE ERROR NUMBER IS 425
(1)							
(1)	012306	022767	125252	166134	CMP	#125252,ANS6	;CHECK DATA FROM STACK
(1)	012314	001402			BEQ	+.6	;BRANCH IF OK
(1)	012316	104006			HLT+6		;DATA ON STACK (125252) CHANGED
(3)	012320	000426			426		;THE ERROR NUMBER IS 426
(1)							
(1)	012322	122767	000055	166054	END55:	CMPB	#55, \$TESTN
(1)	012330	001402			BEQ	+.6	;CHECK THE TEST NUMBER
(1)	012332	104000			HLT		;BRANCH IF OK
(3)	012334	000427			427		;WRONG TEST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 427
(1)							

```

7231
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 012336 104400
(1) 012340 004567 003020
(1) 012344 100125 125252
(1) 012350 052525 052525
(1) 012354 000047
(1) 012356 012410 000113
(1) 012362 012705 000510
(1)
(1) 012366 000240
(1) 012370 075035
(1)
(1) 012372 004767 003020
(1) 012376 010567 166032
(1) 012402 104002
(3) 012404 000430
(1) 012406 000464
(1)
(1) 012410 004767 003032
(1) 012414 010567 166014
(1) 012420 122767 000113 166004
(1) 012426 001402
(1) 012430 104000
(3) 012432 000431
(1)
(1) 012434 022767 000510 165772
(1) 012442 001402
(1) 012444 104000
(3) 012446 000432
(1)
(1) 012450 022767 012372 165760
(1) 012456 001402
(1) 012460 104001
(3) 012462 000433
(1)
(1) 012464 022767 000013 165746
(1) 012472 001402
(1) 012474 104002
(3) 012476 000434
(1)
(1) 012500 022767 100125 165734
(1) 012506 001402
(1) 012510 104004
(3) 012512 000435
(1)
(1) 012514 022767 125252 165722
(1) 012522 001402
(1) 012524 104004
(3) 012526 000436
(1)

```

```

*****
TEST 56: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
          052525,052525 / 100125,125252 ==> DIVIDE BY ZERO
          PS(ON STACK) = 013, STACK POINTER = R5
*****

TST56:  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 ISR56, 113        ;FIS TRAP VECTOR
        MOV   #STACK0,R5        ;SET UP R5 AS STACK POINTER

        NOP
        FDIV  R5                ;FLOATING DIVIDE ON THE R5 STACK

RTA56:  JSR   PC,    POPR        ;POP THE "ANSWER"
        MOV   R5,    $SP        ;SAVE STACK POINTER (R5)
        HLT+2 430              ;FIS TRAP DIDN'T OCCURE!
        BR    END56            ;THE ERROR NUMBER IS 430

ISR56:  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   431              ;PS AFTER FIS TRAP NOT EQUAL TO 113
        BR    431              ;THE ERROR NUMBER IS 431

        CMP   #STACK0,$SP      ;CHECK THE STACK POINTER (R5)
        BEQ   .+6              ;BRANCH IF OK
        HLT   432              ;STACK POINTER (R5) NOT EQUAL TO #STACK0
        BR    432              ;THE ERROR NUMBER IS 432

        CMP   #RTA56, ANS1     ;CHECK FIS TRAP RETURN ADDRESS
        BEQ   .+6              ;BRANCH IF OK
        HLT+1 433              ;FIS TRAP AT WRONG ADDRESS
        BR    433              ;THE ERROR NUMBER IS 433

        CMP   #013, ANS2       ;CHECK PS BEFORE FIS TRAP
        BEQ   .+6              ;BRANCH IF OK
        HLT+2 434              ;PS AT FIS TRAP TIME NOT 013
        BR    434              ;THE ERROR NUMBER IS 434

        CMP   #100125,ANS3     ;CHECK DATA FROM THE STACK
        BEQ   .+6              ;BRANCH IF OK
        HLT+4 435              ;DATA ON STACK (100125) CHANGED
        BR    435              ;THE ERROR NUMBER IS 435

        CMP   #125252,ANS4     ;CHECK DATA FROM STACK
        BEQ   .+6              ;BRANCH IF OK
        HLT+4 436              ;DATA ON STACK (125252) CHANGED
        BR    436              ;THE ERROR NUMBER IS 436

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-62
 DVKACA.SRC TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO

SEQ 0064

(1)	012530	022767	052525	165710	CMP	#052525,ANS5	;CHECK DATA FROM STACK
(1)	012536	001402			BEQ	.+6	;BRANCH IF OK
(1)	012540	104006			HLT+6		;DATA ON STACK (052525) CHANGED
(3)	012542	000437			437		;THE ERROR NUMBER IS 437
(1)							
(1)	012544	022767	052525	165676	CMP	#052525,ANS6	;CHECK DATA FROM STACK
(1)	012552	001402			BEQ	.+6	;BRANCH IF OK
(1)	012554	104006			HLT+6		;DATA ON STACK (052525) CHANGED
(3)	012556	000440			440		;THE ERROR NUMBER IS 440
(1)							
(1)	012560	122767	000056	165616	END56:	CMPB	#56, \$TESTN
(1)	012566	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	012570	104000			HLT		;BRANCH IF OK
(3)	012572	000441			441		;WRONG TEST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 441
(1)							

7232

```

(1) ;*****
(1) ;TEST 57: FDIV (LSI-11 FLOATING DIVIDE INSTRUCTION)
(1) ;100052,052525 / 000006,123456 ==> DIVIDE BY ZERO
(1) ;PS(ON STACK) = 213, STACK POINTER = SP
(1) ;*****
(1)
(1) 012574 104400
(1) 012576 004567 002410 TST57: JSR RS, PUSH5 ;PUSH 4 WORDS ONTO STACK, SET PRIORITY
(1) 012602 000006 123456 ;.WORD 000006,123456 ;SECOND OPERAND ON TOP
(1) 012606 100052 052525 ;.WORD 100052,052525 ;FIRST OPERAND ON BOTTOM
(1) 012612 000357 ;.WORD 357 ;PROCESSOR PRIORITY LEVEL
(1) 012614 012642 000311 ;.WORD ISR57, 311 ;FIS TRAP VECTOR
(1)
(1) 012620 000240
(1) 012622 075036 NOP
(1) ;FDIV SP ;FLOATING DIVIDE ON THE STACK
(1)
(1) 012624 004767 002422 RTA57: JSR PC, POPS ;POP THE "ANSWER"
(1) 012630 104002 ;HLT+2 ;FIS TRAP DIDN'T OCCURE!
(3) 012632 000442 442 ;THE ERROR NUMBER IS 442
(1) 012634 012706 000600 MOV #BEGIN, SP ;RESTORE THE STACK POINTER
(1) 012640 000464 BR END57
(1)
(1) 012642 004767 002436 ISR57: JSR PC, POPES ;POP ALL DATA OFF THE STACK
(1) 012646 022706 000600 ;CMP #BEGIN, SP ;CHECK THE STACK POINTER
(1) 012652 001405 ;BEQ ISA57 ;BRANCH IF OK
(1) 012654 012706 000600 ;MOV #BEGIN, SP ;RESTORE THE STACK POINTER
(1) 012660 104000 ;HLT ;STACK POINTER FOULED UP
(3) 012662 000443 443 ;THE ERROR NUMBER IS 443
(1) 012664 000452 BR END57 ;SKIP REST OF TEST
(1)
(1) 012666 122767 000311 165536 ISA57: CMPB #311, $PSW ;CHECK PS AFTER FIS TRAP
(1) 012674 001402 ;BEQ .+6 ;BRANCH IF OK
(1) 012676 104000 ;HLT ;PS AFTER FIS TRAP NOT EQUAL TO 311
(3) 012700 000444 444 ;THE ERROR NUMBER IS 444
(1)
(1) 012702 022767 012624 165526 CMP #RTA57, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 012710 001402 ;BEQ .+6 ;BRANCH IF OK
(1) 012712 104001 ;HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 012714 000445 445 ;THE ERROR NUMBER IS 445
(1)
(1) 012716 022767 000213 165514 CMP #213, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 012724 001402 ;BEQ .+6 ;BRANCH IF OK
(1) 012726 104002 ;HLT+2 ;PS AT FIS TRAP TIME NOT 213
(3) 012730 000446 446 ;THE ERROR NUMBER IS 446
(1)
(1) 012732 022767 000006 165502 CMP #000006, ANS3 ;CHECK DATA FROM THE STACK
(1) 012740 001402 ;BEQ .+6 ;BRANCH IF OK
(1) 012742 104004 ;HLT+4 ;DATA ON STACK (000006) CHANGED
(3) 012744 000447 447 ;THE ERROR NUMBER IS 447
(1)
(1) 012746 022767 123456 165470 CMP #123456, ANS4 ;CHECK DATA FROM STACK
(1) 012754 001402 ;BEQ .+6 ;BRANCH IF OK
(1) 012756 104004 ;HLT+4 ;DATA ON STACK (123456) CHANGED
(3) 012760 000450 450 ;THE ERROR NUMBER IS 450
(1)

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-64
 DVKACA.SRC TEST FLOATING DIV. INSTRUCTION FOR DIVIDE BY ZERO

SEQ 0066

(1)	012762	022767	100052	165456	CMP	#100052,ANS5	;CHECK DATA FROMONG TK
(1)	012770	001402			BEQ	.+6	;BRANCH IF OK
(1)	012772	104006			HLT+6		;DATA ON STACK (100052) CHANGED
(3)	012774	000451			451		;THE ERROR NUMBER IS 451
(1)							
(1)	012776	022767	052525	165444	CMP	#052525,ANS6	;CHECK DATA FROM STACK
(1)	013004	001402			BEQ	.+6	;BRANCH IF OK
(1)	013006	104006			HLT+6		;DATA ON STACK (052525) CHANGED
(3)	013010	000452			452		;THE ERROR NUMBER IS 452
(1)							
(1)	013012	122767	000057	165364	END57:	CMPB	#57, \$TESTN
(1)	013020	001402			BEQ	.+6	;CHECK THE TEST NUMBER
(1)	013022	104000			HLT		;BRANCH IF OK
(3)	013024	000453			453		;WREST! PC MUST HAVE FOULED UP.
(1)							;THE ERROR NUMBER IS 453
(1)							

(1)				
(1)				
(1)				
(1)				
(1)				
(1)				
(1)				
(1)				
(1)	013026	104400		
(1)	013030	012704	000532	
(1)	013034	012744	107070	
(1)	013040	012744	134343	
(1)	013044	012744	065432	
(1)	013050	012744	032107	
(1)	013054	012744	123456	
(1)	013060	012744	045670	
(1)	013064	012744	125252	
(1)	013070	012744	135252	
(1)	013074	012744	016161	
(1)	013100	012744	040616	
(2)	013104			
(2)	013104	106427		
(1)				
(1)	013110	000240		
(1)	013112	075014		
7238	013114	075034		
(1)	013116	075024		
(1)	013120	075004		
(1)				
(2)	013122			
(2)	013122	106767		
(1)	013126	042767	000020	165276
(1)	013134	012467	165276	
(1)	013140	012467	165274	
(1)	013144	010467	165264	
(1)	013150	122767	000010	165254
(1)	013156	001402		
(1)	013160	104000		
(3)	013162	000454		
(1)				
(1)	013164	022767	000532	165242
(1)	013172	001402		
(1)	013174	104000		
(3)	013176	000455		
(1)				
(1)				
(1)	013200	022767	137201	165230
(1)	013206	001402		
(1)	013210	104002		
(3)	013212	000456		
(1)				
(1)	013214	022767	115230	165216
(1)	013222	001402		
(1)	013224	104002		
(3)	013226	000457		

```

*****
TEST 60:      TEST ALL INSTRUCTION TOGETHER
               032107,065432 * 045670,123456
               134343,107070 + ----- = 137201,115230
               (135252,125252 - 040616,016161)
               PS=010, STACK POINTER=R4
*****

TST60:  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
FSUB      R4              ;135252,125252-040616,016161=140616,017434
FDIV      R4              ;045670,123456/140616,017434=145246,047065
FMUL      R4              ;032107,065432*145246,047065=137201,106137
FADD      R4              ;134343,107070+137201,106137=137201,115230

MFPS      $PSW            ;SAVE FINAL PS
.WORD     106700!..C
BIC       #20             $PSW      ;CLR T-BIT
MOV       (R4)+,ANS1      ;SAVE FIRST HALF OF ANSWER
MOV       (R4)+,ANS2      ;SAVE SECOND HALF OF ANSWER
MOV       R4,$SP          ;SAVE STACK POINTER
CMPB      #010,$PSW       ;CHECK PS (EXCEPT T BIT)
BEQ       .+6             ;BRANCH IF OK
HLT       ;PS NOT EQUAL TO 010
454       ;THE ERROR NUMBER IS 454

CMP        #STAK10,$SP     ;CHECK THE STACK POINTER (R4)
BEQ       .+6             ;BRANCH IF OK
HLT       ;STACK POINTER (R4) NOT EQUAL TO THE
455       ;THE ERROR NUMBER IS 455
          ;ADDRESS OF STAK10

CMP        #137201,ANS1    ;CHECK FIRST HALF OF ANSWER
BEQ       .+6             ;BRANCH IF OK
HLT+2     ;ANS1 NOT EQUAL TO 137201
456       ;THE ERROR NUMBER IS 456

CMP        #115230,ANS2    ;CHECK SECOND HALF OF ANSWER
BEQ       .+6             ;BRANCH IF OK
HLT+2     ;ANS2 NOT EQUAL TO 115230
457       ;THE ERROR NUMBER IS 457

```

006

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-66
DVKACA.SRC TEST OF ALL FIS AT ONCE

SE0 0068

```

(1) 013230 122767 000060 165146 END60: CMPB    #60,    $TESTN ;CHECK THE TEST NUMBER
(1) 013236 001402          BEQ      .+6      ;BRANCH IF OK
(1) 013240 104000          HLT          ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 013242 000460          460          ;THE ERROR NUMBER IS 460
(1)

```

7243

```

(1)
(1)
(1)
(1)
(1)
(1) 013244 104400
(1) 013246 012737 013336 000004 TST61: MOV #ISR61, 2#4 ;SET UP ADDRESS TRAP VECTOR
(1) 013254 012737 000340 000006 MOV #340, 2#6
(1) 013262 004567 002076 JSR R5, PUSH4 ;PUSH 4 WORDS ONTO R2 STACK, SET PRIORITY
(1) 013266 070707 016161 .WORD 070707,016161 ;SECOND OPERAND ON TOP
(1) 013272 146314 143434 .WORD 146314,143434 ;FIRST OPERAND ON BOTTOM
(1) 013276 000143 .WORD 143 ;PROCESSOR PRIORITY LEVEL
(1) 013300 015634 000340 .WORD TRAPER, 340 ;FIS TRAP VECTOR
(2) 013304 MTPS #143 ;SET PROCESSOR STATUS
(2) 013304 106427 .WORD 106400!..C
(1) 013310 012702 177777 MOV #177777,R2 ;SET UP R2 AS STACK POINTER
(1)
(1) 013314 000240 NOP
(1) 013316 075002 FAOD R2 ;FLOATING ADD ON THE R2 STACK
(1)
(1) 013320 004767 002072 RTA61: JSR PC, POPR ;POP THE "ANSWER"
(1) 013324 010267 165104 MOV R2, $SP ;SAVE STACK POINTER (R2)
(1) 013330 104002 HLT+2 ;FIS TRAP DIDN'T OCCURE!
(3) 013332 00461 461 ;THE ERROR NUMBER IS 461
(1) 013334 000464 BR END61
(1)
(1) 013336 004767 002104 ISR61: JSR PC, POPR ;POP ALL DATA OFF THE STACKS
(1) 013342 010267 165066 MOV R2, $SP ;SAVE STACK POINTER (R2)
(1) 013346 122767 000340 165056 CMPB #340, $PSW ;CHECK PS AFTER ADR. ERR. TRAP
(1) 013354 001402 BEQ .+6 ;BRANCH IF OK
(1) 013356 104000 HLT ;PS AFTER TRAP NOT EQUAL TO 340
(3) 013360 000462 462 ;THE ERROR NUMBER IS 462
(1)
(1) 013362 022767 177777 165044 CMP #177777,$SP ;CHECK THE STACK POINTER (R2)
(1) 013370 001402 BEQ .+6 ;BRANCH IF OK
(1) 013372 104000 HLT ;STACK POINTER (R2) NOT EQUAL TO #177777
(3) 013374 000463 463 ;THE ERROR NUMBER IS 463
(1)
(1) 013376 022767 013320 165032 CMP #RTA61, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 013404 001402 BEQ .+6 ;BRANCH IF OK
(1) 013406 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 013410 000464 464 ;THE ERROR NUMBER IS 464
(1)
(1) 013412 022767 000151 165020 CMP #151, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 013420 001402 BEQ .+6 ;BRANCH IF OK
(1) 013422 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 151
(3) 013424 000465 465 ;THE ERROR NUMBER IS 465
(1)
(1) 013426 022767 070707 165006 CMP #070707,ANS3 ;CHECK DATA FROM THE STACK
(1) 013434 001402 BEQ .+6 ;BRANCH IF OK
(1) 013436 104004 HLT+4 ;DATA ON STACK (070707) CHANGED
(3) 013440 000466 466 ;THE ERROR NUMBER IS 466
(1)
(1) 013442 022767 016161 164774 CMP #016161,ANS4 ;CHECK DATA FROM STACK
(1) 013450 001402 BEQ .+6 ;BRANCH IF OK

```

(1)	013452	104004			HLT+4		;DATA ON STACK (016161) CHANGED
(3)	013454	000467			467		;THE ERROR NUMBER IS 467
(1)							
(1)	013456	022767	146314	164762	CMP	#146314,ANS5	;CHECK DATA FROM STACK
(1)	013464	001402			BEQ	.+6	;BRANCH IF OK
(1)	013466	104006			HLT+6		;DATA ON STACK (146314) CHANGED
(3)	013470	000470			470		;THE ERROR NUMBER IS 470
(1)							
(1)	013472	022767	143434	164750	CMP	#143434,ANS6	;CHECK DATA FROM STACK
(1)	013500	001402			BEQ	.+6	;BRANCH IF OK
(1)	013502	104006			HLT+6		;DATA ON STACK (143434) CHANGED
(3)	013504	000471			471		;THE ERROR NUMBER IS 471
(1)	013506	122767	000061	164670	CMPB	#61, STESTN	;CHECK THE TEST NUMBER
(1)	013514	001402			BEQ	.+6	;BRANCH IF OK
(1)	013516	104000			HLT		;WRONG TEST! PC MUST HAVE FOULED UP.
(3)	013520	000472			472		;THE ERROR NUMBER IS 472

7244

```

(1)
(1)
(1)
(1)
(1)
(1) 013522 104400
(1) 013524 012737 013614 000004 TST62: MOV #ISR62, 3#4 ;SET UP ADDRESS TRAP VECTOR
(1) 013532 012737 000340 000006 MOV #340, 3#6
(1) 013540 004567 001620 JSR RS, PUSH#R ;PUSH 4 WORDS ONTO RS STACK, SET PRIORITY
(1) 013544 065432 123456 .WORD 065432,123456 ;SECOND OPERAND ON TOP
(1) 013550 037654 032107 .WORD 037654,032107 ;FIRST OPERAND ON BOTTOM
(1) 013554 000202 .WORD 202 ;PROCESSOR PRIORITY LEVEL
(1) 013556 015634 000340 .WORD TRAPER, 340 ;FIS TRAP VECTOR
(2) 013562 MTPS #202 ;SET PROCESSOR STATUS
(2) 013562 106427 .WORD 106400!..C
(1) 013566 012705 160000 MOV #160000,RS ;SET UP RS AS STACK POINTER
(1)
(1) 013572 000240 NOP
(1) 013574 075025 FMUL RS ;FLOATING MULTIPLY ON THE RS STACK
(1)
(2) 013576 RTA62: MFPS $PSW ;SAVE THE PSW
(2) 013576 106767 .WORD 106700!..C
(1) 013602 010567 164626 MOV RS, $SP ;SAVE STACK POINTER (RS)
(1) 013606 104000 HLT ;FIS TRAP DIDN'T OCCURE!
(3) 013610 000473 473 ;THE ERROR NUMBER IS 473
(1) 013612 000434 BR END62
(1)
(1) 013614 004767 001626 ISR62: JSR PC, POPER ;POP ALL DATA OFF THE STACKS
(1) 013620 010567 164610 MOV RS, $SP ;SAVE STACK POINTER (RS)
(1) 013624 122767 000340 164600 CMPB #340, $PSW ;CHECK PS AFTER ADR. ERR. TRAP
(1) 013632 001402 BEQ .+6 ;BRANCH IF OK
(1) 013634 104000 HLT ;PS AFTER TRAP NOT EQUAL TO 340
(3) 013636 000474 474 ;THE ERROR NUMBER IS 474
(1)
(1) 013640 022767 160000 164566 CMP #160000,$SP ;CHECK THE STACK POINTER (RS)
(1) 013646 001402 BEQ .+6 ;BRANCH IF OK
(1) 013650 104000 HLT ;STACK POINTER (RS) NOT EQUAL TO #160000
(3) 013652 000475 475 ;THE ERROR NUMBER IS 475
(1)
(1) 013654 022767 013576 164554 CMP #RTA62, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 013662 001402 BEQ .+6 ;BRANCH IF OK
(1) 013664 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 013666 000476 476 ;THE ERROR NUMBER IS 476
(1)
(1) 013670 022767 000210 164542 CMP #210, ANS2 ;CHECK PS BEFORE FIS TRAP
(1) 013676 001402 BEQ .+6 ;BRANCH IF OK
(1) 013700 104002 HLT+2 ;PS AT FIS TRAP TIME NOT 210
(3) 013702 000477 477 ;THE ERROR NUMBER IS 477
(1)
(1) 013704 122767 000062 164472 END62: CMPB #62, $TESTN ;CHECK THE TEST NUMBER
(1) 013712 001402 BEQ .+6 ;BRANCH IF OK
(1) 013714 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 013716 000500 500 ;THE ERROR NUMBER IS 500

```

PC	OP	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418	OP419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```

(3) 014166 000501          501          ;THE ERROR NUMBER IS 501
(1) 014170 000457          BR          EN063 ;SKIP REST OF TEST
(1)
(1) 014172 010167 164236 65: MOV R1, SSP ;SAVE STACK POINTER
(1) 014176 122767 000344 164226 CMPB #344, SPSW ;CHECK PS AFTER INTERRUPT
(1) 014204 001402          BEQ .+6 ;BRANCH IF OK
(1) 014206 104000          HLT ;PS AFTER INTERRUPT NOT EQUAL TO LVLA
(3) 014210 000502          502          ;THE ERROR NUMBER IS 502
(1)
(1) 014212 022767 000510 164214 CMP #STACK0, SSP ;CHECK THE STACK POINTER (R1)
(1) 014220 001402          BEQ .+6 ;BRANCH IF OK
(1) 014222 104000          HLT ;STACK POINTER (R1) NOT EQUAL TO #STACK0
(3) 014224 000503          503          ;THE ERROR NUMBER IS 503
(1)
(1) 014226 022767 014042 164202 CMP #RTA63, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 014234 001402          BEQ .+6 ;BRANCH IF OK
(1) 014236 104001          HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 014240 000504          504          ;THE ERROR NUMBER IS 504
(1)
(1)
(1) 014242 022767 000177 164172 CMP #000177, ANS3 ;CHECK DATA FROM THE STACK
(1) 014250 001402          BEQ .+6 ;BRANCH IF OK
(1) 014252 104004          HLT+4 ;DATA ON STACK (000177) CHANGED
(3) 014254 000505          505          ;THE ERROR NUMBER IS 505
(1)
(1) 014256 022767 134543 164160 CMP #134543, ANS4 ;CHECK DATA FROM STACK
(1) 014264 001402          BEQ .+6 ;BRANCH IF OK
(1) 014266 104004          HLT+4 ;DATA ON STACK (134543) CHANGED
(3) 014270 000506          506          ;THE ERROR NUMBER IS 506
(1)
(1) 014272 022767 035700 164146 CMP #035700, ANS5 ;CHECK DATA FROM STACK
(1) 014300 001402          BEQ .+6 ;BRANCH IF OK
(1) 014302 104006          HLT+6 ;DATA ON STACK (035700) CHANGED
(3) 014304 000507          507          ;THE ERROR NUMBER IS 507
(1)
(1) 014306 022767 143235 164134 CMP #143235, ANS6 ;CHECK DATA FROM STACK
(1) 014314 001402          BEQ .+6 ;BRANCH IF OK
(1) 014316 104006          HLT+6 ;DATA ON STACK (143235) CHANGED
(3) 014320 000510          510          ;THE ERROR NUMBER IS 510
(1)
(1) 014322 005367 164206 DEC TEMP ;STAY IN THE LOOP FOR 30 TIMES
(1) 014326 001261          BNE 15
(1)
(1) 014330 022626          END63: CMP (SP)+, (SP)+ ;RESTORE STACK POINTER TO 500
(1) 014332 122767 000063 164044 CMPB #63, $TESTN ;CHECK THE TEST NUMBER
(1) 014340 001402          BEQ .+6 ;BRANCH IF OK
(1) 014342 104000          HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 014344 000511          511          ;THE ERROR NUMBER IS 511
(2) 014346          MTPS #340
(2) 014346 106427          .WORD 106400!..C
(1)

```

7252										
(1)	*****									
(1)	TEST 64: TEST THAT FIS ABORTS PROPERLY WHEN INTERRUPTED									
(1)	107070,070707 * 040200,000000 = 107070,070707									
(1)	PS = .PS, STACK POINTER = RO									
(1)	*****									
(1)	014352	104400			SCOPE					
(1)	014354	132737	000040	000421	TST64:	BITB	#40, 2*SENVM			
(1)	014362	001167				BNE	END64+2			;EXIT THIS TEST IF BIT 5 OF SENVM IS HIGH
(1)	014364	013704	000546			MOV	2*TTYOUT, R4			
(1)	014370	012724	014462			MOV	#ISR64, (R4)+			;SET UP TELEPRINTER INTERRUPT VECTOR
(1)	014374	012714	000340			MOV	#340, (R4)			
(1)	014400	000004	000473			TYPE,	RETURN+1			;RETURN+1 CAN BE REPLACED WITH THE ADDRESS OF RETURN
(1)										;TO TYPE CARRIAGE RETURN, LINE FEED
(1)	014404	012767	014412	164056		MOV	#+6, LAOS			;RESET LOOP ADDRESS
(1)	014412	004567	000746			JSR	RS, PUSHB			;PUSH 4 WORDS ONTO RO STACK, SET PRIORITY
(1)	014416	040200	000000			.WORD	040200,000000			;SECOND OPERAND ON TOP
(1)	014422	107070	070707			.WORD	107070,070707			;FIRST OPERAND ON BOTTOM
(1)	014426	000100				.WORD	100			;PROCESSOR PRIORITY LEVEL
(1)	014430	015634	000340			.WORD	TRAPER, 340			;FIS TRAP VECTOR
(1)	014434	012700	000510			MOV	#STACK0, RO			;SET UP STACK POINTER
(1)	014440	012767	000030	164066		MOV	#30, TEMP			
(1)	014446	112777	000100	164074		MOVB	#100, 2\$TPS			;SET TTY INTERRUPT ENABLE
(1)										
(1)	014454	075020			RTA64:	FMUL	RO			;FLOATING MULTIPLY ON THE STACK
(1)	014456	024040				CMP	-(RO), -(RO)			;RESET THE STACK POINTER FOR NEXT PASS
(1)	014460	000775				BR	RTA64			;REPEAT UNTIL INTERRUPTED
(1)										
(1)	014462	105077	164062		ISR64:	CLRB	2\$TPS			;CLEAR THE INTERRUPT ENABLE
(1)	014466	022716	014454			CMP	#RTA64, (SP)			;CHECK IF INTERRUPT AT FIS INSTR.
(1)	014472	001421				BEQ	3\$			;BRANCH IF IT DID
(1)	014474	022766	014454	000004		CMP	#RTA64, 4(SP)			;CHECK FOR INTERRUPT WITH T-BIT SET
(1)	014502	001420				BEQ	4\$			;BRANCH IF IT DID
(1)	014504	112777	000015	164040	1\$:	MOVB	#15, 2\$TPB			;CONTINUE TO TYPE "CR"
(1)	014512	105777	164032		2\$:	TSTB	2\$TPS			;LOOP HERE UNTILL DONE BIT COMES ON
(1)	014516	100375				BPL	2\$			
(1)	014520	112777	000015	164024		MOVB	#15, 2\$TPB			;TYPE ANOTHER "CR"
(1)	014526	012777	000100	164014		MOV	#100, 2\$TPS			;SET TTY INTERRUPT ENABLE
(1)	014534	000002				RTI				
(1)										
(1)	014536	004767	000704		3\$:	JSR	PC, POPER			;SAVE ALL THE STUFF ON THE STACK
(1)	014542	000403				BR	5\$			
(1)										
(1)	014544	022626			4\$:	CMP	(SP)+, (SP)+			;RESET THE STACK TO IGNORE THE TRACE TRAP
(1)	014546	004767	000700			JSR	PC, POPER1			;POP ALL THE STUFF OFF THE STACK
(1)	014552	005746			5\$:	TST	-(SP)			;SAVE PSW FOR FUTURE RTI
(1)	014554	012746	014454			MOV	#RTA64, -(SP)			;PLACE THE RTI ADDRESS BACK IN SP
(1)	014560	022706	000574			CMP	#BEGIN-4, SP			;CHECK THE STACK POINTER
(1)	014564	001407				BEQ	6\$			;BRANCH IF OK
(1)	014566	010667	163642			MOV	SP, SSP			;SAVE FOR TYPING
(1)	014572	012706	000574			MOV	#BEGIN-4, SP			;RESTORE THE STACK POINTER
(1)	014576	104000				HLT				;STACK POINTER FOULED UP
(3)	014600	000512				512				

```

(1) 014604 010067 163624 6$: MOV R0, SSP ;SAVE STACK POINTER
(1) 014610 122767 000344 163614 CMPB #344, SPSW ;CHECK PS AFTER INTERRUPT
(1) 014616 001402 BEQ .+6 ;BRANCH IF OK
(1) 014620 104000 HLT ;PS AFTER INTERRUPT NOT EQUAL TO LVLA
(3) 014622 000513 513 ;THE ERROR NUMBER IS 513
(1)
(1) 014624 022767 000510 163602 CMP #STACK0, SSP ;CHECK THE STACK POINTER (R0)
(1) 014632 001402 BEQ .+6 ;BRANCH IF OK
(1) 014634 104000 HLT ;STACK POINTER (R0) NOT EQUAL TO #STACK0
(3) 014636 000514 514 ;THE ERROR NUMBER IS 514
(1)
(1) 014640 022767 014454 163570 CMP #RTA64, ANS1 ;CHECK FIS TRAP RETURN ADDRESS
(1) 014646 001402 BEQ .+6 ;BRANCH IF OK
(1) 014650 104001 HLT+1 ;FIS TRAP AT WRONG ADDRESS
(3) 014652 000515 515 ;THE ERROR NUMBER IS 515
(1)
(1)
(1) 014654 022767 040200 163560 CMP #040200, ANS3 ;CHECK DATA FROM THE STACK
(1) 014662 001402 BEQ .+6 ;BRANCH IF OK
(1) 014664 104004 HLT+4 ;DATA ON STACK (040200) CHANGED
(3) 014666 000516 516 ;THE ERROR NUMBER IS 516
(1)
(1) 014670 005767 163550 TST ANS4 ;CHECK DATA FROM STACK
(1) 014674 001402 BEQ .+6 ;BRANCH IF OK
(1) 014676 104004 HLT+4 ;DATA ON STACK (000000) CHANGED
(3) 014700 000517 517 ;THE ERROR NUMBER IS 517
(1)
(1) 014702 022767 107070 163536 CMP #107070, ANS5 ;CHECK DATA FROM STACK
(1) 014710 001402 BEQ .+6 ;BRANCH IF OK
(1) 014712 104006 HLT+6 ;DATA ON STACK (107070) CHANGED
(3) 014714 000520 520 ;THE ERROR NUMBER IS 520
(1)
(1) 014716 022767 070707 163524 CMP #070707, ANS6 ;CHECK DATA FROM STACK
(1) 014724 001402 BEQ .+6 ;BRANCH IF OK
(1) 014726 104006 HLT+6 ;DATA ON STACK (070707) CHANGED
(3) 014730 000521 521 ;THE ERROR NUMBER IS 521
(1)
(1) 014732 005367 163576 DEC TEMP ;STAY IN THE LOOP FOR 30 TIMES
(1) 014736 001262 BNE IS
(1)
(1) 014740 022626 000064 163434 END64: CMP (SP)+, (SP)+ ;RESTORE STACK POINTER TO 500
(1) 014742 122767 000064 163434 CMPB #64, $TESTN ;CHECK THE TEST NUMBER
(1) 014750 001402 BEQ .+6 ;BRANCH IF OK
(1) 014752 104000 HLT ;WRONG TEST! PC MUST HAVE FOULED UP.
(3) 014754 000522 522 ;THE ERROR NUMBER IS 522
(2) 014756 MTPS #340
(2) 014756 106427 .WORD 106400!..C
(1)

```

L06

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-74
DVKACA.SRC INTERUPT ABORT TEST SECTION

SEQ 0076

7253	014762	012767	000377	163546	MOV	#377,	TIMES ;SET NUMBER OF ITERATIONS TO 377
7254	014770	010477	163552		MOV	R4,	TTYOUT ;RESTORE TTY INTERUPT VECTOR
7255	014774	005014			CLR	(R4)	
7256							

```

7258      ;*****
      (1)      .SBTTL  END OF PASS ROUTINE
      (1)      ;*INCREMENT THE PASS NUMBER ($PASS)
      (1)      ;*TYPE "END PASS"
      (1)      ;*IF THERES A MONITOR GO TO IT
      (1)      ;*IF THERE ISN'T JUMP TO BEGIN
      (1)      ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
      (1)      ;*SENDMG CAN BE CHANGED TO 7.
      (1)
      (1)      014776      SEOP:
      (1)      014776      104400      SCOPE
      (1)      015000      005267      163402      INC      $PASS      ;;INCREMENT THE PASS NUMBER
      (1)      015004      042767      100000      163374      BIC      #100000,$PASS      ;;DON'T ALLOW A NEG. NUMBER
      (1)      015012      005327      DEC      (PC)+      ;;LOOP?
      (1)      015014      000001      SEOPCT: .WORD      1
      (1)      015016      003015      BGT      $DOAGN      ;;YES
      (1)      015020      012737      MOV      (PC)+,2(PC)+      ;;RESTORE COUNTER
      (1)      015022      000001      SENDCT: .WORD      1
      (1)      015024      015014      $EOPCT
      (1)      015026      000004      015056      TYPE      ,SENDMG      ;;TYPE "END PASS"
      (1)      015032
      (1)
      (1)      015032      013700      000042      MOV      2#42,R0      ;;GET MONITOR ADDRESS
      (1)      015036      001405      BEQ      $DOAGN      ;;BRANCH IF NO MONITOR
      (1)      015040      000005      RESET
      (1)      015042      004710      SENDAD: JSR      PC,(R0)      ;;CLEAR THE WORLD
      (1)      015044      000240      NOP      ;;GO TO MONITOR
      (1)      015046      000240      NOP      ;;SAVE ROOM
      (1)      015050      000240      NOP      ;;FOR
      (1)      015052      $DOAGN:      ;;ACT11
      (1)      015052      000137      000600      JMP      2#BEGIN      ;;RETURN
      (1)      015056      005015      047105      020104      SENDMG: .ASCII <15><12>/END PASS/
      (1)      015064      040520      051523
      (1)      015070      377      377      000      $ENULL: .BYTE      -1,-1,0      ;;NULL CHARACTER STRING
      (1)      015074      .EVEN
7259
7264      015022      000001      ENDCT: 1

```

7268

;*****

.SBTTL SCOPE ROUTINE

(1)	015074	032737	000400	000422	SCOPE\$:	BIT	#SW08,2#SSWREG	;KILL LDUB OR LOOP ON SPEC. TEST
(1)	015102	001404				BEQ	1\$	
(1)	015104	123767	000422	163272		CMPB	2#SSWREG,\$TESTN	;ON RIGHT TEST? *SW7-0*
(1)	015112	001431				BEQ	OVER\$	
(1)	015114	032737	040000	000422	1\$:	BIT	#SW14,2#SSWREG	;LOOP ON TEST
(1)	015122	001023				BNE	KIT\$	
(1)	015124	032737	004000	000422		BIT	#SW11,2#SSWREG	;KILL ITERATIONS
(1)	015132	001412				BEQ	SVLAD\$	
(1)	015134	105767	163343			TSTB	\$ICNT	
(1)	015140	001404				BEQ	2\$	;BRANCH IF FIRST
(1)	015142	126767	163370	163333		CMPB	TIMES,\$ICNT	;DONE?
(1)	015150	001010				BNE	KIT\$	;BRANCH IF NOT
(1)	015152	112767	000001	163323	2\$:	MOVB	#1,\$ICNT	;FIRST ITERATION
(1)	015160	105267	163220		SVLAD\$:	INCB	\$TESTN	;COUNT TEST NUMBERS
(1)	015164	011667	163300			MOV	(6),LAD\$	;SAVE LOOP ADDRESS
(1)	015170	000002				RTI		;RETURN
(1)								
(1)	015172	105267	163305		KIT\$:	INCB	\$ICNT	
(1)	015176	005767	163266		OVER\$:	TST	LAD\$	;FIRST ONE?
(1)	015202	001766				BEQ	SVLAD\$	
(1)	015204	016716	163260			MOV	LAD\$, (6)	;FUDGE RETURN ADDRESS
(1)	015210	000002				RTI		;FIXES PS

```

7273
7274 ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
7275
7276 PUSH5: TST (SP)+ ;POP STACK BY 1
7277 ADD #10, RS ;POINT TO END OF DATA
7278 MOV -(RS), -(SP) ;PUSH DATA ONTO THE STACK
7279 MOV -(RS), -(SP) ;PUSH DATA ONTO THE STACK
7280 MOV -(RS), -(SP) ;PUSH DATA ONTO THE STACK
7281 MOV -(RS), -(SP) ;PUSH DATA ONTO THE STACK
7282 ADD #10, RS ;POINT TO END OF DATA
7286 MTPS (RS)+ ;SET THE PROCESSOR STATUS
7287 .WORD 106400!..C
7290 INC RS
7291 MOV (RS)+, @FISVEC ;SET UP FIS ERROR TRAP VECTOR
7292 MOV (RS)+, @FISLVL ;TRAP STATUS
7293 JMP (RS) ;RETURN
7294
7295 ;SUBROUTINE TO POP 2 WORDS OFF THE STACK
7296 ;ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7297
7298 POPS: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
7302 .WORD 106700!..C
7303 BIC #20, $PSW ;CLEAR T-BIT
7306 MOV (SP)+, R4 ;SAVE RTS ADDRESS
7307 MOV (SP)+, ANS1 ;SAVE THE ANSWER
7308 MOV (SP)+, ANS2
7309 MOV (SP)+, ANS2
7310 MOV SP, $SP ;SAVE THE STACK POINTER
7311 JMP (R4) ;RETURN
7312
7313 ;SUBROUTINE TO POP 6 WORDS OFF THE STACK.
7314 ;THE FIRST TWO WERE PUT ON BY THE ERROR TRAP,
7315 ;THE LAST FOUR WERE THE ORIGINAL INPUT DATA.
7316 ;ALSO SAVES THE PS AND STACK POINTER.
7317
7318 POPES: MFPS $PSW ;SAVE PROCESSOR STATUS WORD
7322 .WORD 106700!..C
7323 MOV (SP)+, R4 ;SAVE RTS ADDRESS
7326 MOV (SP)+, ANS1 ;SAVE RTI ADDRESS
7327 MOV (SP)+, ANS2 ;SAVE RTI STATUS
7328 MOV (SP), ANS2 ;CLEAR THE T-BIT
7329 BIC #20, ANS2
7330 MOV #18, -(SP)
7331 RTI ;RESTORE THE PROCESSOR STATUS
7332 18: MOV (SP)+, ANS3 ;SAVE DATA
7333 MOV (SP)+, ANS4
7334 MOV (SP)+, ANS5
7335 MOV (SP)+, ANS6
7336 MOV SP, $SP ;SAVE SP
7337 JMP (R4) ;RTS
7338
7339 ;SUBROUTINE TO PUSH 4 WORDS ONTO THE STACK
7340
7341 PUSH4: MOV @STACK0, R4 ;SET R4 TO STACK
7342 MOV (RS)+, (R4)+ ;PUT DATA ON STACK
7343 MOV (RS)+, (R4)+

```

```

7344 015374 012524      MOV      (R5)+, (R4)+ ;
7345 015376 012524      MOV      (R5)+, (R4)+ ;
7349 015400      MTPS      (R5)+ ; SET THE PROCESSOR STATUS
(1) 015400 106425      .WORD 106400!..C
7353 015402 005205      INC      RS
7354 015404 012577 163054  MOV      (R5)+, 2FISVEC ; SET UP FIS ERROR TRAP VECTOR
7355 015410 012577 163052  MOV      (R5)+, 2FISLVL ; TRAP STATUS
7356 015414 000205      RTS      RS ; RETURN
7357
7358
7359 ; SUBROUTINE TO POP 2 WORDS OFF THE STACK
7360 ; ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7361
7365 015416      POPR:  MFPS      $PSW ; SAVE PROCESSOR STATUS WORD
(1) 015416 106767      .WORD 106700!..C
7369 015422 042767 000020 163002  BIC      #20, $PSW ; CLEAR T-BIT
7370 015430 016767 163060 163000  MOV      STACK4, ANS1 ; SAVE THE ANSWER
7371 015436 016767 163054 162774  MOV      STACK6, ANS2 ;
7372 015444 000207      RTS      PC
7373
7374
7375 ; SUBROUTINE TO POP 6 WORDS OFF THE STACKS.
7376 ; THE TWO OFF THE R6 STACK WERE PUT ON BY THE ERROR TRAP
7377 ; THE FOUR OFF THE SOFTWARE STACK WERE THE ORIGINAL INPUT DATA.
7378 ; ALSO SAVES THE PS AND STACK POINTER AFTER THE FIS TRAP.
7379
7383 015446      POPER:  MFPS      $PSW ; SAVE PROCESSOR STATUS WORD
(1) 015446 106767      .WORD 106700!..C
7387 015452 012667 000056      POPER1: MOV      (SP)+, SAVRTS ; SAVE RTS ADDRESS
7388 015456 012667 162754      MOV      (SP)+, ANS1 ; SAVE RTI ADDRESS
7389 015462 011667 162752      MOV      (SP), ANS2 ; SAVE RTI STATUS
7390 015466 042767 000020 162744  BIC      #20, ANS2 ; CLEAR THE T-BIT
7391 015474 012746 015502      MOV      #15, -(SP)
7392 015500 000002      RTI ; RESTORE PROCESSOR STATUS
7393 015502 016767 163002 162732 15:  MOV      STACK0, ANS3 ; SAVE DATA
7394 015510 016767 162776 162726      MOV      STACK2, ANS4 ;
7395 015516 016767 162772 162722      MOV      STACK4, ANS5 ;
7396 015524 016767 162766 162716      MOV      STACK6, ANS6 ;
7397 015532 000137      JMP      2(PC)+ ; SIMULATED RTS
7398 015534 000000      SAVRTS: 0
7399
7400 ; SUBROUTINE TO PUSH 4 WORDS ONTO THE PC STACK
7401
7402 015536 012504      PUSH7: MOV      (R5)+, R4 ; SET R4 TO STACK
7403 015540 012524      MOV      (R5)+, (R4)+ ; PUT DATA ON STACK
7404 015542 012524      MOV      (R5)+, (R4)+ ;
7405 015544 012524      MOV      (R5)+, (R4)+ ;
7406 015546 012524      MOV      (R5)+, (R4)+ ;
7410 015550      MTPS      (R5)+ ; SET THE PROCESSOR STATUS
(1) 015550 106425      .WORD 106400!..C
7414 015552 005205      INC      RS
7415 015554 012577 162704      MOV      (R5)+, 2FISVEC ; SET UP FIS ERROR TRAP VECTOR
7416 015560 012577 162702      MOV      (R5)+, 2FISLVL ; TRAP STATUS
7417 015564 000205      RTS      RS ; RETURN
7418
7419 ; SUBROUTINE TO POP 4 WORDS OFF THE PC "STACK"

```

```

7420                                     ; ALSO SAVES THE PROCESSOR STATUS WORD (EXCEPT T BIT)
7421
7425 015566                                     POP7: MFPS SPSW                                     ; SAVE PROCESSOR STATUS WORD
      (1) 015566 106767                                     WORD 106700!..C
7429 015572 042767 000020 162632               BIC #20, SPSW                                     ; CLEAR T-BIT
7430 015600 011600                                     MOV (SP), RO                                     ; GET RETURN ADDRESS
7431 015602 162700 000014               SUB #14, RO                                     ; POINT TO TOP OF "PC STACK"
7432 015606 012067 162624               MOV (RO)+, ANS1                                     ; SAVE 1ST HALF INPUT DATA
7433 015612 012067 162622               MOV (RO)+, ANS2                                     ; SAVE 2ND HALF INPUT DATA
7434 015616 010067 162612               MOV RO, SSP                                     ; SAVE ASSUMED END PC "STACK POINTER"
7435 015622 012067 162614               MOV (RO)+, ANS3                                     ; SAVE 1ST HALF OF ANSWER
7436 015626 012067 162612               MOV (RO)+, ANS4                                     ; SAVE 2ND HALF OF ANSWER
7437 015632 000207               RTS PC
7438
7439                                     ; ERRONIOUS TRAP SERVICE ROUTINE
7440
7441 015634 104000               TRAPER: HLT                                     ; FIS SHOULDN'T HAVE TRAPED
7442 015636 000523               523                                     ; THE ERROR NUMBER IS 523
7443 015640 000002               RTI
7444

```

```

7445      ;*****
(1)      .SBTTL  HLT ROUTINE (ERROR TYPEOUT)
(1)      HLTs:  BIT    #SW10,2#SSWREG  ; SHOULD IT RING THE BELL ON ERROR?
(1)      BEQ    1$                    ; NO - SKIP
(1)      TYPE   SPELL                 ; RING BELL
(1)      INC    ERRORS                ; COUNT THE NUMBER OF ERRORS
(1)      1$:  BIT    #SW13,2#SSWREG  ; SKIP TYPEOUT IF SET
(1)      BNE    2$                    ; SKIP TYPEOUTS
(1)      TYPE   RETURN
(1)      MOV    2(6)+,2#SFATAL        ; PLACE THE ERROR NUMBER IN LOCATION SFATAL
(1)      MOV    -(6),HLTADS          ; PUT ADDRESS OF INSTRUCTION ON STACK
(1)      SUB    #2,HLTADS
(2)      MOV    2(6),TTY              ; TYPE 2(6) IN OCTAL
(1)      JSR    %7,PRINTR            ; TYPE LEADING ZERO'S
(1)      ADD    #2,(6)                ; ADJUST THE RETURN ADDRESS
(1)      TYPE   SPACE+3
(1)      JSR    PC,ERRORS             ; GO TO USER ERROR ROUTINE
(1)      2$:  TSTB   $ENV              ; ARE WE RUNNING UNDER APT?
(1)      BEQ    4$                    ; IF NOT THEN GO TO 4$
(1)      INC    2#SMSGTY              ; OTHERWISE INFORM APT
(1)      BR     AND LOOP
(1)      4$:  TST    2#SSWREG          ; HALT ON ERROR
(1)      BPL    .+4                  ; SKIP IF CONTINUE
(1)      HALT   ; HALT ON ERROR!
(1)      BIT    #SW09,2#SSWREG        ; CHECK FOR INHIBIT LOOP ON ERROR
(1)      BNE    .+4                  ; SKIP IF LOOP ON ERROR
(1)      RTI
(1)      CLRB   $ICNT
(1)      JMP    KITS                 ; LOOP ON TEST UNTIL NO ERRORS
(1)      015642 032737 002000 000422
(1)      015650 001402
(1)      015652 000004 000504
(1)      015656 005267 162600
(1)      015662 032737 020000 000422
(1)      015670 001023
(1)      015672 000004 000472
(1)      015676 013637 000402
(1)      015702 014667 162522
(1)      015706 162767 000002 162514
(2)      015714 017605 000000
(2)      015720 004767 000124
(1)      015724 062716 000002
(1)      015730 000004 000500
(1)      015734 004767 000046
(1)      015740 105767 162454 2$:
(1)      015744 001403
(1)      015746 005237 000400
(1)      015752 000777
(1)      015754 005737 000422 4$:
(1)      015760 100001
(1)      015762 000000
(1)      015764 032737 001000 000422
(1)      015772 001001
(1)      015774 000002
(1)      015776 105067 162501
(1)      016002 000167 177164
(1)
7446

```

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 52-81
 DVKACA.SRC HLT ROUTINE (ERROR TYPEOUT)

SEQ 0083

```

7448
7449 ;*****
7450
7451 .SBTTL USER ERROR ROUTINE
7452
7453 016006 117767 162416 162524 ERRORS: MOVB 2HLTADS, TYPCNT ; TYPE COUNT IS LOW BYTE OF HLT
7454 016014 062767 000002 162516 AOB 82, TYPCNT ; TYPE COUNT = X+2
7455 016022 012703 000430 MOV 2HLTADS, R3 ; TOP OF DATA TO BE TYPED
7456 016026
7457 (1) 016026 012305 ERRIS: MOV (R3)+, TTY ; TYPE (R3)+ IN OCTAL
7458 (1) 016030 004767 000014 JSR %7, PRINTR ; TYPE LEADING ZERO'S
7459 016034 000004 000501 TYPE, SPACE+4 ; SPACE
7460 016040 105367 162474 DECB TYPCNT ; CHECK FOR DONE
7461 016044 100370 BPL ERRIS ; BRANCH IF NOT DONE
016046 000207 RTS PC
  
```

```

7463 016050 112767 000001 162466 PRINTR: MOVB #1,.PR ;SET ZERO FILL SWITCH
(1) 016056 000402 BR .+6 ;SKIP
(1) 016060 005067 162460 PRINTS: CLR .PR ;SUPPRESS LEADING ZERO'S
(1) 016064 112767 177772 162453 MOVB #-6,.PR+1 ;SET COUNT
(1) 016072 010446 MOV R4,-(6) ;SAVE R4
(1) 016074 012704 016176 MOV #.PRBUF,R4 ;SET POINTER TO FIRST ASCII CHAR.
(1) 016100 105014 CLRB (4) ;CLEAR FIRST BYTE
(1) 016102 000405 BR .PRF ;ROTATE FIRST BIT
(1) 016104 105014 .PRL: CLRB (4) ;CLEAR BYTE OF CHARACTER
(1) 016106 006105 ROL TTY ;ROTATE BIT INTO C
(1) 016110 106114 ROLB (4) ;PACK IT
(1) 016112 006105 ROL TTY ;ROTATE BIT INTO C
(1) 016114 106114 ROLB (4) ;PACK IT
(1) 016116 006105 .PRF: ROL TTY ;ROTATE BIT INTO C
(1) 016120 106114 ROLB (4) ;PACK IT
(1) 016122 105714 TSTB (4) ;IS IT ZERO?
(1) 016124 001402 BEQ .+6 ;SKIP INC
(1) 016126 105267 162412 INCB .PR ;SET FILL SWITCH
(1) 016132 105767 162406 TSTB .PR ;CHECK FILL SWITCH
(1) 016136 001402 BEQ .+6 ;SKIP BITSET
(1) 016140 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCII CHAR
(1) 016144 105267 162375 INCB .PR+1 ;INC COUNT
(1) 016150 001355 BNE .PRL ;REPEAT
(1) 016152 022704 016176 CMP #.PRBUF,R4 ;EMPTY BUFFER?
(1) 016156 001002 BNE .+6 ;SKIP IF NOT
(1) 016160 112724 000060 MOVB #'0,(4)+ ;LOAD 1 ZERO
(1) 016164 105014 CLRB (4) ;NULL TERMINATOR
(1) 016166 000004 016176 TYPE .PRBUF ;TYPE IT
(1) 016172 012604 MOV (6)+,R4 ;RESTORE R4
(1) 016174 000207 RTS PC ;RETURN
(1) 016176 000004 .PRBUF: .BLKW 4 ;OUTPUT BUFFER

```

```

7465 ;*****
(1)
(1) .SBTTL POWER DOWN AND UP ROUTINES
(1)
(1) :POWER DOWN ROUTINE
(1) 016206 012737 016330 000024 $PWRDN: MOV $SILLUP,2*$PWRVEC ;;SET FOR FAST UP
(1) 016214 012737 000340 000026 MOV $340,2*$PWRVEC+2 ;;PRIO:7
(3) 016222 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
(3) 016224 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
(3) 016226 010246 MOV R2,-(SP) ;;PUSH R2 ON STACK
(3) 016230 010346 MOV R3,-(SP) ;;PUSH R3 ON STACK
(3) 016232 010446 MOV R4,-(SP) ;;PUSH R4 ON STACK
(3) 016234 010546 MOV R5,-(SP) ;;PUSH R5 ON STACK
(1) 016236 010667 000072 MOV SP,$SAVR6 ;;SAVE SP
(1) 016242 012737 016254 000024 MOV $PWRUP,2*$PWRVEC ;;SET UP VECTOR
(1) 016250 000000 HALT
(1) 016252 000776 BR .-2 ;;HANG UP
(1)
(1) :POWER UP ROUTINE
(1) 016254 016706 000054 $PWRUP: MOV $SAVR6,SP ;;GET SP
(1) 016260 005067 000050 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
(1) 016264 005267 000044 IS: INC $SAVR6 ;;WAIT FOR THE INC
(3) 016270 001375 BNE IS ;;OF WORD
(3) 016272 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
(3) 016274 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
(3) 016276 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
(3) 016300 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
(3) 016302 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
(3) 016304 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
(1) 016306 012737 016206 000024 MOV $PWRDN,2*$PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1) 016314 012737 000340 000026 MOV $340,2*$PWRVEC+2 ;;PRIO:7
(1) 016322 000004 TYPE $POWER ;;REPORT THE POWER FAILURE
(1) 016324 016336 $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
(1) 016326 000002 RTI
(1) 016330 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
(1) 016332 000776 BR .-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
(1) 016334 000000 $SAVR6: 0 ;;PUT THE SP HERE
(1) 016336 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
(1) 016344 000122 .EVEN

```

```

7470
7471      ;*      TYPE OUT ROUTINE
7472      ;*      -----
7473      ;*
7474      ;*
7475      ;*      THIS ROUTINE IS USED TO TYPE ASCIZ MESSAGES
7476      ;*
7477
7478 016346 132737 000040 000421 $TYPE: BITB    #40,2,$SENVH ;HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
7479 016354 001007          BNE      3$      ;IF SO THEN RETURN FROM THE SUBROUTINE VIA 3$
7480 016356 010046          MOV      RO,-(SP) ;OTHERWISE SAVE RO
7481 016360 017600 000002          MOV      @2(SP),RO ;GET THE ADDRESS OF THE ASSCIZ STRING
7482 016364 112046          2$:      MOVB    (RO)+,-(SP) ;PUSH THE CHARACTER TO BE TYPED ONTO STACK
7483 016366 001005          BNE      4$      ;BRANCH IF IT IS NOT THE TERMINATOR
7484 016370 005726          TST      (SP)+
7485 016372 012600          MOV      (SP)+,RO ;OTHERWISE RESTORE THE STACK AND RO
7486 016374 062716 000002          3$:      ADD      #2,(SP) ;ADJUST THE RETURN PC
7487 016400 000002          RTI          ;AND RETURN
7488
7489 016402 105777 162142          4$:      TSTB    @STPS ;IS THE PRINTER AVAILABLE?
7490 016406 100375          BPL      4$      ;IF NOT THEN LOOP HERE
7491 016410 112677 162136          MOVB    (SP)+,@STPB ;OUT PUT THE CHARACTER
7492 016414 000763          BR      2$      ;AND GO BACK
7493          000001          .END

```

[illegible]

K07

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 53-1
DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0088

[illegible]

[illegible]

MO7

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 53-3
DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0090

POPER	015446	7163	7168	7189	7206	7223	7227	7231	7243	7244	7251	7252	7383#	
POPER1	015452	7251	7252	7387#										
POPES	015304	7164	7169	7185	7210	7232	7322#							
POPR	015416	7147	7148	7149	7152	7153	7154	7155	7158	7163	7168	7173	7174	7176
		7178	7179	7180	7181	7189	7193	7194	7195	7196	7198	7206	7214	7216
		7217	7223	7227	7231	7243	7365#							
POPS	015252	7150	7151	7156	7157	7164	7169	7175	7177	7185	7197	7210	7215	7232
		7302#												
POP7	015566	7159	7200	7202	7219	7425#								
PRINTR	016050	7445	7456	7463#										
PRINTS	016060	7463#												
PUSHR	015364	7147	7148	7149	7152	7153	7154	7155	7158	7163	7168	7173	7174	7176
		7178	7179	7180	7181	7189	7193	7194	7195	7196	7198	7206	7214	7216
		7217	7223	7227	7231	7243	7244	7251	7252	7341#				
PUSHS	015212	7150	7151	7156	7157	7164	7169	7175	7177	7185	7197	7210	7215	7232
		7276#												
PUSH7	015536	7159	7200	7202	7219	7402#								
PWRVEC=	000024	5306#	7465*											
RESTR	000222	7116#												
RETURN	000472	5362#	7251	7252	7445									
RTA16	003260	7163#												
RTA17	003506	7164#												
RTA20	003742	7168#												
RTA21	004174	7169#												
RTA33	006146	7185#												
RTA34	006404	7189#												
RTA45	010250	7206#												
RTA46	010500	7210#												
RTA54	011676	7223#												
RTA55	012134	7227#												
RTA56	012372	7231#												
RTA57	012624	7232#												
RTA61	013320	7243#												
RTA62	013576	7244#												
RTA63	014042	7251#												
RTA64	014454	7252#												
RO	=%000000	5297#	7111*	7112*	7113	7127*	7135*	7136*	7137*	7138*	7147*	7174*	7179*	7206*
		7214*	7252*	7258*	7430*	7431*	7432	7433	7434	7435	7436	7465*	7480	7481*
		7482	7485*											
R1	=%000001	5298#	7148*	7153*	7168*	7173*	7198*	7223*	7251*	7465*				
R2	=%000002	5299#	7149*	7181*	7194*	7216*	7243*	7465*						
R3	=%000003	5300#	7163*	7178*	7189*	7196*	7217*	7455*	7456	7465*				
R4	=%000004	5301#	7155*	7158*	7176*	7193*	7227*	7237*	7238*	7251*	7252*	7254	7255*	7307*
		7311	7326*	7337	7341*	7342*	7343*	7344*	7345*	7402*	7403*	7404*	7405*	7406*
		7463*	7465*											
RS	=%000005	5302#	7147*	7148*	7149*	7150*	7151*	7152*	7153*	7154*	7155*	7156*	7157*	7158*
		7159*	7163*	7164*	7168*	7169*	7173*	7174*	7175*	7176*	7177*	7178*	7179*	7180*
		7181*	7185*	7189*	7193*	7194*	7195*	7196*	7197*	7198*	7200*	7202*	7206*	7210*
		7214*	7215*	7216*	7217*	7219*	7223*	7227*	7231*	7232*	7243*	7244*	7251*	7252*
		7277*	7278	7279	7280	7281	7282*	7286*	7290*	7291	7292	7293	7342	7343
		7344	7345	7349*	7353*	7354	7355	7356*	7402	7403	7404	7405	7406	7410*
		7414*	7415	7416	7417*	7465*								
SAVRTS	015534	7387*	7398#											
SAVTPS	000506	5367#												
SCOPE =	104400	5307#	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158
		7159	7163	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180

NO7

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 53-4
DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0091

		7181	7185	7189	7193	7194	7195	7196	7197	7198	7200	7202	7206	7210
		7214	7215	7216	7217	7219	7223	7227	7231	7232	7237	7243	7244	7251
		7252	7258											
SCOPE\$	015074	7130	7268#											
SP	=%000006	5304#	7122*	7150*	7151*	7156*	7157*	7164*	7169*	7175*	7177*	7185*	7197*	7210*
		7215*	7232*	7251*	7252*	7276	7278*	7279*	7280*	7281*	7307	7308	7309	7310
		7326	7327	7328	7330*	7332	7333	7334	7335	7336	7387	7388	7389	7391*
		7430	7465*	7480*	7481	7482*	7484	7485	7486*	7491				
SPACE	000475	5363#	7445	7457										
STACK0	000510	5368#	5374	7147	7148	7149	7152	7153	7154	7155	7158	7163	7168	7173
		7174	7176	7178	7179	7180	7181	7189	7193	7194	7195	7196	7198	7206
		7214	7216	7217	7223	7227	7231	7251	7252	7341	7393			
STACK1=	000511	5374#												
STACK2	000512	5369#	7394											
STACK4	000514	5370#	7147	7148	7149	7152	7153	7154	7155	7158	7173	7174	7176	7178
		7179	7180	7181	7193	7194	7195	7196	7198	7214	7216	7217	7370	7395
STACK6	000516	5371#	7371	7396										
STACK8	000520	5372#												
STACK10	000532	5373#	7237	7238										
STK15	003104	7159#												
STK43	007716	7200#												
STK44	010072	7202#												
STK53	011520	7219#												
SVLADS	015160	7268#												
SW08	= 000400	5283	5315#	7268										
SW09	= 001000	5283	5314#	7445										
SW10	= 002000	5283	5313#	7445										
SW11	= 004000	5283	5312#	7268										
SW12	= 010000	5283	5311#											
SW13	= 020000	5283	5310#	7445										
SW14	= 040000	5283	5309#	7268										
SW15	= 100000	5283	5308#											
TEMP	000534	5375#	7251*	7252*										
TIMES	000536	5376#	7110*	7250*	7253*	7268								
TRAPER	015634	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159
		7173	7174	7175	7176	7177	7178	7179	7180	7181	7193	7194	7195	7196
		7197	7198	7200	7202	7214	7215	7216	7217	7219	7243	7244	7251	7252
		7441#												
TSA12	002500	7156#												
TSA13	002632	7157#												
TSA24	004746	7175#												

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 53-5
 DVKACA.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

TST20	003710	7168#							
TST21	004146	7169#							
TST22	004400	7173#							
TST23	004540	7174#							
TST24	004674	7175#							
TST25	005030	7176#							
TST26	005166	7177#							
TST27	005322	7178#							
TST3	001212	7149#							
TST30	005462	7179#							
TST31	005622	7180#							
TST32	005762	7181#							
TST33	006120	7185#							
TST34	006352	7189#							
TST35	006610	7193#							
TST36	006744	7194#							
TST37	007104	7195#							
TST4	001346	7150#							
TST40	007240	7196#							
TST41	007374	7197#							
TST42	007530	7198#							
TST43	007666	7200#							
TST44	010042	7202#							
TST45	010216	7206#							
TST46	010452	7210#							
TST47	010704	7214#							
TST5	001500	7151#							
TST50	011042	7215#							
TST51	011176	7216#							
TST52	011334	7217#							
TST53	011470	7219#							
TST54	011644	7223#							
TST55	012102	7227#							
TST56	012340	7231#							
TST57	012576	7232#							
TST6	001632	7152#							
TST60	013030	7237#							
TST61	013246	7243#							
TST62	013524	7244#							
TST63	013742	7251#							
TST64	014354	7252#							
TST7	001772	7153#							
TTY	=/000005	5303	7445#	7456#	7463#				
TTYOUT	000546	538#	7251	7252	7254#				
TYPCNT	000540	5377#	7453#	7454#	7458#				
TYPE	= 000004	5316#	7251	7252	7258	7445	7457	7463	7465
YESRT	000542	5378#	7123						
\$APTHD	000430	5340#	5341						
\$BELL	000504	5366#	7445						
\$CPUOP	000426	5339#							
\$DEVCT	000410	5339#	7111						
\$OCAGN	015052	7258#							
\$ENDAD	015042	5330	7258#						
\$ENDCT	015022	7258#	7262						
\$ENDMG	015056	7258#							
\$ENULL	015070	7258#							

[illegible]

[illegible]

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 54
 DVKACA.SRC CROSS REFERENCE TABLE -- MACRO NAMES

COMMEN	9748																		
DUMP	51328	7445	7456																
ENDCOM	9868																		
ESCAPE	10978																		
MFPS	50708	7238	7244	7302	7322	7365	7383	7425											
MTPS	50858	7237	7243	7244	7251	7252	7286	7349	7410										
MULT	34668																		
NEWTST	10308																		
POP	14838	50568	7465																
PRINT	51428																		
PUSH	14758	50568	7465																
REPORT	44378																		
SDUMP	51378																		
SETUP	8018																		
SKIP	11318																		
SLASH	9268																		
STARS	8958	50558	5319	5330	5339	5340	7104	7258	7268	7445	7449	7465							
TYPBIN	14198																		
TYPDEC	13898																		
TYPNAM	11778																		
TYPNUM	13568																		
TYPOCS	13098																		
TYPOCT	12728																		
TYPTXT	12268																		
\$ADDER	56058	7163	7168																
\$ADDES	55078	7164	7169																
\$ADOR	54438	7147	7148	7149	7152	7153	7154	7155	7158										
\$ADOS	53838	7150	7151	7156	7157														
\$ADDER	61108	7243	7244																
\$ALL	68818	7237																	
\$ALL2	69148	7238																	
\$DIVER	67798	7223	7227	7231															
\$DIVES	66818	7232																	
\$DIVR	66178	7214	7216	7217															
\$DIVS	65538	7215																	
\$FATL	50598	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163				
	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193				
	7194	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216	7217	7219	7223				
	7227	7231	7232	7238	7243	7244	7251	7252	7442										
\$FINT	69728	7251	7252																
\$FIS7	60318	7159	7200	7202	7219														
\$HLT	51818	7445																	
\$HULER	64518	7206																	
\$HULES	63538	7210																	
\$HULR	62898	7193	7194	7195	7196	7198													
\$HULS	62338	7197																	
\$OCTAL	52268	7463																	
\$SCOPE	51508	7268																	
\$SUBER	59298	7189																	
\$SUBES	58318	7185																	
\$SUBR	57678	7173	7174	7176	7178	7179	7180	7181											
\$SUBS	57078	7175	7177																
\$SHDOC	50998	5283																	
\$SESCA	11108																		
\$SFATL	50668	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163				
	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193				

[illegible]

ADD BEQ	7277	7282	7445	7454	7486										
	7134	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163
	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193
	7194	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216	7217	7219	7223
BGT BIC BISB BIT BITB BNE BPL BR	7227	7231	7232	7238	7243	7244	7251	7252	7258	7268	7445	7463			
	7258														
	7238	7258	7306	7329	7369	7390	7429								
	7463														
CLR CLRB CMP	7268	7445													
	7133	7251	7252	7478											
	7114	7251	7252	7268	7445	7463	7465	7479	7483						
	7251	7252	7445	7459	7490										
CMPB	7150	7151	7156	7157	7163	7164	7168	7169	7175	7177	7185	7189	7197	7206	7210
	7215	7223	7227	7231	7232	7243	7244	7251	7252	7445	7463	7465	7492		
	7112	7139	7140	7237	7238	7243	7244	7246	7251	7252	7255	7286	7302	7322	7349
	7365	7383	7410	7425	7463	7465									
DEC DECB EXT FADD	7251	7252	7445	7463											
	7113	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163
	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193
	7194	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216	7217	7219	7223
FDIV FMUL FSUB HALT INC INCB IOT JMP JSR	7227	7231	7232	7238	7243	7244	7251	7252	7463						
	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163	7164
	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7185	7189	7200	7237	
	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216	7217	7219	7223	7227
MOV	7231	7232	7243	7244	7251	7252	7258	7445	7456	7463					
	7110	7111	7122	7123	7124	7125	7127	7128	7129	7130	7131	7132	7135	7136	7137
	7138	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7163	7164
	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193	7194
MOVB NOP	7195	7196	7197	7198	7206	7210	7214	7215	7216	7217	7223	7227	7231	7232	7237
	7238	7243	7244	7245	7251	7252	7253	7254	7258	7268	7278	7279	7280	7281	7281
	7291	7292	7307	7308	7309	7310	7326	7327	7328	7330	7332	7333	7334	7335	7336
	7341	7342	7343	7344	7345	7354	7355	7370	7371	7387	7388	7389	7391	7393	7394
	7395	7396	7402	7403	7404	7405	7406	7415	7416	7430	7432	7433	7434	7435	7436
	7445	7455	7456	7463	7465	7480	7481	7485							
	7251	7252	7268	7453	7463	7482	7491								
	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157	7158	7159	7163	7164
	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181	7185	7189	7193	7194
	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216	7217	7219	7223	7227
	7231	7232	7237	7243	7244	7258									

RESET	7258														
ROL	7463														
ROLB	7463														
RTI	7251	7252	7268	7331	7392	7443	7445	7465	7487						
RTS	7356	7372	7417	7437	7460	7463									
RTT	5378														
SUB	7431	7445													
TRAP	5307														
TST	7147	7148	7149	7150	7151	7154	7155	7156	7159	7163	7164	7174	7176	7193	7195
	7196	7198	7200	7202	7206	7216	7217	7219	7251	7252	7268	7276	7445	7484	
TSTB	7148	7159	7163	7174	7181	7214	7251	7252	7268	7445	7463	7489			
.ABS	5049														
.ASCII	7258														
.ASCII2	5362	5363	7465												
.BLKW	7463														
.BYTE	5339	5364	7258												
.ENABL	4														
.END	7493														
.ENDC	5057	5283	5319	5328	5330	5339	5340	7104	7147	7148	7149	7150	7151	7152	7153
	7154	7155	7156	7157	7158	7159	7163	7164	7168	7169	7173	7174	7175	7176	7177
	7178	7179	7180	7181	7185	7189	7193	7194	7195	7196	7197	7198	7200	7202	7206
	7210	7214	7215	7216	7217	7219	7223	7227	7231	7232	7238	7243	7244	7251	7252
	7258	7268	7445	7449	7463	7465									
.EVEN	5339	5365	7258	7465											
.IF	5057	5283	5319	5328	5330	5339	5340	7104	7147	7148	7149	7150	7151	7152	7153
	7154	7155	7156	7157	7158	7159	7163	7164	7168	7169	7173	7174	7175	7176	7177
	7178	7179	7180	7181	7185	7189	7193	7194	7195	7196	7197	7198	7200	7202	7206
	7210	7214	7215	7216	7217	7219	7223	7227	7231	7232	7238	7243	7244	7251	7252
	7258	7268	7445	7449	7463	7465									
.IFF	5319	5330	5339	5340	7104	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156
	7157	7158	7159	7163	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180
	7181	7185	7189	7193	7194	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215
	7216	7217	7219	7223	7227	7231	7232	7238	7243	7244	7251	7252	7258	7268	7445
	7449	7465													
.IFNE	7243	7244													
.IIF	5057	5339	7258	7463											
.IRP	5328	7465													
.LIST	2	4623	5053	5054	5281	5327	5328	5335	5339	7145	7147	7148	7149	7150	7151
	7152	7153	7154	7155	7156	7157	7158	7159	7162	7163	7164	7167	7168	7169	7172
	7173	7174	7175	7176	7177	7178	7179	7180	7181	7184	7185	7188	7189	7192	7193
	7194	7195	7196	7197	7198	7200	7202	7205	7206	7209	7210	7213	7214	7215	7216
	7217	7219	7222	7223	7226	7227	7230	7231	7232	7236	7237	7238	7241	7243	7244
	7249	7251	7252	7258	7263	7267	7271	7284	7285	7286	7289	7300	7301	7302	7305
	7320	7321	7322	7325	7347	7348	7349	7352	7363	7364	7365	7368	7381	7382	7383
	7386	7408	7409	7410	7413	7423	7424	7425	7428	7442	7463	7468			
.MACRO	39	81	166	308	485	515	586	747	801	895	926	974	986	1030	1064
	1097	1110	1131	1144	1177	1226	1272	1309	1356	1389	1419	1475	1483	1535	1739
	1947	2140	2228	2353	2450	2528	2613	2827	2919	2995	3095	3223	3286	3348	3466
	3504	3568	3666	3751	3790	3853	3891	3934	4032	4080	4350	4397	4437	4513	5059
	5066	5070	5085	5099	5132	5137	5142	5150	5181	5226	5383	5443	5507	5605	5707
	5767	5831	5929	6031	6110	6233	6289	6353	6451	6553	6617	6681	6779	6881	6914
	6972														
.MCALL	5055	5056													
.MEXIT	5339														
.NLIST	1	3	5050	5052	5279	5322	5328	5333	5339	7143	7147	7148	7149	7150	7151
	7152	7153	7154	7155	7156	7157	7158	7159	7160	7163	7164	7165	7168	7169	7170

DVKACA MACY11 27(732) 24-AUG-76 15:37 PAGE 55-2
 DVKACA.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SE0 0099

	7173	7174	7175	7176	7177	7178	7179	7180	7181	7182	7185	7186	7189	7190	7193
	7194	7195	7196	7197	7198	7200	7202	7203	7206	7207	7210	7211	7214	7215	7216
	7217	7219	7220	7223	7224	7227	7228	7231	7232	7234	7237	7238	7239	7243	7244
	7247	7251	7252	7258	7260	7265	7269	7283	7286	7287	7288	7299	7302	7303	7304
	7319	7322	7323	7324	7346	7349	7350	7351	7362	7365	7366	7367	7380	7383	7384
	7385	7407	7410	7411	7412	7422	7425	7426	7427	7442	7463	7466			
.NTYPE	7237	7238	7243	7244	7251	7252	7286	7302	7322	7349	7365	7383	7410	7425	
.PAGE	5045	5047	5336	6232	7103	7146	7147	7148	7149	7150	7151	7152	7153	7154	7155
	7156	7157	7158	7163	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180
	7181	7185	7189	7193	7194	7195	7196	7197	7198	7199	7201	7206	7210	7214	7215
	7216	7217	7218	7223	7227	7231	7232	7233	7242	7243	7244	7251	7252	7257	7268
	7272	7445	7447	7462	7464	7469									
.REPT	4624	5323													
.SBTTL	5280	5330	5334	5339	5340	7106	7144	7161	7166	7171	7183	7187	7191	7204	7208
	7212	7221	7225	7229	7235	7240	7248	7258	7268	7270	7445	7451	7463	7465	7467
.TITLE	5057														
.WORD	5330	5339	5340	5366	7147	7148	7149	7150	7151	7152	7153	7154	7155	7156	7157
	7158	7159	7163	7164	7168	7169	7173	7174	7175	7176	7177	7178	7179	7180	7181
	7185	7189	7193	7194	7195	7196	7197	7198	7200	7202	7206	7210	7214	7215	7216
	7217	7219	7223	7227	7231	7232	7237	7238	7243	7244	7251	7252	7258	7286	7302
	7322	7349	7365	7383	7410	7425	7465								

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

* DVKACA/CRF=DVKACA.SML DVKACA.SRC
 RUN-TIME: 44 56 5 SECONDS
 RUN-TIME RATIO: 568/106=5.3
 CORE USED: 41K (81 PAGES)

