

FP11

BASIC INSTRUCTION EXERCISER
MD-11-DCFPO-D

EP-DCFPO-D-DL

MAY 1978

COPYRIGHT '72-75

digital

FICHE 1 OF 1

MADE IN USA

[Microfiche Frame 1]	[Microfiche Frame 2]	[Microfiche Frame 3]	[Microfiche Frame 4]
[Microfiche Frame 5]	[Microfiche Frame 6]	[Microfiche Frame 7]	[Microfiche Frame 8]
[Microfiche Frame 9]	[Microfiche Frame 10]	[Microfiche Frame 11]	[Microfiche Frame 12]
[Microfiche Frame 13]	[Microfiche Frame 14]	[Microfiche Frame 15]	[Microfiche Frame 16]
[Microfiche Frame 17]	[Microfiche Frame 18]	[Microfiche Frame 19]	[Microfiche Frame 20]
[Microfiche Frame 21]	[Microfiche Frame 22]	[Microfiche Frame 23]	[Microfiche Frame 24]
[Microfiche Frame 25]	[Microfiche Frame 26]	[Microfiche Frame 27]	[Microfiche Frame 28]
[Microfiche Frame 29]	[Microfiche Frame 30]	[Microfiche Frame 31]	[Microfiche Frame 32]
[Microfiche Frame 33]	[Microfiche Frame 34]	[Microfiche Frame 35]	[Microfiche Frame 36]
[Microfiche Frame 37]	[Microfiche Frame 38]	[Microfiche Frame 39]	[Microfiche Frame 40]
[Microfiche Frame 41]	[Microfiche Frame 42]	[Microfiche Frame 43]	[Microfiche Frame 44]
[Microfiche Frame 45]	[Microfiche Frame 46]	[Microfiche Frame 47]	[Microfiche Frame 48]

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPD-D-D
PRODUCT NAME: FP11 BASIC INSTRUCTION EXERCISER
DATE CREATED: JANUARY 1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHORS: BOB BRAIN/STANLEY MARACKIEWICZ

COPYRIGHT: (C) 1972,1973,1975

DIGITAL EQUIPMENT CORPORATION

THIS MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

339		SWITCH OPTIONS AND ERROR BITS
392		MACRO'S AND ASSIGNMENTS
522		SETUP AND ANSWER AREA
566	T1	TEST OF WRITABILITY OF FPS
589	T2	TEST OF CFCC
615	T3	TEST OF LDD, STD, CMPD OF -1,0,-1,0
658	T4	TEST OF LDD, STD, CMPD OF 0,-1,0,-1
686	T5	TEST OF LDF, STF, CHPF OF -1,0
721	T6	TEST OF LDF, STF, CHPF WITH <=> IN ALL AC'S
772	T7	TEST OF CHPF WITH DATA IN AC0-AC5
788	T10	TEST OF TSTF AND TSTD USING OLD AC0-AC5
828	T11	TEST OF CLRX INSTRUCTIONS
867	T12	TEST OF NEGX
913	T13	TEST OF ABSX
956	T14	TEST OF LDEXP & STEXP
1011	T15	TEST OF ADDF & SUBF
1034	T16	TEST OF ADDD AND SUBD
1066	T17	TEST OF MULF AND DIVF
1088	T20	TEST OF MULD AND DIVD
1110	T21	TEST OF LDCFD,LCDF
1162	T22	TEST OF STCFD,STCDF
1193	T23	TEST OF LDCIF,LCID,STCFI,STCFI
1253	T24	TEST OF LDCLF,LDCLD,STCFL,STCDL
1311	T25	TEST OF MODF AND MODD
1361		ERROR CONDITIONS (STST)
1363	T26	LDD OF -0
1388	T27	MULF ERROR - OVERFLOW
1423	T30	DIVF ERROR - UNDERFLOW
1416	T31	STCFI ERROR - CONVERSION(6)
1428	T32	DIVF BY 0 ERROR
1439	T33	LDF -0 ERROR
1458	T34	OPCODE ERROR
1461	T35	UBREAK TRAP
1478	T36	ADDF ERROR - OVERFLOW
1491	T37	SUBF ERROR - UNDERFLOW
1562	T40	TEST FOR CONVERSION, ADDD AND SUBD
1574	T41	LDD AND STD TEST
1586	T42	MULD AND DIVD TEST
1624	T43	EXERCISER TEST
1645	T44	MODE ONE TEST
1667	T45	ADD EXERCISER
1684	T46	TEST DIVDE BY 0
1698	T47	EXERCISER FOR ADDD,SUBD,MULD AND DIVD
1804		BELL AND SCOPE ROUTINE
1874		HLT ROUTINE (ERROR TYPEOUT)
1924		OCTAL DUMP OF A WORD
1938		POWER DOWN AND UP ROUTINES
1998		TYPE ROUTINE

52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

MAINDEC-11-DCFPO-D
TABLE OF CONTENTS

FP11 BASIC INSTRUCTION EXERCISER

PAGE 2

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATOR ACTION
5.	OPERATING PROCEDURE
5.1	OPERATIONAL SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACT
6.	ERRORS
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
8.3	POWER FAIL
9.	PROGRAM DESCRIPTION

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

MAINDEC-11-DCFPO-D
DESCRIPTION

FP11 BASIC INSTRUCTION EXERCISER

PAGE 3

1. ABSTRACT

THIS PROGRAM TESTS THE FP11 IN ALL MODES WITH FIXED NUMBER PATTERNS. IT RUNS WITH INTERRUPTS BOTH ENABLED AND DISABLED AND CAUSES ERROR CONDITIONS. THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/45 STANDARD COMPUTER WITH FP11 OPTION

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) LOAD ADDRESS 200.
- 3) SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
- 4) PRESS START.
- 5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS

157
158
159
160

6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN,
7) THE DISPLAY ON THE 11/49 WILL SHOW THE ITERATION COUNT IN
THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE
DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

MAINDEC-11-DCFPO-D
DESCRIPTION

FP11 BASIC INSTRUCTION EXERCISER

PAGE 4

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 ,, ALL SWITCHES DOWN IS WORST CASE TESTING. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

SW<15> = 1 HALT ON ERROR
SW<14> = 1 SCOPE LOOP
SW<13> = 1 INHIBIT PRINTOUT
SW<12> = 1 INHIBIT TRACE TRAPPING
SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 BELL ON ERROR
 0 BELL ON PASS COMPLETE
SW<09> = 1 INHIBIT RELOCATION
SW<08> = 1 LOOP ON TEST IN SW<710>
 0 LOAD SW<710> INTO UB REGISTER

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRTRAP

IF SW<12> IS ON A 0, THE T BIT WILL BE SET ON ALTERNATE PASSES. WHEN SET, IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTT" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS

MAINCEC-11-DCFPO-D
DCFPCD,P11

FP11 EXERCISER STAND ALONE

MACY11 27(657) 16-JAN-75 10190 PAGE 9-1

216

REACHED.

218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271

MAINDEC-11-DCFPO-D
DESCRIPTION

FP11 BASIC INSTRUCTION EXERCISER

PAGE 5

5.2.4 TRAPCATCHER

A ",+2" = "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS, THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR = 2.

5.2.5 FLOATING POINT TRAP (TO 244)

WHEN THE FP11 INTERRUPT DISABLE BIT IS SET, NO TRAPS TO 244 SHOULD OCCUR. IF AN INTERRUPT OCCURS, THE PROGRAM WILL HALT AT 766 IN THE ROUTINE CALLED FLTERR AND DISPLAY THE FPS REGISTER IN R0. WHEN THE INTERRUPT DISABLE BIT IS CLEAR, THE PROGRAM WILL HANDLE THE TRAPS.

5.2.6 RELOCATOR

WITH SWC9) ON A ZERO, FLOATING POINT TEST (STARTING AT TEST 40) ARE RELOCATED AND EXECUTED THROUGHOUT MEMORY IN 1 K BLOCKS. WHEN AN ERROR IS ENCOUNTERED, THE ACTUAL ADDRESS IS PRINTED OUT. TO FIND THE LISTING ADDRESS, MASK OUT BITS 12 THROUGH 15 AND ADD 10000 TO THE RESULT.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR FPS ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
PEC FEA

WHERE:

ADR = ADDRESS OF ERROR HLT
FPS = FLOATING POINT STATUS OR BAD DATA IN "FPS"
PEC = FLOATING EXCEPTION CODES (ERROR CODES)
FEA = FLOATING EXCEPTION ADDRESS (ERROR ADDRESS)
ANS1-8 = ERROR DATA READ FROM THE FP11. FROM 0-8 OF THESE MAY BE TYPED DEPENDING ON THE NUMBER FOLLOWING THE HLT; I.E., HLT+3 WOULD TYPE ANS1-ANS3.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325

MAINDEC-11-DCFPO-D
DESCRIPTION

FP11 BASIC INSTRUCTION EXERCISER

PAGE 6

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 3MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 POWER FAIL

THIS TEST CAN BE POWER FAILED WITH NO ERRORS EXCEPT ON THE
FEC AND FEA. TO USE, START THE TEST AS USUAL AND POWER DOWN
THEN UP AT ANY TIME. THE PROGRAM SHOULD TYPE "POWER" AND
CONTINUE TO RUN WITH NO OTHER TYPEOUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TEST ALL THE INSTRUCTIONS ON THE FP11 AND ALL
THE MODES. IT IS DIVIDED INTO SUBTESTS (THE CODE BETWEEN 2
SCOPE STATEMENTS) WHICH ARE RUN 256 TIMES BEFORE CONTINUING
TO THE NEXT. SW<11> ON A 1 CAUSES EACH SUBTEST TO BE RUN
ONLY ONCE. WITH SW<9> ON A 1, THE PROGRAM WILL NOT
RELOCATE. THE ADDRESS ICNT (LOC 1000) AND DISPLAY REGISTER
ON THE 11/49 EACH CONTAIN THE ITERATION COUNT IN THE LEFT
BYTE AND THE TEST NUMBER IN THE RIGHT BYTE. ALL THE
SUBTESTS SHOULD BE RUN SEQUENTIALLY BY STARTING AT 200 NOT
BY STARTING AT THE BEGINNING OF THE SUBTEST. TO LOOP ON A
PARTICULAR SUBTEST, PUT THE TEST NUMBER (SEE LISTING) IN THE
RIGHT BYTE OF THE SWITCH REGISTER AND SW<8> ON A 1. THIS
TEST WILL BE LOOPED UPON UNTIL SW<8> IS PUT ON A 0 OR THE
RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE
PROGRAM WILL BE RUN AS USUAL.

.ENDR

.TITLE MAINDEC-11-DCFPO-D FP11 EXERCISER STAND ALONE
 ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
 ;PROGRAM BY BOB BRAIN AND STAN HARACKIEWICZ
 ;REMO

334
341
342
344
345
346
347
348
349
358
351
352
353
354
355
356
357
358
359
368
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384

SWITCH	USE
0	0 - LOAD UB REGISTER WITH SW<7:0>
1	1 - LOOP ON TEST IN SW<7:0>
9	RELOCATE * EXECUTE TESTS THROUGHOUT MEMORY
10	0 - BELL ON PASS COMPLETE
	1 - BELL ON ERROR
11	INHIBIT ITERATIONS
12	INHIBIT TRACE TRAP
13	INHIBIT ERROR TYPEOUTS
14	LOOP ON TEST
15	HALT ON ERROR

OUTPUT FORMI

ADR FPS ANS1 ANS2 ANS3 ANS4 ANS5 ANS6 ANS7 ANS8
 FEC FEA

BIT	FPS REASON	CODE	FEC ERROR
0	CARRY	0	ADDRESS ERROR
1	OVERFLOW	2	OPCODE ERROR
2	ZERO	4	DIVIDE BY ZERO
3	NEGATIVE	6	CONVERSION ERROR
4	MAINTAINANCE MODE	10	OVERFLOW
5	TRUNCATE MODE	12	UNDERFLOW
6	LONG INTEGER MODE	14	UNDEFINED VARIABLE (-0)
7	DOUBLE PRECISION MODE	16	UBREAK TRAP
8	INTERUPT ON CONVERSION ERROR		
9	INTERUPT ON OVERFLOW		
10	INTERUPT ON UNDERFLOW		
11	INTERUPT ON UNDEFINED VARIABLE		
12			
13			
14	INTERUPT DISABLE		
15	ERROR FLAG		

430				.ENABL	ABS	
432		000001		N=	1	
433		177776		PS=	177776	
435		177570		SHR=	177570	
439		040000		SH14=	40000	
440		020000		SH13=	20000	
441		010000		SH12=	10000	
442		004000		SH11=	4000	
443		002000		SH10=	2000	
444		001000		SH09=	1000	
445		000400		SH08=	400	
446		000000		FPS=	X0	
447		000000		RS=	X0	
448		000001		RETURN=	X1	
449		000001		R1=	X1	
450		000002		R2=	X2	
451		000003		R3=	X3	
452		000004		R4=	X4	
453		000005		R5=	X5	
454		000005		TTY=	X5	
455		000006		SP=	X6	
456		000007		PC=	X7	
457		000004		TYPE=	IOT	
458		000007		BELL=	7	
459		000000		AC0=	X0	
460		000001		AC1=	X1	
461		000002		AC2=	X2	
462		000003		AC3=	X3	
463		000004		AC4=	X4	
464		000005		AC5=	X5	
465		170003		LDUB=	170003	
466		170005		STAB=	170005	
467		170007		STOB=	170007	
468		170006		MRS=	170006	
469		170004		LDSC=	170004	
470		177776		PS=	177776	
472				.EQUIV	TRAP,SCOPE	
473				.EQUIV	ENT,HLT	
479						
482		000000		.=	0	;TRAP CATCHER FROM 0 - 776
(1)						
(1)		000200		.=	200	
(1)						
(1)	000200	000157	000632		JMP	START ;JUMP TO STARTING ADDRESS OF PROGRAM
489		000046		.=	46	
490	000046	011406			LOGICAL	
491		000052		.=	52	
492	000052	040000			40000	
494						
495		000760		.=	760	
496	000760	170200		FLTERRI	STFPS	FPS
497	000762	170367	000044		STST	FEC
498	000766	000000			HALT	
499	000770	000002			RTI	

```
526          001000          ,=      1000
511
512 001000 000000          ICNT: 0          ;ITERATION COUNT - LM TEST NO. - RM
513 001002 000000          ERRORS: 0          ;ERROR COUNT
514 001004 000000 000000 PCNT: 0,0          ;2 WORD PASS COUNT
515 001010 000000          LAD: 0          ;LOOP ADDRESS FOR SCOPE
516 001012 000000          ANS1: 0          ;FIRST ANSWER (SEE CODE)
517 001014 000000          ANS2: 0
518 001016 000000          ANS3: 0
519 001020 000000          ANS4: 0
520 001022 000000          ANS5: 0
521 001024 000000          ANS6: 0
522 001026 000000          ANS7: 0
523 001030 000000          ANS8: 0
524 001032 000000          FECI 0          ;FLOATING EXCEPTION CODES
525 001034 000000          FEAI 0          ;FLOATING EXECPTION ADDRESS
526
527          START:
530 001036 012706 000600          MOV      #600,SP          ;** STACK AT 600 **
531 001042 012737 000002 000016          MOV      #2,0010          ;SET RTI INTO 10
532 001050 012737 001072 000004          MOV      #M1120,004          ;FIND OUT WHICH MACHINE THIS IS
533 001056 005737 177772          TST      #0177772          ;IS PIRQ THERE?
534 001062 012737 000006 000016          MOV      #6,0010          ;FUDGE IN RTI IF 11/45
535 001070 000403          BR      BEGIN
536
537 001072 016737 011542 000010 M1120: MOV      FPTADR,0010          ;LOAD THE ILLEGAL INSTRUCTION VECTOR
538          ; WITH THE ADDRESS OF THE FPU,
539          ; THE FPU WILL HANDLE THE BAD OPCODES
540          ;RESET 4
540 001100 012737 000006 000004 BEGIN: MOV      #6,004
541 001106 012706 000600          MOV      #600,SP
542 001112 012777 012306 011504          MOV      #P0WDWN,0DWNVEC
543 001120 012777 000340 011500          MOV      #340,0DWNVEC+2
544 001126 012737 012506 000020          MOV      #10TS,0020          ;SET UP VECTOR 20
545 001134 012700 000030          MOV      #30,R0          ;SET R0 TO VECTOR 30
546 001140 012720 011560          MOV      #,TRP,(0)+          ;SET ENT VECTOR
547 001144 012720 000340          MOV      #340,(0)+
548 001150 012720 011420          MOV      #,ENT,(0)+          ;SET TRAP VECTOR
549 001154 012710 000340          MOV      #340,(0)
553 001160 005067 177614          CLR      ICNT
554 001164 005067 177620          CLR      LAD
555 001170 005001          CLR      RETURN
```

```

566          |.....
(2)          |TEST 1          TEST OF WRITABILITY OF FPS
(2)          |.....
(2) 001172 104400 TST11 SCOPE
567
568 001174 170127 147757 LDFPS 0147757 ;TEST FPS WITH ALL 1'S
569 001200 170200 STFPS FPS ;GET STATUS
570 001202 022700 147757 CMP 0147757,FPS ;CHECK IT
571 001206 001401 BEQ ,+4
572 001210 104000 HLT ;FPS NOT 147777
573
574 001212 170127 040000 LDFPS 040000 ;TEST FPS WITH 40000
575 001216 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001220 022700 040000 CMP 040000,FPS ;CHECK FLOATING POINT STATUS
(1) 001224 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001226 104000 HLT ;FPS NOT EQUAL TO 40000
(1)
576
577 001230 170011 SETD ;TEST OF DOUBLE BIT ON A 1
578 001232 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001234 022700 040200 CMP 040200,FPS ;CHECK FLOATING POINT STATUS
(1) 001240 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001242 104000 HLT ;FPS NOT EQUAL TO 40200
(1)
579
580 001244 170001 SETF ;TEST OF DOUBLE BIT ON A 0
581 001246 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001250 022700 040000 CMP 040000,FPS ;CHECK FLOATING POINT STATUS
(1) 001254 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001256 104000 HLT ;FPS NOT EQUAL TO 40000
(1)
582
583 001260 170012 SETL ;TEST OF LONG BIT ON A 1
584 001262 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001264 022700 040100 CMP 040100,FPS ;CHECK FLOATING POINT STATUS
(1) 001270 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001272 104000 HLT ;FPS NOT EQUAL TO 40100
(1)
585
586 001274 170002 SETI ;TEST OF LONG BIT ON A 0
587 001276 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001300 022700 040000 CMP 040000,FPS ;CHECK FLOATING POINT STATUS
(1) 001304 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001306 104000 HLT ;FPS NOT EQUAL TO 40000
(1)

```

589
(2)
(2)
(2) 021310 104400
592
591 021312 170127 040017
592 021316 170000
593 021320 013700 177776
594 021324 042700 177760
595 021330 022700 000017
596 021334 001401
597 021336 104000
598
599 021340 170127 040012
600 021344 170000
601 021346 013700 177776
602 021352 042700 177760
603 021356 022700 000012
604 021362 001401
605 021364 104000
606
607 021366 170127 040005
608 021372 170000
609 021374 013700 177776
610 021400 042700 177760
611 021404 022700 000005
612 021410 001401
613 021412 104000

```
.....  
;TEST 2 TEST OF CFCC  
.....  
TST21 SCOPE  
  
LDFPS #40017 ;LOAD ALL STATUS BITS TO 1'S  
CFCC ;GET THEM INTO PS  
MOV #0PS,FPS ;GET THEM FOR TYPING  
BIC #177760,FPS ;CLEAR JUNK  
CMP #17,FPS ;ALL SET?  
BEQ ,+4  
HLT ;PS NOT 17  
  
LDFPS #40012 ;LOAD FPS WITH 12  
CFCC ;GET INTO PS  
MOV #0PS,FPS ;GET FOR TYPING  
BIC #177760,FPS ;CLEAR JUNK  
CMP #12,FPS ;SAME AS LD?  
BEQ ,+4  
HLT ;PS NOT 12  
  
LDFPS #40005 ;LOAD FPS WITH 5  
CFCC ;GET BITS  
MOV #0PS,FPS ;SET FOR TYPING  
BIC #177760,FPS ;CLEAR JUNK  
CMP #5,FPS ;SAME?  
BEQ ,+4  
HLT ;PS NOT 5
```

```

615 ;.....
(2) ;TEST 3 TEST OF LOD, STD, CMPD OF -1,0,-1,0
(2) ;.....
(2) 001414 104400 TST3! SCOPE
616
617 001416 170127 047600 LD FPS #47600
618 001422 012700 011102 MOV #D1010,FPS ;GET ADDRESS OF DATA WORD
619 001426 172420 LD (FPS)+,ACB ;LOAD INTO ACB
620 001430 022700 011112 CMP #D1010+10,FPS ;IS THE NEW ADDRESS RIGHT?
621 001434 001401 BEQ ,+4
622 001436 104000 HLT ;FPS NOT D1010+10
623
624 001440 010667 011146 MOV SP,,TYPE
625 001444 162767 000010 011140 SUB #10,,TYPE
626 001452 174046 STD ACB,-(SP) ;GET THE DATA BACK
627 001454 010600 MOV SP,FPS ;SAVE THE SP FOR TYPING
628 001456 026706 011130 CMP ,TYPE,SP ;SP DECREMENTED PROPERLY?
629 001462 001401 BEQ ,+4
630 001464 104000 HLT ;SP NOT SP-10
631
632 001466 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 001470 022700 047610 CMP #47610,FPS ;CHECK FLOATING POINT STATUS
(1) 001474 001401 BEQ ,+4 ;BRANCH IF OK
(1) 001476 104000 HLT ;FPS NOT EQUAL TO 47610
633
634 001500 021667 007376 CMP (SP),D1010 ;CHECK FIRST POCCE OF DATA
635 001504 001401 BEQ ,+4
636 001506 104204 HLT 204 ;DATA IN (SP) NOT D1010
637 001510 026667 000002 007366 CMP 2(SP),D1010+2 ;CHECK SECOND
638 001516 001401 BEQ ,+4
639 001520 104204 HLT 204 ;DATA IN 2(SP) NOT D1010+2
640 001522 026667 000004 007356 CMP 4(SP),D1010+4 ;CHECK THIRD
641 001530 001401 BEQ ,+4
642 001532 104204 HLT 204 ;DATA IN 4(SP) NOT D1010+4
643 001534 026667 000006 007346 CMP 6(SP),D1010+6 ;CHECK FOURTH
644 001542 001401 BEQ ,+4
645 001544 104204 HLT 204 ;DATA IN 6(SP) NOT D1010+6
646
647 001546 062767 000010 011036 ADD #10,,TYPE
648 001554 173426 CMPD (SP)+,ACB ;RECHECK DATA AND SP
649 001556 010600 MOV SP,FPS ;SAVE SP FOR TYPING
650 001560 026706 011026 CMP ,TYPE,SP ;CHECK ADDRESS IN SP
651 001564 001401 BEQ ,+4
652 001566 104000 HLT ;SP NOT RESTORED
653 001570 170200 STFPS FPS ;GET FPS IN CASE CMPD FAILED
654 001572 170000 CFCC ;NOW GET THE FP CONDITION CODES
655 001574 001401 BEQ ,+4 ;IF IT HALTS HERE IT MUST BE THE
656 001576 104000 HLT ;CMPD BECAUSE THE IS ALREADY CONFIRMED

```



```

658      |.....|
(2)      |TEST 4      TEST OF LOD, STD, CMPD OF 0,-1,0,-1
(2)      |.....|
(2) 001600 104400      TST41  SCOPE
659
660 001602 170127 047600      LD FPS      #47600
661 001606 172437 011104      LD          0000101,ACB      ;LOAD 0,-1,0,-1 INTO ACB *PIC*
662 001612 170200      ST FPS      FPS              ;STORE FLOATING POINT STATUS
(1) 001614 022700 047604      CMP          #47604,FPS      ;CHECK FLOATING POINT STATUS
(1) 001620 001401      BEQ         ,+4              ;BRANCH IF OK
(1) 001622 104000      HLT
663
664 001624 012700 001012      MOV          #ANS1,FPS      ;ADDRESS TO BE STORED INTO
665 001630 174010      STD          ACB,(FPS)      ;STORE IT INTO ANS1-4 *PIC*
666 001632 026767 007246 177152      CMP          00101,ANS1     ;FIRST WORD OK?
667 001640 001401      BEQ         ,+4
668 001642 104004      HLT 4              ;ANS1 NOT 00101
669 001644 026767 007236 177142      CMP          00101+2,ANS2    ;SECOND
670 001652 001401      BEQ         ,+4
671 001654 104004      HLT 4              ;ANS2 NOT 00101+2
672 001656 026767 007226 177132      CMP          00101+4,ANS3    ;THIRD
673 001664 001401      BEQ         ,+4
674 001666 104004      HLT 4              ;ANS3 NOT 00101+4
675 001670 026767 007216 177122      CMP          00101+6,ANS4    ;FOURTH
676 001676 001401      BEQ         ,+4
677 001700 104004      HLT 4              ;ANS4 NOT 00101+6
678
679 001702 012704 011126      MOV          #A00101-2,R4   ;ADDRESS-2 OF DATA
680 001706 173474 000002      CMPD        02(4),ACB      ;CHECK DATA IN ACB *PIC*
681 001712 170200      ST FPS      FPS              ;GET STAU FOR TYPING
682 001714 170000      CFCC
683 001716 001401      BEQ         ,+4              ;GET CONDITION CODES
684 001720 104000      HLT
;CMPD FAILED

```

```

686 .....
(2) ;TEST 5 TEST OF LDF, STF, CMPF OF -1,0
(2) .....
(2) TST51 SCOPE
687
688 001722 104400 LDFPS #47400 ;SET FLOATING MODE
689 001724 170127 047400 LDF D1010,ACB ;LOAD -1,0 INTO ACB
690 001730 172467 007146 MOV #ANS1,FPS ;POINTED TO ANSWER AREA *PIC*
691 001734 012700 001012 STF ACB,(FPS)+ ;STORE RESULT
692 001740 174020 001016 CMP #ANS3,FPS ;INCREMENTED PROPERLY
693 001742 022700 BEQ ,+4
694 001746 001401 HLT ;FPS NOT ANS1+4
695
696 001750 104000
697 001752 026767 007124 177032 CMP D1010,ANS1 ;CHECK FIRST WORD
698 001760 001401 BEQ ,+4
699 001762 104002 HLT 2 ;ANS1 NOT D1010
700 001764 026767 007114 177022 CMP D1010+2,ANS2 ;SECOND
701 001772 001401 BEQ ,+4
702 001774 104002 HLT 2 ;ANS2 NOT D1010+2
703
704 001776 170011 SETD ;GO TO DOUBLE MODE
705 002000 012700 001022 MOV #ANS5,FPS ;ADDRESS OF DATA+10
706 002004 174040 STD ACB,-(FPS) ;GET DATA
707 002006 022700 001012 CMP #ANS1,FPS ;CHECK FOR PROPER DECREMENTATION
708 002012 001401 BEQ ,+4
709 002014 104000 HLT ;FPS NOT ANS1
710
711 002016 012700 011134 MOV #AD1000,FPS ;LOAD ADDRESS OF ADDRESS OF DATA
712 002022 173430 CMPD 0(FPS)+,ACB ;CHECK THE DATA
713 002024 022700 011136 CMP #AD1000+2,FPS ;FPS GETS INCREMENTED BY 2
714 002030 001401 BEQ ,+4
715 002032 104000 HLT ;FPS NOT AD1001+2
716
717 002034 170200 STFPS FPS ;GET FPS
718 002036 170000 CFCC ;COPY CONDITION CODES
719 002040 001401 BEQ ,+4 ;EITHER CMPD FAILED OR THE
720 002042 104004 HLT 4 ;LDF MODIFIED RIGHT HALF

```

```

721          ;.....
(2)          ;TEST 6          TEST OF LDF, STF, CMPF WITH <=> IN ALL AC'S
(2)          ;.....
(2) 002044 104400          TST61  SCOPE
722
723 002046 170127 047400          LDFPS  #47400          ;LOAD FLOATING MODE
724 002052 172427 140640          LDF    #=5,AC0          ;LOAD AC0 WITH -5
725 002056 172527 140200          LDF    #=1,AC1          ;LOAD AC1 WITH -1
726 002062 172627 000000          LDF    #0,AC2           ;LOAD AC2 WITH 0
727 002066 172727 040640          LDF    #5,AC3           ;LOAD AC3 WITH 5
728 002072 174004          STF    AC0,AC4          ;LOAD AC4 WITH -5
729 002074 174305          STF    AC3,AC5          ;LOAD AC5 WITH 5
730
731 002076 174067 176710          STF    AC0,ANS1         ;GET AC0
732 002102 173427 140640          CMPF   #=5,AC0          ;CHECK IT
733 002106 170200          STFPS  FPS              ;GET STATUS
734 002110 170000          CFCC                   ;GET CC
735 002112 001401          BEQ    ,+4              ;AC0 NOT -5
736 002114 104002          HLT    2
737
738 002116 174137 001012          STF    AC1,00ANS1       ;GET AC1
739 002122 173567 176664          CMPF   ANS1,AC1         ;CHECK IT
740 002126 170200          STFPS  FPS              ;GET STATUS
741 002130 170000          CFCC                   ;GET CC
742 002132 001401          BEQ    ,+4              ;AC1 NOT -1
743 002134 104002          HLT    2
744 002136 012704 001012          ADANS1; MOV  #ANS1,R4     ;POINTER TO ANSWER AREA
745 002142 174214          STF    AC2,(4)          ;PUT DATA INTO ANS1
746 002144 173667 176642          CMPF   ANS1,AC2         ;CHECK DATA
747 002150 170200          STFPS  FPS              ;SET STATUS
748 002152 170000          CFCC                   ;GET CC
749 002154 001401          BEQ    ,+4              ;AC2 NOT 0
750 002156 104002          HLT    2
751
752 002160 174377 177754          STF    AC3,0ADANS1+2    ;PUT DATA INTO ANS1
753 002164 173767 176622          CMPF   ANS1,AC3         ;CHECK DATA
754 002170 170200          STFPS  FPS              ;GET STATUS
755 002172 170000          CFCC                   ;GET CC
756 002174 001401          BEQ    ,+4              ;AC3 NOT 5
757 002176 104002          HLT    2
758
759 002200 173404          CMPF   AC4,AC0          ;CHECK AC4 FOR -5
760 002202 170200          STFPS  FPS              ;GET STATUS
761 002204 170000          CFCC                   ;GET CC
762 002206 001401          BEQ    ,+4
763 002210 104000          HLT
764
765
766 002212 173705          CMPF   AC5,AC3         ;CHECK AC5 FOR 5
767 002214 170200          STFPS  FPS              ;GET STATUS
768 002216 170000          CFCC                   ;GET CC
769 002220 001401          BEQ    ,+4
770 002222 104000          HLT

```

```
772 |.....  
(2) |TEST 7 TEST OF CMPF WITH DATA IN AC0-AC5  
(2) |.....  
(2) 002224 104000 TST71 SCOPE  
773  
774 002226 170127 041000 LDFPS #41000 ;LOAD STATUS WITH 0  
775 002232 173405 CMPF AC5,AC0 ;CMP 5 TO -5  
776 002234 170200 STFPS FPS ;STORE FLOATING POINT STATUS  
(1) 002236 022700 041000 CMP #41000,FPS ;CHECK FLOATING POINT STATUS  
(1) 002242 001401 BEQ ,+4 ;BRANCH IF OK  
(1) 002244 104000 HLT ;FPS NOT EQUAL TO 41000  
(1)  
777  
778 002246 173704 CMPF AC4,AC3 ;CMP -5 TO 5  
779 002250 170200 STFPS FPS ;STORE FLOATING POINT STATUS  
(1) 002252 022700 041010 CMP #41010,FPS ;CHECK FLOATING POINT STATUS  
(1) 002256 001401 BEQ ,+4 ;BRANCH IF OK  
(1) 002260 104000 HLT ;FPS NOT EQUAL TO 41010  
(1)  
780  
781 002262 173767 006650 CMPF 0B1G,AC3 ;MAKE IT OVERFLOW  
782 002266 170200 STFPS FPS ;STORE FLOATING POINT STATUS  
783 002270 022700 041000 CMP #41000,FPS ;CHECK FLOATING POINT STATUS  
784 002274 001401 BEQ ,+4  
785 002276 104000 HLT ;FPS NOT 141002  
786
```

```

788
(2)
(2)
(2) 002300 104400
789
790 002302 170127 040000 LDFPS #40000
791 002306 170501 TSTF AC1 ;TEST AC1 = -1
792 002310 170200 STFPS FPS ;GET THE STATUS
793 002312 022700 040010 CMP #40010,FPS ;CHECK STATUS
794 002316 001401 BEQ ,+4
795 002320 104000 HLT ;N BIT NOT SET
796
797 002322 170505 TSTF AC5 ;TEST AC5 = 5
798 002324 170200 STFPS FPS ;GET STATUS
799 002326 022700 040000 CMP #40000,FPS ;CHECK STATUS
800 002332 001401 BEQ ,+4
801 002334 104000 HLT ;NOT 0
802
803 002336 170502 TSTF AC2 ;TEST AC2 = 0
804 002340 170200 STFPS FPS ;GET STATUS
805 002342 022700 040004 CMP #40004,FPS ;CHECK STATUS
806 002346 001401 BEQ ,+4
807 002350 104000 HLT ;Z BIT NOT SET
808
809 002352 170527 177777 TSTF #-1 ;TEST FOR THE N BIT IN LINE
810 002356 000401 BR ,+4 ;SHOULD GO HERE
811 002360 104000 HLT ;INCREMENTED BY 4 NOT 2
812 002362 170200 STFPS FPS ;GET STATUS
813 002364 022700 040010 CMP #40010,FPS ;CHECK THE N BIT
814 002370 001401 BEQ ,+4
815 002372 104000 HLT ;N BIT NOT SET
816
817 002374 170011 SETD ;SET DOUBLE MGGE
818 002376 170527 000000 TSTD #0 ;TEST FOR Z BIT IN LINE
819 002402 000403 BR ,+10 ;SHOULD GO HERE
820 002404 000000 HALT ;NOT HERE
821 002406 000000 HALT ;OR HERE
822 002410 000000 HALT ;OR HERE
823 002412 170200 STFPS FPS ;GET STATUS
824 002414 022700 040204 CMP #40204,FPS ;CHECK STATUS
825 002420 001401 BEQ ,+4
826 002422 104000 HLT ;Z BIT NOT SET

```

```

028 (2) .....
(2) ITEST 11 TEST OF CLRX INSTRUCTIONS
(2) .....
(2) 002424 104400 TST11 SCOPE
029
030 002426 170127 040200 LDFPS 040200 ;DOUBLE MODE
031 002432 172467 006444 LOD D1010,AC0 ;LOAD A -1
032 002436 174067 176350 STD AC0,ANS1 ;PUT INTO ANS1
033 002442 170001 SETF ;SET FLOATING
034 002444 170467 176342 CLRF ANS1 ;CLEAR IT OUT
035 002450 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 002452 022700 040004 CMP 040004,FPS ;CHECK FLOATING POINT STATUS
(1) 002456 001401 BEQ ,04 ;BRANCH IF OK
(1) 002460 104000 HLT ;FPS NOT EQUAL TO 40004
(1)
036
037 002462 170567 176324 TSTF ANS1 ;TEST FOR ZERO
038 002466 170000 CFCC ;GET CC
039 002470 001401 BEQ ,04
040 002472 104002 HLT 2 ;AC0 NOT ZERO
041
042 002474 026767 006406 176314 CMP D1010*4,ANS3 ;CHECK THIRD WORD
043 002502 001401 BEQ ,04
044 002504 104004 HLT 4 ;ANS3 NOT -1
045 002506 026767 006376 176304 CMP D1010*6,ANS4 ;CHECK FOURTH
046 002514 001401 BEQ ,04
047 002516 104004 HLT 4 ;ANS4 NOT 0
048
049 002520 170011 SETD
050 002522 170400 CLRD AC0 ;CLEAR THE REST OF AC0
051 002524 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 002526 022700 040204 CMP 040204,FPS ;CHECK FLOATING POINT STATUS
(1) 002532 001401 BEQ ,04 ;BRANCH IF OK
(1) 002534 104000 HLT ;FPS NOT EQUAL TO 40204
(1)
052 002536 174067 176250 STD AC0,ANS1 ;STORE RESULT
053 002542 170500 TSTD AC0 ;DID IT CLEAR
054 002544 170000 CFCC ;GET STATUS
055 002546 001401 BEQ ,04
056 002550 104004 HLT 4 ;DID NOT CLEAR
057
058 002552 172467 006370 LOD DMZERO,AC0 ;LOAD A MINUS ZERO
059 002556 170400 CLRD AC0 ;CLEAR IT OUT
060 002560 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 002562 022700 040204 CMP 040204,FPS ;CHECK FLOATING POINT STATUS
(1) 002566 001401 BEQ ,04 ;BRANCH IF OK
(1) 002570 104000 HLT ;FPS NOT EQUAL TO 40204
(1)
061 002572 174067 176214 STD AC0,ANS1 ;STORE RESULT
062 002576 170500 TSTD AC0 ;CHECK IT
063 002600 170000 CFCC ;GET CC
064 002602 001401 BEQ ,04
065 002604 104004 HLT 4 ;DID NOT CLEAR

```

```

.....
TEST 12      TEST OF NEGX
.....
TST12) SCOPE

867          (2)
868          (2)
869          (2) 002606 104400
870          002610 170127 040000      LDPPS      040000
871          002614 172427 140640      LDF        0-5,ACB      ;LOAD ACB WITH -5
872          002620 170700      NEGF       ACB          ;MAKE IT 5
873          002622 170200      STPPS     FPS          ;STORE FLOATING POINT STATUS
874          (1) 002624 022700 040000      CMP       040000,FPS    ;CHECK FLOATING POINT STATUS
875          (1) 002630 001401      BEQ       ,+4          ;BRANCH IF OK
876          (1) 002632 104000      HLT
877          002634 174067 176152      STP       ACB,ANS1     ;GET THE RESULT
878          002640 173427 040640      CMPI     05,ACB        ;CHECK THE RESULT
879          002644 170000      CFCC
880          002646 001401      BEQ       ,+4          ;GET CC
881          002650 104002      HLT          ;RESULT NOT 5
882          002652 170767 176134      NEGF     ANS1
883          002656 170200      STPPS     FPS          ;MAKE IT -5
884          (1) 002660 022700 040010      CMP       040010,FPS    ;STORE FLOATING POINT STATUS
885          (1) 002664 001401      BEQ       ,+4          ;CHECK FLOATING POINT STATUS
886          (1) 002666 104000      HLT          ;BRANCH IF OK
887          002670 022707 140640 176114      CMP       0140640,ANS1  ;CHECK THE RESULT
888          002676 001401      BEQ       ,+4          ;RESULT NOT -5
889          002700 104002      HLT          ;RESULT NOT -5
890          002702 005767 176106      TST      ANS2
891          002706 001401      BEQ       ,+4          ;REST 07
892          002710 104002      HLT          ;SKIP IF OK
893          002712 170127 047400      LDPPS     047400
894          002716 170400      CLR      ACB          ;TURN ON INTERRUPTS
895          002720 170700      NEGF     ACB          ;CLEAR ACB
896          002722 170200      STPPS     FPS          ;NEGATE IT
897          (1) 002724 022700 047404      CMP       047404,FPS    ;STORE FLOATING POINT STATUS
898          (1) 002730 001401      BEQ       ,+4          ;CHECK FLOATING POINT STATUS
899          (1) 002732 104000      HLT          ;BRANCH IF OK
900          002734 174067 176052      STP       ACB,ANS1     ;TURN ON INTERRUPTS
901          002740 170500      TSTF     ACB          ;CLEAR ACB
902          002742 170000      CFCC     ;NEGATE IT
903          002744 001401      BEQ       ,+4          ;GET RESULT
904          002746 104002      HLT          ;CHECK IT
905          002750 170011      SETD
906          002752 170400      CLRD     ACB          ;GET CC
907          002754 170700      NEGD     ACB          ;CHECK IT
908          002756 170200      STPPS     FPS          ;GET CC
909          ;RESULT NOT 0
910          ;SET DOUBLE MODE
911          ;CLEAR ACB
912          ;NEGATE ACB
913          ;STORE FLOATING POINT STATUS

```

(1)	002760	022700	047604						
(1)	002764	001401							
(1)	002766	104000							
(1)									
926									
927	002770	174067	176016						
928	002774	170500							
929	002776	170000							
910	003000	001401							
911	003002	104004							
912									
913									
(2)									
(2)									
(2)	003004	104400							
914									
915	003006	170127	040000						
916	003012	172427	140200						
917	003016	174067	175770						
918	003022	170607	175764						
919	003026	170200							
(1)	003030	022700	040000						
(1)	003034	001401							
(1)	003036	104000							
(1)									
920									
921	003040	022767	040200	175744					
922	003046	001401							
923	003050	104002							
924									
925	003052	009767	175736						
926	003056	001401							
927	003060	104002							
928									
929	003062	170600							
930	003064	170200							
931	003066	173427	040200						
932	003072	170000							
933	003074	001401							
934	003076	104002							
935									
936	003100	172467	006042						
937	003104	170600							
938	003106	170200							
(1)	003110	022700	040004						
(1)	003114	001401							
(1)	003116	104000							
(1)									
939									
940	003120	174067	175666						
941	003124	170500							
942	003126	170000							
943	003130	001401							
944	003132	104002							

.....
 ;TEST 13 TEST OF ABSX
 ;.....
 TST131 SCOPE

945							
946	003134	170011		SETD			;SET DOUBLE MODE
947	003136	172467	009740	LDD	D1010,AC0		;LOAD -1,0,-1,0
948	003142	170600		ABSD	AC0		;ABS IT
949	003144	170200		STFPS	FPS		;STORE FLOATING POINT STATUS
(1)	003146	022700	040200	CMP	#40200,FPS		;CHECK FLOATING POINT STATUS
(1)	003152	001401		BEO	,+4		;BRANCH IF OK
(1)	003154	104000		HLT			;FPS NOT EQUAL TO 40200
(1)							
950	003156	174067	175630	STD	AC0,ANS1		;GET RESULT
951	003162	173467	009750	CMPO	DBIG,AC0		;CHECK THE RESULT
952	003166	170000		CFCC			;GET CC
953	003170	001401		BEO	,+4		
954	003172	104004		HLT	4		;RESULT NOT 77777,0,-1,0

.....
;TEST 14 TEST OF LDEXP & STEXP
;.....
TST14) SCOPE

955							
956							
(2)							
(2)							
(2)	003174	104400					
957							
958	003176	170127	040200	LDFPS	#40200		;SET DOUBLE MODE
959	003202	172467	009674	LDD	D1010,AC0		;LOAD A -1,0,-1,0
960	003206	176427	177600	LDEXP	#-200,AC0		;CLEAR THE EXPONENT
961	003212	170200		STFPS	FPS		;STORE FLOATING POINT STATUS
(1)	003214	022700	040204	CMP	#40204,FPS		;CHECK FLOATING POINT STATUS
(1)	003220	001401		BEO	,+4		;BRANCH IF OK
(1)	003222	104000		HLT			;FPS NOT EQUAL TO 40204
(1)							
962							
963	003224	174067	175562	STD	AC0,ANS1		;GET THE RESULT
964	003230	170500		TSTD	AC0		;IS IT 0
965	003232	170000		CFCC			
966	003234	001401		BEO	,+4		
967	003236	104004		HLT	4		;AC0 NOT 0
968							
969	003240	175067	175546	STEXP	AC0,ANS1		;GET THE RESULT
970	003244	013700	177776	MOV	00PS,FPS		;GET PS BITS
971	003250	042700	177760	BIC	#177760,FPS		;CLEAR JUNK
972	003254	022700	000010	CMP	#10,FPS		;IS IT OK?
973	003260	001401		BEO	,+4		;SKIP IF OK
974	003262	104000		HLT			;PS IS WRONG
975							
976	003264	022707	177600 175520	CMP	#-200,ANS1		;CHECK IT
977	003272	001401		BEO	,+4		
978	003274	104001		HLT	1		;EXPONENT NOT 0
979							
980	003276	170001		SETF			;SET FLOATING MODE
981	003300	172467	009600	LDF	DB101,AC0		;LOAD A 0,-1
982	003304	176427	000200	LDEXP	#200,AC0		;SET EXPONENT TO -1
983	003310	170200		STFPS	FPS		;STORE FLOATING POINT STATUS
(1)	003312	022700	040006	CMP	#40006,FPS		;CHECK FLOATING POINT STATUS
(1)	003316	001401		BEO	,+4		;BRANCH IF OK
(1)	003320	104000		HLT			;FPS NOT EQUAL TO 40006

984								
985	003322	174067	175464	STF	AC0,ANS1			;SAVE RESULT
986	003326	005767	175460	TST	ANS1			;CHECK FIRST WORD
987	003332	001401		BEO	,04			
988	003334	104002		HLT	2			;ANS1 NOT 0
989								
990	003336	022767	177777	175450	CHP	#-1,ANS2		;CHECK SECOND WORD
991	003344	001401		BEO	,04			
992	003346	104002		HLT	2			;ANS2 NOT -1
993								
994	003350	175067	175436	STEXP	AC0,ANS1			;GET THE EXPONENT BACK
995	003354	022767	177600	175430	CHP	#-200,ANS1		;CHECK IT
996	003362	001401		BEO	,04			
997	003364	104000		HLT				;EXPONENT NOT -200
998								
999	003366	176527	000052	LDEXP	052,AC1			;LOAD ALT 1'S
1000	003372	175100		STEXP	AC1,FPS			;GET THEM BACK
1001	003374	022700	000052	CHP	052,FPS			;OK?
1002	003400	001401		BEO	,04			
1003	003402	104000		HLT				;EXP NOT 252
1004								
1005	003404	176627	000025	LDEXP	025,AC2			;LOAD OTHER ALT 1'S
1006	003410	175200		STEXP	AC2,FPS			;GET IT BACK
1007	003412	022700	000025	CHP	025,FPS			;CHECK IT
1008	003416	001401		BEO	,04			
1009	003420	104000		HLT				;EXP NOT 125
1010								
1011								
(2)								
(2)								
(2)	003422	104400						
1012								
1013	003424	170127	047400	LDPPS	047400			;LOAD FLOATING MODE
1014	003430	172427	040600	LDF	04,AC0			;LOAD A 4
1015	003434	172027	040400	ADDF	02,AC0			;ADD A 2
1016	003440	170200		STFPS	FPS			;STORE FLOATING POINT STATUS
(1)	003442	022700	047400	CHP	047400,FPS			;CHECK FLOATING POINT STATUS
(1)	003446	001401		BEO	,04			;BRANCH IF OK
(1)	003450	104000		HLT				;FPS NOT EQUAL TO 47400
(1)								
1017								
1018	003452	174067	175334	STF	AC0,ANS1			;STORE RESULT
1019	003456	173427	040700	CHPF	06,AC0			;CHECK RESULT
1020	003462	170000		CFCC				;GET CC
1021	003464	001401		BEO	,04			
1022	003466	104002		HLT	2			;RESULT NOT 6
1023								
1024	003470	173027	041020	SUBF	09,AC0			;SUBTRACT 9 FROM 6
1025	003474	170200		STFPS	FPS			;STORE FLOATING POINT STATUS
(1)	003476	022700	047410	CHP	047410,FPS			;CHECK FLOATING POINT STATUS
(1)	003502	001401		BEO	,04			;BRANCH IF OK
(1)	003504	104000		HLT				;FPS NOT EQUAL TO 47410
(1)								
1026								

.....
;TEST 15 TEST OF ADDF & SUBF
;.....
TST15: SCOPE

1027	003506	174067	179300	STF	ACB,ANS1	;STORE RESULT IN ANS1
1028	003512	173427	140500	CHPF	0=3,ACB	;CHECK RESULT
1029	003516	170000		CFCC		;GET CC
1030	003520	001401		BEO	,04	
1031	003522	104002		HLT	2	;0 = 9 NOT -37

.....
;TEST 16 TEST OF ADDD AND SUBD
;.....
TST16: SCOPE

1035	003524	104400				
1036	003526	170127	047600	LDFPS	047600	;SET DOUBLE MODE
1037	003532	172527	141400	LDD	0=32,,AC1	;LOAD A -32.
1038	003536	172127	040640	ADD	05,AC1	;ADD 9 TO -32.
1039	003542	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	003544	022700	047610	CHP	047610,FPS	;CHECK FLOATING POINT STATUS
(1)	003550	001401		BEO	,04	;BRANCH IF OK
(1)	003552	104000		HLT		;FPS NOT EQUAL TO 47610

1040						
1041	003554	174177	005376	STD	AC1,0AANS1	;GET RESULT
1042	003560	173527	141330	CHPD	0=27,,AC1	;CHECK RESULT
1043	003564	170000		CFCC		;GET CC
1044	003566	001401		BEO	,04	
1045	003570	104004		HLT	4	;=32,05 NOT -27

1046						
1047	003572	172667	005362	LDD	DALTA,AC2	;LOAD ALT 1'S
1048	003576	172267	005366	ADD	DALTB,AC2	;ADD OTHER 1'S
1049	003602	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	003604	022700	047600	CHP	047600,FPS	;CHECK FLOATING POINT STATUS
(1)	003610	001401		BEO	,04	;BRANCH IF OK
(1)	003612	104000		HLT		;FPS NOT EQUAL TO 47600

1050						
1051	003614	174277	005336	STD	AC2,0AANS1	;GET RESULT
1052	003620	173627	040500	CHPD	03,AC2	;CHECK RESULT
1053	003624	170000		CFCC		;GET CC
1054	003626	001401		BEO	,04	
1055	003630	104004		HLT	4	;ANS1 NOT DALTA

1056						
1057	003632	173267	005322	SUBD	DALTA,AC2	;SUBTRACT IT BACK
1058	003636	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	003640	022700	047600	CHP	047600,FPS	;CHECK FLOATING POINT STATUS
(1)	003644	001401		BEO	,04	;BRANCH IF OK
(1)	003646	104000		HLT		;FPS NOT EQUAL TO 47600

1059						
1060	003650	174267	175136	STD	AC2,ANS1	;GET THE RESULT
1061	003654	173667	005320	CHPD	DALTC,AC2	;CHECK IT
1062	003660	170000		CFCC		;GET CC
1063	003662	001401		BEO	,04	
1064	003664	104004		HLT	4	;ANS1 NOT DALTA

```
1066 .....  
(2) ;TEST 17 TEST OF MULF AND DIVF  
(2) ;.....  
(2) 023666 104400 TST17: SCOPE  
1067  
1068 003670 170127 047400 LDFPS #47400 ;LOAD FLOATING MODE  
1069 003674 172727 040740 LDF #7,AC3 ;LOAD A7  
1070 003700 171327 140500 MULF #-3,AC3 ;X =3  
1071 003704 170200 STFPS FPS ;STORE FLOATING POINT STATUS  
(1) 003706 022700 047410 CMP #47410,FPS ;CHECK FLOATING POINT STATUS  
(1) 023712 001401 BEQ ,04 ;BRANCH IF OK  
(1) 023714 104000 HLT ;FPS NOT EQUAL TO 47410  
(1)  
1072  
1073 003716 174367 175070 STP AC3,ANS1 ;GET RESULT  
1074 003722 173727 141250 CMPF #-21,,AC3 ;CHECK RESULT  
1075 003726 170000 CFCC ;GET CC  
1076 003730 001401 BEQ ,04  
1077 003732 104002 HLT 2 ;X =3 NOT =21.  
1078  
1079 003734 174727 140740 DIVF #-7,AC3 ;DIVIDE BY -3  
1080 003740 170200 STFPS FPS ;STORE FLOATING POINT STATUS  
(1) 003742 022700 047400 CMP #47400,FPS ;CHECK FLOATING POINT STATUS  
(1) 003746 001401 BEQ ,04 ;BRANCH IF OK  
(1) 003750 104000 HLT ;FPS NOT EQUAL TO 47400  
(1)  
1081  
1082 003752 174367 175034 STP AC3,ANS1 ;GET RESULT  
1083 003756 173727 040500 CMPF #3,AC3 ;CHECK RESULT  
1084 003762 170000 CFCC ;GET CC  
1085 003764 001401 BEQ ,04  
1086 003766 104002 HLT 2 ;=21, / -7 NOT 3
```

```

1088 ;.....
(2) ;TEST 20 TEST OF MULD AND DIVD
(2) ;.....
(2) 003770 104400 TST20) SCOPE
1089
1090 003772 170127 047600 LDFPS #47600 ;LOAD DOUBLE MODE
1091 003776 172427 140640 LDD #=5,ACB ;LOAD A =5
1092 004002 171027 140500 MULD #=3,ACB ;MUL BY =3
1093 004006 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 004010 022700 047600 CMP #47600,FPS ;CHECK FLOATING POINT STATUS
(1) 004014 001401 BEQ ,+4 ;BRANCH IF OK
(1) 004016 104000 HLT ;FPS NOT EQUAL TO 47600
(1)
1094
1095 004020 174067 174766 STD ACB,ANS1 ;GET RESULT
1096 004024 173427 041160 CHPD #15.,ACB ;CHECK RESULT
1097 004030 170000 CFCC ;GET CC
1098 004032 001401 BEQ ,+4
1099 004034 104004 HLT 4 ;=5 X =3 NOT =15,
1100
1101 004036 174427 140400 DIVD #=2,ACB ;15, / =2
1102 004042 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 004044 022700 047610 CMP #47610,FPS ;CHECK FLOATING POINT STATUS
(1) 004050 001401 BEQ ,+4 ;BRANCH IF OK
(1) 004052 104000 HLT ;FPS NOT EQUAL TO 47610
(1)
1103
1104 004054 174067 174732 STD ACB,ANS1 ;STORE RESULT
1105 004060 173427 140760 CHPD #=7.5,ACB ;CHECK RESULT
1106 004064 170000 CFCC ;GET CC
1107 004066 001401 BEQ ,+4
1108 004070 104004 HLT 4 ;15, / =2 NOT =7.5

```

```

1110 .....
(2) ;TEST 21 TEST OF LDCFD,LDCDF
(2) ;.....
(2) TST21 SCOPE
1111
1112 004074 170127 047600 LDFPS 047600 ;SET DOUBLE MODE
1113 004100 172477 005024 LDD 0AD0101,AC0 ;LOAD A 0,-1.0,-1
1114 004104 012700 011102 MOV 0D1010,FPS ;GET ADDRESS OF DATA
1115 004110 177420 LDCFD (FPS)+,AC0 ;LOAD A -1.0,0.0
1116 004112 022700 011106 CMP 0D1010+4,FPS ;INC BY 4
1117 004116 001401 BEQ ,+4
1118 004120 104000 HLT ;FPS NOT 0D1010+4
1119
1120 004122 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 004124 022700 047610 CMP 047610,FPS ;CHECK FLOATING POINT STATUS
(1) 004130 001401 BEQ ,+4 ;BRANCH IF OK
(1) 004132 104000 HLT ;FPS NOT EQUAL TO 47610
1121
1122 004134 174007 174652 STD AC0,ANS1 ;GET ANSWER
1123 004140 173467 005106 CMPD 01000,AC0 ;CHECK RESULT
1124 004144 170000 CFCC ;GET CC
1125 004146 001401 BEQ ,+4
1126 004150 104004 HLT 4 ;AC0 NOT -1.0,0.0
1127
1128 004152 172507 004726 LDD 00101,AC1 ;LOAD A 0,-1.0,-1
1129 004156 170001 SETF ;SET FLOATING MODE
1130 004160 012700 011112 MOV 0D1010+10,FPS ;GET ADDRESS OF DATA +10
1131 004164 177540 LDCDF -(FPS),AC1 ;LOAD A -1.0
1132 004166 022700 011102 CMP 0D1010,FPS ;ADDRESS DECREMENT BY 10
1133 004172 001401 BEQ ,+4
1134 004174 104000 HLT ;FPS NOT 0D1010
1135
1136 004176 170200 STFPS FPS ;STORE FLOATING POINT STATUS
(1) 004200 022700 047410 CMP 047410,FPS ;CHECK FLOATING POINT STATUS
(1) 004204 001401 BEQ ,+4 ;BRANCH IF OK
(1) 004206 104000 HLT ;FPS NOT EQUAL TO 47410
1137
1138 004210 170011 SETD
1139 004212 174167 174574 STD AC1,ANS1 ;GET RESULT
1140 004216 173507 004676 CMPD WEIRD,AC1 ;CHECK RESULT
1141 004222 170000 CFCC
1142 004224 001401 BEQ ,+4
1143 004226 104004 HLT 4 ;RESULT NOT -1.1,0,-1
1144
1145 004230 170127 047440 LDFPS 047440 ;SET DOUBLE AND TRUNCATE MODES
1146 004234 177507 004642 LDCDF 01010,AC1 ;LOAD IT
1147 004240 174167 174546 STD AC1,ANS1 ;GET RESULT
1148 004244 173507 004642 CMPD 01001,AC1 ;CHECK RESULT
1149 004250 170200 STFPS FPS
1150 004252 170000 CFCC ;GET CC
1151 004254 001401 BEQ ,+4
1152 004256 104004 HLT 4 ;AC1 NOT -1.0,0.1

```

1153						
1154	004260	170127	047400	LDFPS	#47400	;SET ROUND AND FLOATING MODES
1155	004264	177567	004612	LDCDF	D1010,AC1	;LOAD A -1.0
1156	004270	174167	174510	STD	AC1,ANS1	;GET THE RESULT
1157	004274	170200		STFPS	FPS	
1158	004276	022767	000001 174510	CHP	#1,ANS2	;CHECK WORD 2 FOR A 1
1159	004304	001401		BEO	,+4	
1160	004306	104004		HLT	4	;LDCDF DID NOT ROUND PROPERLY

.....
;TEST 22 TEST OF STCFD,STCDF
;.....
TST22; SCOPE

1163						
1164	004312	170127	040200	LDFPS	#40200	;SET DOUBLE MODE
1165	004316	172667	004560	LDD	D1010,AC2	;LOAD A -1.0,-1.0
1166	004322	170001		SETF		;SET FLOATING MODE
1167	004324	176202		STCDF	AC2,AC2	;CLEAR RIGHT HALF
1168	004326	170011		SETD		;SET DOUBLE MODE
1169	004330	174267	174456	STD	AC2,ANS1	;GET RESULT
1170	004334	173667	004712	CHPD	D1000,AC2	;IS IT -1.0,0.0
1171	004340	170000		CFCC		;GET CC
1172	004342	001401		BEO	,+4	
1173	004344	104004		HLT	4	;AC1 NOT -1.0,0.0

1174						
1175	004346	172767	004716	LDD	D0111,AC3	;LOAD A 0,-1.0,-1
1176	004352	176303		STCFD	AC3,AC3	;CLEAR OUT RIGHT HALF?
1177	004354	174367	174432	STD	AC3,ANS1	;GET RESULT
1178	004360	173767	004676	CHPD	D0100X,AC3	;CHECK RESULT
1179	004364	170000		CFCC		;GET CC
1180	004366	001401		BEO	,+4	
1181	004370	104004		HLT	4	;ANS3 NOT 100.0,0 (ROUND)

1182						
1183	004372	172467	004672	LDD	D0111,AC0	;LOAD 0,-1.0,-1
1184	004376	170127	000040	LDFPS	#40	;FLOATING AND TRUNCATE MODES
1185	004402	176000		STCFD	AC0,AC0	;CLEAN RIGHT HALF
1186	004404	170011		SETD		;SET DOUBLE
1187	004406	174067	174400	STD	AC0,ANS1	;GET RESULT
1188	004412	173467	004632	CHPD	D0100,AC0	;CHECK IT
1189	004416	170000		CFCC		;GET CC
1190	004420	001401		BEO	,+4	
1191	004422	104004		HLT	4	;AC0 NOT 0,-1.0,0 (TRUNCATE)

.....
;TEST 23 TEST OF LDCIF,LDCID,STCFI,STCFI
;.....
TST23; SCOPE

1192						
1193						
(2)						
(2)						
(2)	004424	104400				
1194						
1195	004426	170127	047400	LDFPS	#47400	;FLOATING MODE
1196	004432	177027	000005	LDCIF	#5,AC0	;STORE A 5
1197	004436	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	004440	022700	047400	CHP	#47400,FPS	;CHECK FLOATING POINT STATUS
(1)	004444	001401		BEO	,+4	;BRANCH IF OK
(1)	004446	104000		HLT		;FPS NOT EQUAL TO 47400

(1)						
1198						
1199	004450	174067	174336	STF	AC0,ANS1	;GET THE RESULT
1200	004454	173427	040640	CMPF	#5,AC0	;CHECK IT
1201	004460	170000		CFCC		;GET CC
1202	004462	001401		BEO	,+4	
1203	004464	104002		HLT	2	;AC0 NOT 5.0
1204						
1205	004466	175400		STCFI	AC0,FPS	;CONVERT IT BACK
1206	004470	022700	000005	CMP	#5,FPS	;CHECK RESULT
1207	004474	001401		BEO	,+4	
1208	004476	104000		HLT		;FPS NOT 5
1209						
1210	004500	170011		SETD		;SET DOUBLE MODE
1211	004502	172567	004376	LDD	00101,AC1	;LOAD JUNK
1212	004506	012700	011102	MOV	001010,FPS	;LOAD ADDRESS OF DATA
1213	004512	177120		LDCID	(FPS)+,AC1	;CONVERT TO -1.0
1214	004514	022700	011104	CMP	001010+2,FPS	;CHECK ADDRESS
1215	004520	001401		BEO	,+4	
1216	004522	104000		HLT		;FPS NOT 001010+2
1217						
1218	004524	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	004526	022700	047610	CMP	047610,FPS	;CHECK FLOATING POINT STATUS
(1)	004532	001401		BEO	,+4	;BRANCH IF OK
(1)	004534	104000		HLT		;FPS NOT EQUAL TO 47610
(1)						
1219						
1220	004536	174167	174250	STD	AC1,ANS1	;GET RESULT
1221	004542	173527	140200	CMPD	#-1,AC1	;CHECK RESULT
1222	004546	170000		CFCC		;GET CC
1223	004550	001401		BEO	,+4	
1224	004552	104004		HLT	4	;AC1 NOT -1.0
1225						
1226	004554	175500		STCDI	AC1,FPS	;CONVERT IT BACK
1227	004556	022700	177777	CMP	#-1,FPS	;CHECK RESULT
1228	004562	001401		BEO	,+4	
1229	004564	104000		HLT		;FPS NOT -1
1230						
1231	004566	170001		SETF		;SET FLOATING MODE
1232	004570	177227	054321	LDCIF	054321,AC2	;LOAD 54321
1233	004574	174267	174212	STF	AC2,ANS1	;GET RESULT
1234	004600	170200		STFPS	FPS	;GET THE STATUS
1235	004602	173667	004500	CMPF	F9T01,AC2	;CHECK IT
1236	004606	170000		CFCC		;CHECK CC
1237	004610	001401		BEO	,+4	
1238	004612	104002		HLT	2	;AC2 NOT 54321.
1239						
1240	004614	052737	000017 177776	BIS	#17,00PS	;SET PS
1241	004622	175667	174164	STCFI	AC2,ANS1	;CONVERT IT BACK
1242	004626	013700	177776	MOV	00PS,FPS	;GET PS
1243	004632	042700	177760	BIC	#177700,FPS	;CLEAR JUNK
1244	004636	005700		TST	FPS	;IS IT 0
1245	004640	001401		BEO	,+4	;SKIP IF OK
1246	004642	104000		HLT		;PS NOT 0

1247	004644	170200		STFPS	FPS	;GET THE FPS
1248						
1249	004646	022767	054321 174136	CMP	#54321,ANS1	;CHECK RESULT
1250	004654	001401		BEO	,+4	
1251	004656	104001		HLT	1	;ANS1 NOT 54321
1252						
1253						
(2)						
(2)						
(2)	004660	104400				
1254						
1255	004662	170127	047500	LDFPS	047500	;FLOATING AND LONG MODES
1256	004666	177327	177773	LDCLF	0-5,AC3	;LOAD A -5
1257	004672	000401		BR	,+4	
1258	004674	000000		HALT		;LDCLF INCREMENTED BY 4 NOT 2
1259						
1260	004676	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	004700	022700	047510	CMP	047510,FPS	;CHECK FLOATING POINT STATUS
(1)	004704	001401		BEO	,+4	;BRANCH IF OK
(1)	004706	104000		HLT		;FPS NOT EQUAL TO 47510
(1)						
1261						
1262	004710	174367	174076	STP	AC3,ANS1	;GET THE RESULT
1263	004714	173727		CMP	(7)*,AC3	;CHECK IT
1264	004716	144640		144640		
1265	004720	170000		CPCG		;GET CC
1266	004722	001401		BEO	,+4	
1267	004724	104002		HLT	2	;AC3 NOT -5
1268						
1269	004726	012700	011156	MOV	#AANS1,FPS	;SET UP ADDRESS OF ADDRESS
1270	004732	175730		STCFI	AC3,0(FPS)+	;STORE IN ANS1
1271	004734	022700	011160	CMP	#AANS1+2,FPS	;CHECK ADDRESS
1272	004740	001401		BEO	,+4	
1273	004742	104000		HLT		;ADDRESS IN FPU NOT AANS1+2
1274						
1275	004744	170200		STFPS	FPS	;GET STATUS
1276	004746	022767	177773 174036	CMP	0-5,ANS1	;CHECK LEFT HALF
1277	004754	001401		BEO	,+4	
1278	004756	104002		HLT	2	;LEFT NOT -5
1279						
1280	004760	005767	174030	TST	ANS2	;CHECK RIGHT HALF OF RESULT
1281	004764	001401		BEO	,+4	
1282	004766	104002		HLT	2	;ANS2 NOT 0
1283						
1284	004770	170011		SETD		;SET DOUBLE MODE
1285	004772	177067	004302	LDCLD	D9T01,AC0	;LOAD WEIRD NUMBER
1286	004776	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	005000	022700	047700	CMP	047700,FPS	;CHECK FLOATING POINT STATUS
(1)	005004	001401		BEO	,+4	;BRANCH IF OK
(1)	005006	104000		HLT		;FPS NOT EQUAL TO 47700
(1)						
1287						
1288	005010	174067	173776	STD	AC0,ANS1	;GET RESULT
1289	005014	173467	004266	CMPD	F9T01,AC0	;CHECK IT

.....
;TEST 24 TEST OF LDCLF,LDCLD,STCFI,STCDI
;.....
TST24: SCOPE

1290	005020	170000				CFCC);GET CC
1291	005022	001401				BEO	,+4		
1292	005024	104004				HLT	4);ACB NOT F9T01
1293									
1294	005026	012700	001016			MOV	#ANS3,FPS		
1295	005032	175440				STCDL	ACB,=(FPS));GET IT BACK
1296	005034	022700	001012			CMP	#ANS1,FPS);CHECK ADDRESS
1297	005040	001401				BEO	,+4		
1298	005042	104000				HLT);FPS NOT #ANS1
1299									
1300	005044	170200				STFPS	FPS);STORE FLOATING POINT STATUS
(1)	005046	022700	047700			CMP	047700,FPS);CHECK FLOATING POINT STATUS
(1)	005052	001401				BEO	,+4);BRANCH IF OK
(1)	005054	104000				HLT);FPS NOT EQUAL TO 47700
(1)									
1301									
1302	005056	170200				STFPS	FPS		
1303	005060	026767	004214	173724		CMP	D5T01,ANS1);CHECK LEFT
1304	005066	001401				BEO	,+4		
1305	005070	104002				HLT	2);ANS1 NOT
1306									
1307	005072	026767	004204	173714		CMP	D5T01+2,ANS2);CHECK RIGHT
1308	005100	001401				BEO	,+4		
1309	005102	104002				HLT	2);ANS2 NOT
1310									
1311									
(2));TEST 25 TEST OF MODF AND MODD
(2)									
(2)	005104	104400);TEST 25; SCOPE
1312									
1313	005106	170127	047400			LDFPS	047400);SET FLOATING MODE
1314	005112	172467	004054			LDF	DALTB+2,ACB);LOAD 52525,52525 INTO ACB
1315	005116	171427	040200			MODF	#1,ACB);MODF BY A 1 #MGY*
1316	005122	170200				STFPS	FPS);STORE FLOATING POINT STATUS
(1)	005124	022700	047404			CMP	047404,FPS);CHECK FLOATING POINT STATUS
(1)	005130	001401				BEO	,+4);BRANCH IF OK
(1)	005132	104000				HLT);FPS NOT EQUAL TO 47404
(1)									
1317									
1318	005134	174067	173652			STF	ACB,ANS1);GET RESULT
1319	005140	170500				YSTF	ACB);CHECK FRACTION
1320	005142	170000				CFCC			
1321	005144	001401				BEO	,+4		
1322	005146	104002				HLT	2);FRACTION NOT 0
1323									
1324	005150	174167	173636			STF	AC1,ANS1);GET IT
1325	005154	173567	004012			CMPF	DALTB+2,AC1);CHECK INTEGER
1326	005160	170000				CFCC			
1327	005162	001401				BEO	,+4		
1328	005164	104002				HLT	2);INTEGER IS NOT 52525,52525
1329									
1330	005166	172627	040200			LDF	#1,AC2);LOAD A 1
1331	005172	171667	004042			MODF	D20,AC2);MOD BY 20000,0
1332	005176	170200				STFPS	FPS);STORE FLOATING POINT STATUS

(1)	005200	022700	047400	CMP	047400,FPS	;CHECK FLOATING POINT STATUS
(1)	005204	001401		BEO	,+4	;BRANCH IF OK
(1)	005206	104000		HLT		;FPS NOT EQUAL TO 47400
(1)						
1333						
1334	005210	174267	173576	STF	AC2,ANS1	;GET IT
1335	005214	173667	004020	CMPF	D20,AC2	;CHECK FRACT
1336	005220	170000		CFCC		
1337	005222	001401		BEO	,+4	
1338	005224	104000		HLT	2	;RESULT NOT 20000,0
1339						
1340	005226	174367	173560	STF	AC3,ANS1	;GET INT
1341	005232	170503		TSTF	AC3	;CHECK FOR 0
1342	005234	170000		CFCC		
1343	005236	001401		BEO	,+4	
1344	005240	104000		HLT	2	;RESULT NOT 0
1345						
1346	005242	170011		SETD		;SET DOUBLE MODE
1347	005244	172467	003760	LDD	D40,AC0	;LOAD A 40000,0,0,0
1348	005250	171467	003744	MODD	D37,AC0	;MOD BY 37400,0,0,0
1349	005254	173467	003730	CMPD	D40,AC0	;CHECK FOR 40000,0,0,0
1350	005260	170000		CFCC		
1351	005262	001401		BEO	,+4	
1352	005264	104000		HLT	4	;RESULT NOT 40000,0,0,0
1353						
1354	005266	174367	173520	STD	AC3,ANS1	;GET THE RESULT
1355	005272	170503		TSTD	AC3	;CHECK FOR 0
1356	005274	170000		CFCC		
1357	005276	001401		BEO	,+4	
1358	005300	104000		HLT	4	;RESULT NOT 0
1359						
1363						
(2)						
(2)						
(2)	005302	104400				
1364						
1365	005304	170127	047600	LDFPS	047600	;DOUBLE MODE
1366	005310	172467	003674	LDD	D40,AC0	;LOAD DUMMY DATA
1367	005314	172467	003626	LDD	DHZERO,AC0	;LOAD A -0
1368	005320	170200		STFPS	FPS	;STORE FLOATING POINT STATUS
(1)	005322	170367	173504	STST	FEC	;STORE EXCEPTION CODES
(1)	005326	022700	147614	CMP	0147614,FPS	;CHECK FLOATING POINT STATUS
(1)	005332	001401		BEO	,+4	;BRANCH IF OK
(1)	005334	104000		HLT		;FPS NOT EQUAL TO 147614
(1)						
(1)	005336	022767	000014 173466	CMP	014,FEC	;CHECK FLOATING EXCEPTION CODE
(1)	005344	001401		BEO	,+4	;BRANCH IF OK
(1)	005346	104000		HLT		;FEC NOT EQUAL TO 14
(1)						
(1)	005350	022767	005314 173456	CMP	05314,FEA	;CHECK FLOATING EXCEPTION ADDRESS
(1)	005356	001401		BEO	,+4	;BRANCH IF OK
(1)	005360	104000		HLT		;FEA NOT EQUAL TO 5314
(1)						
1369						

.....
;TEST 26 LDD OF -0
;.....
TST26: SCOPE

151

1370	005362	174067	173424		STD	ACB,ANS1	IGET RESULT FOR TYPING
1371	005366	173467	003616		CHPD	D48,ACB	IDID IT CHANGE ACB?
1372	005372	170200			STFPS	FPS	IGET FPS FOR TYPING
1373	005374	170000			CFCC		IGET CC
1374	005376	001401			BEO	,+4	IBKIP IF OK
1375	005400	104004			HLT	4	IRESULT IS WRONG
1376							
1377	005402	170127	040200		LDFPS	040200	IDOUBLE MODE
1378	005406	172467	003534		LDF	DMZERO,ACB	ILOAD A -0
1379	005412	170200			STFPS	FPS	ISTORE FLOATING POINT STATUS
(1)	005414	022700	040214		CHP	040214,FPS	ICHECK FLOATING POINT STATUS
(1)	005420	001401			BEO	,+4	IBRANCH IF OK
(1)	005422	104000			HLT		IFPS NOT EQUAL TO 40214
(1)							
1380							
1381	005424	174067	173362		STD	ACB,ANS1	IGET RESULT
1382	005430	173467	003512		CHPD	DMZERO,ACB	ICHP TO -0
1383	005434	170200			STFPS	FPS	IGET FPS
1384	005436	170000			CFCC		IGET CC
1385	005440	001401			BEO	,+4	IBKIP IF OK
1386	005442	104004			HLT	4	IRESULT NOT -0
1387							
1388							
(2)							
(2)							
(2)	005444	104400					
1389							
1390	005446	170127	001000		LDFPS	01000	IFLOATING/OVERFLOW
1391	005452	013777	177776	005164	MOV	00PS,0FPVECT+2	ISET THE MODE IN PS WORD
1392	005460	005001			CLR	R1	ICLEAR FLAG WORD
1393	005462	012777	011074	005152	MOV	0CHKERR,0FPVECT	ISET TRAP ADDRESS
1394	005470	172427	076101		LDF	01E36,ACB	ILOAD A LARGE NUMBER INTO ACB,
1395	005474	171027	076101		MULF	01E36,ACB	IMULTIPLY BY A LARGE NUMBER
1396	005500	174067	173306		STF	ACB,ANS1	IGET FOR TYPING
1397	005504	170200			STFPS	FPS	IGET STATUS
1398	005506	005701			TST	R1	IDID IT TRAP?
1399	005510	001001			BNE	JS	IYES
1400	005512	104002			HLT	2	IDID NOT TRAP ON OVERFLOW
1401	005514						
(1)	005514	170200					
(1)	005516	170367	173310		STFPS	FPS	ISTORE FLOATING POINT STATUS
(1)	005522	022700	101002		STST	FEC	ISTORE EXCEPTION CODES
(1)	005526	001401			CHP	0101002,FPS	ICHECK FLOATING POINT STATUS
(1)	005530	104000			BEO	,+4	IBRANCH IF OK
(1)					HLT		IFPS NOT EQUAL TO 101002
(1)							
(1)	005532	022767	000010	173272	CHP	010,FEC	ICHECK FLOATING EXCEPTION CODE
(1)	005540	001401			BEO	,+4	IBRANCH IF OK
(1)	005542	104000			HLT		IFEC NOT EQUAL TO 10
(1)							
(1)	005544	022767	005474	173262	CHP	05474,FEA	ICHECK FLOATING EXCEPTION ADDRESS
(1)	005552	001401			BEO	,+4	IBRANCH IF OK
(1)	005554	104000			HLT		IFEA NOT EQUAL TO 5474
(1)							
1402							

```

.....
ITEST 27      MULF ERROR - OVERFLOW
.....
TST27: SCOPE

```

```
1403 .....  
(2) ;TEST 30 DIVF ERROR - UNDERFLOW  
(2) .....  
(2) 005556 104400 TST30) SCOPE  
1404  
1405 005560 170127 002000 LDPPS #2000 ;FLOATING/UNDERFLOW  
1406 005564 005001 CLR R1 ;CLEAR FLAG WORD  
1407 005566 172427 002252 LDF #1E-36,AC0 ;LOAD A SMALL NUMBER  
1408 005572 174427 076101 15) DIVF #1E36,AC0 ;DIVIDE BY A LARGE NUMBER  
1409 005576 174067 173210 STP AC0,ANS1 ;GET FOR TYPING  
1410 005602 170200 STPPS FPS ;GET STATUS  
1411 005604 005701 TST R1 ;DID IT TRAP?  
1412 005606 001001 BNE 35 ;SKIP IF SET  
1413 005610 104002 HLT 2 ;DID NOT TRAP ON UNDERFLOW  
1414 005612 35) STPPS FPS ;STORE FLOATING POINT STATUS  
(1) 005612 170200 STST FEC ;STORE EXCEPTION CODES  
(1) 005614 170367 173212 CMP #102000,FPS ;CHECK FLOATING POINT STATUS  
(1) 005620 022700 102000 BEQ ,+4 ;BRANCH IF OK  
(1) 005624 001401 HLT ;FPS NOT EQUAL TO 102000  
(1) 005626 104000  
(1) 005630 022767 000012 173174 CMP #12,FEC ;CHECK FLOATING EXCEPTION CODE  
(1) 005636 001401 BEQ ,+4 ;BRANCH IF OK  
(1) 005640 104000 HLT ;FEC NOT EQUAL TO 12  
(1) 005642 022767 005572 173164 CMP #5572,FEA ;CHECK FLOATING EXCEPTION ADDRESS  
(1) 005650 001401 BEQ ,+4 ;BRANCH IF OK  
(1) 005652 104000 HLT ;FEA NOT EQUAL TO 5572  
1415  
1416 .....  
(2) ;TEST 31 STCF! ERROR - CONVERSION(6)  
(2) .....  
(2) 005654 104400 TST31) SCOPE  
1417  
1418 005656 170127 000400 LDPPS #400 ;FLOATING/INTEGER/CONVERSION  
1419 005662 005001 CLR R1 ;CLEAR FLAG WORD  
1420 005664 172527 076101 LDF #1E36,AC1 ;LOAD LARGE NUMBER  
1421 005670 175567 173116 15) STCF! AC1,ANS1 ;TRY TO STUFF INTO 16 BITS  
1422 005674 170200 STPPS FPS ;GET STATUS  
1423 005676 005701 TST R1 ;TRAP FLAG SET?  
1424 005700 001001 BNE 35 ;SKIP IF SET  
1425 005702 104000 HLT ;DID NOT TRAP ON CONVERT  
1426 005704 35) STPPS FPS ;STORE FLOATING POINT STATUS  
(1) 005704 170200 STST FEC ;STORE EXCEPTION CODES  
(1) 005706 170367 173120 CMP #100400,FPS ;CHECK FLOATING POINT STATUS  
(1) 005712 022700 100400 BEQ ,+4 ;BRANCH IF OK  
(1) 005716 001401 HLT ;FPS NOT EQUAL TO 100400  
(1) 005720 104000  
(1) 005722 022767 000006 173102 CMP #6,FEC ;CHECK FLOATING EXCEPTION CODE  
(1) 005730 001401 BEQ ,+4 ;BRANCH IF OK  
(1) 005732 104000 HLT ;FEC NOT EQUAL TO 6  
(1)
```

```
(1) 005734 022767 005670 173072      CMP      05670,FEA      ;CHECK FLOATING EXCEPTION ADDRESS
(1) 005742 001401      BEQ      ,+4          ;BRANCH IF OK
(1) 005744 104000      HLT                               ;FEA NOT EQUAL TO 5670
(1)
1427
1428      ;.....
(2)      ;TEST 32      DIVF BY 0 ERROR
(2)      ;.....
(2) 005746 104400      TST32:  SCOPE
1429
1430 005750 170127 000000      LDPPS   00          ;FLOATING
1431 005754 005001      CLR     R1          ;CLEAR FLAG
1432 005756 174527 000000      1SI    DIVF   00,AC1 ;DIVIDE BY 0
1433 005762 170200      STPPS   FPS        ;GET STATUS
1434 005764 005701      TST     R1          ;CHECK FLAG
1435 005766 001001      BNE     35          ;SKIP IF SET
1436 005770 104000      HLT                               ;DIVIDE BY 0 DID NOT TRAP
1437 005772      3SI
(1) 005772 170200      STPPS   FPS        ;STORE FLOATING POINT STATUS
(1) 005774 170367 173032      STST    FEC        ;STORE EXCEPTION CODES
(1) 006000 022700 100000      CMP     0100000,FPS ;CHECK FLOATING POINT STATUS
(1) 006004 001401      BEQ     ,+4          ;BRANCH IF OK
(1) 006006 104000      HLT                               ;FPS NOT EQUAL TO 100000
(1)
(1) 006010 022767 000004 173014      CMP     04,FEC      ;CHECK FLOATING EXCEPTION CODE
(1) 006016 001401      BEQ     ,+4          ;BRANCH IF OK
(1) 006020 104000      HLT                               ;FEC NOT EQUAL TO 4
(1)
(1) 006022 022767 005756 173004      CMP     05756,FEA   ;CHECK FLOATING EXCEPTION ADDRESS
(1) 006030 001401      BEQ     ,+4          ;BRANCH IF OK
(1) 006032 104000      HLT                               ;FEA NOT EQUAL TO 5756
(1)
1438
1439      ;.....
(2)      ;TEST 33      LDF -0 ERROR
(2)      ;.....
(2) 006034 104400      TST33:  SCOPE
1440
1441 006036 170127 004000      LDPPS   04000      ;FLOATING/UNDEFINED VARIABLE
1442 006042 005001      CLR     R1          ;CLEAR FLAG
1443 006044 172667 003076      1SI    LDF    0MZERO,AC2 ;LOAD AN UNDEFINED VARIABLE
1444 006050 170200      STPPS   FPS        ;GET STATUS
1445 006052 005701      TST     R1          ;CHECK FLAG
1446 006054 001001      BNE     35          ;SKIP IF SET
1447 006056 104000      HLT                               ;LOAD OF -0 DID NOT TRAP
1448 006060      3SI
(1) 006060 170200      STPPS   FPS        ;STORE FLOATING POINT STATUS
(1) 006062 170367 172744      STST    FEC        ;STORE EXCEPTION CODES
(1) 006066 022700 104014      CMP     0104014,FPS ;CHECK FLOATING POINT STATUS
(1) 006072 001401      BEQ     ,+4          ;BRANCH IF OK
(1) 006074 104000      HLT                               ;FPS NOT EQUAL TO 104014
(1)
(1) 006076 022767 000014 172726      CMP     014,FEC     ;CHECK FLOATING EXCEPTION CODE
(1) 006104 001401      BEQ     ,+4          ;BRANCH IF OK
```

```

(1) 006106 104000          MLT          IFEC NOT EQUAL TO 14
(1)
(1) 006110 022767 006044 172716      CMP      00644,FEA      ;CHECK FLOATING EXCEPTION ADDRESS
(1) 006116 001401          BEQ      ,+4          ;BRANCH IF OK
(1) 006120 104000          MLT          IFEA NOT EQUAL TO 6044
(1)
1449
1450
(2)
(2)
(2) 006122 104400          ;=====
;TEST 34          OPCODE ERROR
;=====
TST34: SCOPE
1451
1452 006124 170127 000000          LDPPS   00          ;FLOATING
1453 006130 005001          CLR     R1          ;CLEAR FLAG
1454 006132 177707          15:    177707          ;ILLEGAL OPCODE
1455 006134 170200          STPPS   FPS        ;GET STATUS
1456 006136 005701          TST     R1          ;CHECK FLAG
1457 006140 001001          ONE     35         ;SKIP IF SET
1458 006142 104000          MLT          ;NOT AN ILLEGAL OPCODE?
1459 006144          35:
(1) 006144 170200          STPPS   FPS        ;STORE FLOATING POINT STATUS
(1) 006146 170367 172660          STST   FEC        ;STORE EXCEPTION CODES
(1) 006152 022700 100000          CMP     0100000,FPS ;CHECK FLOATING POINT STATUS
(1) 006156 001401          BEQ     ,+4          ;BRANCH IF OK
(1) 006160 104000          MLT          IFPS NOT EQUAL TO 100000
(1)
(1) 006162 022767 000002 172642      CMP     02,FECC     ;CHECK FLOATING EXCEPTION CODE
(1) 006170 001401          BEQ     ,+4          ;BRANCH IF OK
(1) 006172 104000          MLT          IFEC NOT EQUAL TO 2
(1)
(1) 006174 022767 006132 172632      CMP     00132,FEA   ;CHECK FLOATING EXCEPTION ADDRESS
(1) 006202 001401          BEQ     ,+4          ;BRANCH IF OK
(1) 006204 104000          MLT          IFEA NOT EQUAL TO 0132
(1)
1460
1461
(2)
(2)
(2) 006206 104400          ;=====
;TEST 35          UBREAK TRAP
;=====
TST35: SCOPE
1462
1463 006210 005001          CLR     R1          ;CLEAR FLAG
1464 006212 013767 177776 003100      MOV     00PS,SAVSTS ;GET PS
1465 006220 005037 177776          CLR     00PS        ;ZERO IT
1466 006224 012703 000314      MOV     0314,R3     ;LOAD 314 INTO
1467 006230 170003          LDUB           ;UBREAK REG.
1468 006232 170127 000020          LDPPS   020        ;FLOATING/MAINT
1469 006236 174427 040200          15:    DIVF   01,ACB ;DIVIDE SHOULD TRAP
1470 006242 170200          STPPS   FPS        ;GET STATUS
1471 006244 005701          TST     R1          ;CHECK FLAG
1472 006246 001001          ONE     35         ;SKIP IF NOT SET
1473 006250 104000          MLT          ;UBREAK DID NOT TRAP
1474 006252          35:
(1) 006252 170200          STPPS   FPS        ;STORE FLOATING POINT STATUS
(1) 006254 170367 172552          STST   FEC        ;STORE EXCEPTION CODES

```

```

(1) 006260 022700 100020      CMP      0100020,FPS      ;CHECK FLOATING POINT STATUS
(1) 006264 001401      BEQ      ,04              ;BRANCH IF OK
(1) 006266 104000      HLT
(1)
(1) 006270 022767 000016 172534  CMP      016,FEC        ;CHECK FLOATING EXCEPTION CODE
(1) 006276 001401      BEQ      ,04              ;BRANCH IF OK
(1) 006300 104000      HLT                    ;FEC NOT EQUAL TO 16
(1)
(1) 006302 022767 006236 172524  CMP      06236,FEA      ;CHECK FLOATING EXCEPTION ADDRESS
(1) 006310 001401      BEQ      ,04              ;BRANCH IF OK
(1) 006312 104000      HLT                    ;FEA NOT EQUAL TO 6236
(1)
1475 006314 170127 040000      LDPPS   040000          ;CLEAR FPS
1476 006320 016737 002774 179776  MOV     SAVSTS,00PS    ;RESTORE PS
1477
1478
(2)
(2)
(2) 006326 104400      ;.....
ITEST 36      ADDF ERROR - OVERFLOW
;.....
TST36) SCOPE
1479
1480 006330 170127 001000      LDPPS   01000          ;FLOATING/OVERFLOW
1481 006334 005001      CLR     R1              ;CLEAR FLAG
1482 006336 172767 002574      LDF     0016,AC3       ;LOAD A BIG NUMBER
1483 006342 172367 002570 181      ADDF   0016,AC3       ;MAKE OVERFLOW
1484 006346 170200      STPPS  FPS              ;GET STATUS
1485 006350 174367 172436  STP    AC3,ANS1       ;GET RESULT
1486 006354 005701      TST    R1              ;FLAG SET?
1487 006356 001001      BNE    38              ;SKIP IF SET
1488 006360 104002      HLT    2              ;DID NOT OVERFLOW
1489
(1) 006362 170200      STPPS  FPS              ;STORE FLOATING POINT STATUS
(1) 006364 170367 172442  STST   FEC              ;STORE EXCEPTION CODES
(1) 006370 022700 101006  CMP    0101006,FPS    ;CHECK FLOATING POINT STATUS
(1) 006374 001401      BEQ    ,04              ;BRANCH IF OK
(1) 006376 104000      HLT                    ;FPS NOT EQUAL TO 101006
(1)
(1) 006400 022767 000010 172424  CMP    010,FEC        ;CHECK FLOATING EXCEPTION CODE
(1) 006406 001401      BEQ    ,04              ;BRANCH IF OK
(1) 006410 104000      HLT                    ;FEC NOT EQUAL TO 10
(1)
(1) 006412 022767 006342 172414  CMP    06342,FEA      ;CHECK FLOATING EXCEPTION ADDRESS
(1) 006420 001401      BEQ    ,04              ;BRANCH IF OK
(1) 006422 104000      HLT                    ;FEA NOT EQUAL TO 6342
(1)
1490
1491
(2)
(2)
(2) 006424 104400      ;.....
ITEST 37      SUBF ERROR - UNDERFLOW
;.....
TST37) SCOPE
1492
1493 006426 170127 002000      LDPPS   02000          ;FLOATING/UNDERFLOW
1494 006432 005001      CLR     R1              ;CLEAR FLAG
1495 006434 172427 000430 181      LDF     0,07E-37,AC0   ;LOAD SMALL NUMBER
1496 006440 173027 000504      SUBF   0,09E-37,AC0   ;SUB BIG NUMBER

```


1497	006444	174067	172342	STP	ACB,ANS1	:GET RESULT
1498	006450	170200		STFPS	FPS	:GET STATUS
1499	006452	005701		TST	R1	:FLAG SET?
1522	006454	001001		BNE	38	:SKIP IF SET
1521	006456	104002		HLT	2	:NO UNDERFLOW
1522	006460					
			381			
(1)	006460	170200		STFPS	FPS	:STORE FLOATING POINT STATUS
(1)	006462	170367	172344	STST	FEC	:STORE EXCEPTION CODES
(1)	006466	022700	102014	CHP	#102014,FPS	:CHECK FLOATING POINT STATUS
(1)	006472	001401		BEO	,+4	:BRANCH IF OK
(1)	006474	104000		HLT		:FPS NOT EQUAL TO 102014
(1)						
(1)	006476	022767	000012 172326	CHP	#12,FEC	:CHECK FLOATING EXCEPTION CODE
(1)	006504	001401		BEO	,+4	:BRANCH IF OK
(1)	006506	104000		HLT		:FEC NOT EQUAL TO 12
(1)						
(1)	006510	022767	006440 172316	CHP	06440,FEA	:CHECK FLOATING EXCEPTION ADDRESS
(1)	006516	001401		BEO	,+4	:BRANCH IF OK
(1)	006520	104000		HLT		:FEA NOT EQUAL TO 6440
(1)						
1523						
1524	006522	104400		SCOPE		
1525						
1526	006524	170127	040000	LDFPS	040000	:I/O FPS
1527	006530	012777	000760 004104	MOV	#FLTERR,#FPVECT	:RESTORE VECTOR
1528	006536	012777	000340 004100	MOV	0340,#FPVECT+2	:RESTORE LEVEL 7
1529	006544	004767	001230	JSR	PC,CODE	:DO FLOATING POINT EXERCISER THROUGHOUT MEMORY
1510	006550	032737	001000 177570	BIT	#SW09,#SWR	:RELOCATE CODE?
1511	006556	001402		BEO	RELEX	:RELOCATE
1512	006560	000167	002540	JMP	DONE+2	:DO NOT RELOCATE

				;RELOCATOR AND EXECUTOR		
1514						
1515	006564	012737	006672	000004	RELEXI	MOV #TRAP4,004
1516	006572	012701	016000			MOV #CODE,R1 ;LOAD STARTING POINT
1517	006576	012704	020000			MOV #20000,R4 ;LOAD DESTINATION
1518	006602	004767	000102			JSR PC,MOVE ;MOVE THE CODE THERE
1519	006606	004737	020000			JSR PC,0#20000 ;EXECUTE THE SK BANK
1520	006612	012704	024000			MOV #24000,R4
1521	006616	010401		151		MOV R4,R1 ;GET THE ADDRESS
1522	006620	010146				MOV R1,-(6) ;SAVE ON THE STACK
1523	006622	012704	020000			MOV #20000,R4 ;SET TO GET THE LOADER
1524	006626	004767	000056			JSR PC,MOVE ;GET THE PAGE
1525	006632	011604				MOV (6),R4 ;GET THE DESTINATION ADDRESS
1526	006634	012701	010000			MOV #CODE,R1 ;DATA TO BE TRANSFERED
1527	006640	004767	000044			JSR PC,MOVE ;MOVE IT
1528	006644	004776	000000			JSR PC,0(6) ;EXECUTE IT
1529	006650	012604				MOV (6)*,R4 ;GET DEST ADDRESS
1530	006652	012701	020000			MOV #20000,R1 ;GET SAVE ADDRESS
1531	006656	004767	000026			JSR PC,MOVE ;RESTORE LOADER
1532	006662	022704	160000			CHP #160000,R4 ;END OF WORLD
1533	006666	001353				BNE 15 ;NO - LOOP
1534	006670	000402				BR X2X ;YES = SKIP STACK CLEANER
1535						
1536	006672	062706	000010		TRAP4)	ADD #10,SP ;CLEAR THE STACK
1537	006676	012737	000006	000004	X2XI	MOV #6,004 ;RESET 4
1538	006704	000167	002412			JMP DONE
1539						
1540						;MOVE SUBROUTINE = MOVES 1K CHUNK FROM (1) TO (4)
1541						
1542	006710	010246			MOVEI	MOV R2,-(6) ;SAVE R2
1543	006712	012702	002000			MOV #2000,R2 ;LOAD A 1K COUNT
1544	006716	016746	171054			MOV PS,-(6) ;SAVE PSW
1545	006722	005046				CLR -(6) ;PUT 0 ON STACK
1546	006724	012746	006732			MOV #15,-(6) ;PUT RETURN ADD ON STACK
1547	006730	000002				RTI ;CONT. WITH NEW PS
1548	006732	011114		151		MOV (1),(4) ;MOVE A WORD
1549	006734	010400				MOV R4,FP5 ;SAVE NEW ADD FOR TYPE OUT
1550	006736	022124				CHP (1)*,(4)* ;DID WD GET TO NEW LOCATION OK
1551	006740	001401				BEG ,*4 ;YES
1552	006742	104000				HLT ;DATA TRANSFER BAD R1=OLD ADD *2 R4=NEW ADD *2
1553	006744	005302				DEC R2 ;DEC COUNT
1554	006746	001371				BNE 15 ;LOOP UNTIL DONE
1555	006750	012746	006756			MOV #25,-(6) ;STOR RETURN ADD
1556	006754	000002				RTI ;GET OLD PSW AND NEW RETURN ADD
1557	006756	012602		251		MOV (6)*,R2 ;RESTORE R2
1558	006760	000207				RTS PC ;RETURN

```

1560      010000      010000      ,=10000
1561      010000      012737      000037      001000      CODE1      MOV      #37,00|CNT
1562      (2)
1562      (2)
1562      (2)      010006      104400      ;=====
1563      010010      170127      047600      ;TEST 40      TEST FOR CONVERSION, ADD AND SUBD
1564      010014      177327      000004      ;=====
1565      010020      177227      000002      TST40|      SCOPE
1566      010024      172302      ;LDFPS      #47600
1567      010026      173302      ;LDCID      #4,AC3      ;LOAD AC3 WITH A FLOATING 4
1568      010030      175737      001012      ;LDCID      #2,AC2      ;LOAD AC2 WITH A FLOATING 2
1569      010034      022737      000004      001012      ;ADD      AC2,AC3      ;ADD 2+4 = 6 IN AC3
1570      010042      170200      ;SUBD      AC2,AC3      ;SUB 2 FROM 6
1571      010044      001401      000004      001012      ;STCDI     AC3,00ANS1      ;00ANS1 SHOULD =4
1572      010046      104001      ;CMP      #4,00ANS1      ;DOES 00ANS1 =4
1573      ;STFPS      FPS
1574      (2)      ;BEO      ,+4      ;YES
1574      (2)      ;HLT      1      ;00ANS1 SHOULD = 4
1574      (2)      ;=====
1574      (2)      ;TEST 41      LOD AND STD TEST
1574      (2)      ;=====
1574      (2)      010050      104400      ;TST41|      SCOPE
1575      010052      177027      000005      ;LDCID      #5,AC0      ;LOAD AC0 WITH A FLOATING 5
1576      010056      174005      ;STD      AC0,AC5      ;NOW PUT IT INTO AC5
1577      010060      172605      ;LDD      AC5,AC2      ;NOW PUT IT INTO AC2
1578      010062      172200      ;ADD      AC0,AC2      ;ADD 5 TO 5
1579      010064      173205      ;SUBD     AC5,AC2      ;SUB 5 FROM 10
1580      010066      175637      001012      ;STCDI     AC2,00ANS1      ;PUT ANS INTO 00ANS1
1581      010072      022737      000005      001012      ;CMP      #5,00ANS1      ;WERE THE TWO AC'S EQUAL
1582      010100      170200      ;STFPS     FPS
1583      010102      001401      ;BEO      ,+4      ;YES
1584      010104      104001      ;HLT      1      ;ANS1 SHOULD = 5
1585
1586      (2)      ;=====
1586      (2)      ;TEST 42      MULD AND DIVD TEST
1586      (2)      ;=====
1586      (2)      010106      104400      ;TST42|      SCOPE
1587      010110      177227      000032      ;LDCID      #52,AC2      ;LOAD AC2 WITH FLOATING DOUBLE 52
1588      010114      174204      ;STD      AC2,AC4      ;PUT IT INTO AC4
1589      010116      177327      000025      ;LDCID      #25,AC3      ;LOAD AC3 WITH FL DB 25
1590      010122      171304      ;MULD     AC4,AC3      ;MUL 52X25 RESULT IN AC3
1591      010124      174754      ;DIVD     AC4,AC3      ;DIV 52 INTO RESULT IN AC3
1592      010126      171304      ;MULD     AC4,AC3      ;MUL 52X25
1593      010130      172304      ;ADD      AC4,AC3      ;ADD AC4 TO AC3 TO MAKE 53 TIMES
1594      010132      012704      000025      ;MOV      #25,R4      ;SET UP COUNTER
1595      010136      173304      ;SUBD     AC4,AC3      ;SUB 52 FROM AC3 25 TIMES
1596      010140      005304      ;DEC      R4
1597      010142      001375      ;BNE      SUBT
1598      010144      175737      001012      ;STCDI     AC3,00ANS1      ;ANS SHOULD BE 52
1599      010150      170200      ;STFPS     FPS
1600      010152      022737      000032      001012      ;CMP      #52,00ANS1      ;IS ANS CORRECT?
1601      010160      001401      ;BEO      ,+4      ;YES
1602      010162      104001      ;HLT      1      ;ANS SHULD BE 52

```

```

1624 (2)
1625 (2)
1626 (2)
1627 010164 104400
1628 010166 177027 029252
1629 010172 177127 000025
1630 010176 171100
1631 010200 174137 001012
1632 010204 012702 000024
1633 010210 174005
1634 010212 172005
1635 010214 005302
1636 010216 001375
1637 010220 174037 001022
1638 010224 170200
1639 010226 173437 001012
1640 010232 170000
1641 010234 001401
1642 010236 104010
1643 010240 177327 000002
1644 010244 012702 000013
1645 010250 174304
1646 010252 171304
1647 010254 005302
1648 010256 001375
1649 010260 175703
1650 010262 010337 001012
1651 010264 022703 010000
1652 010272 170200
1653 010274 001401
1654 010276 104001
1655 010300 177227 010000
1656 010304 177127 000002
1657 010310 012702 000013
1658 010314 174601
1659 010316 005302
1660 010320 001375
1661 010322 175603
1662 010324 010237 001012
1663 010330 022703 000002
1664 010334 170200
1665 010336 001401
1666 010340 104001

```

```

.....
TEST 43 EXERCISER TEST
.....
TST43: SCOPE
LDCID #29252,AC0 ;LOAD 29252 INTO AC0
LDCID #29,AC1 ;LOAD AC1 WITH 29
MULD AC0,AC1 ;MUL 29252X29 ANS IN AC1
STD AC1,@ANS1 ;SAV ANS IN @ANS1
MOV #24,R2 ;SET UP COUNT
STD AC0,AC9 ;PUT 29252 INTO AC9
AAD: ADD AC9,AC0 ;ADD 29252 TO 29252
DEC R2 ;DO 29 TIMES
BNE AAD ;DONE?
STD AC0,@ANS9 ;LOAD TO PRINT
STFPS FPS ;IS ANS CORRECT?
CFCC ;YES
BEQ ,+4 ;EITHER THE ADD OR THE MUL DID NOT WORK
HLT 0 ;LOAD AC3 WITH A 2
LDCID #2,AC3 ;SET UP COUNTER
MOV #13,R2 ;LOAD AC4 WITH A 2
STD AC3,AC4 ;MUL 2 X 2
MMUL: MULD AC4,AC3 ;DO 16 TIMES
DEC R2 ;PUT ANS IN R3
BNE MMUL ;NOW PUT IT INTO @ANS1
STCDI AC3,R3 ;ANS SHOULD BE 10000
MOV R3,@ANS1 ;CONT, IF ANS IS CORRECT
CMP #10000,R3 ;ANS SHOULD BE 10000
STFPS FPS ;LOAD AC2 WITH 10000
BEQ ,+4 ;LOAD AC1 WITH A 2
HLT 1 ;SET UP COUNTER
LDCID #10000,AC2 ;DIVD 2 INTO 65536 16 TIMES
LDCID #2,AC1 ;COUNT NO OF TIMES
MOV #13,R2 ;ARE WE DONE?
STD AC1,AC2 ;STOR ANS INTO R3
DOIV: DIVD AC1,AC2 ;PUT IT INTO @ANS1 FOR TYPING
DEC R2 ;IS ANS CORRECT?
BNE DOIV ;ANS IS CORRECT
STCDI AC2,R3 ;ANS SHOULD EQUAL #2
MOV R2,@ANS1
CMP #2,R3
STFPS FPS
BEQ ,+4
HLT 1

```

1645
(2)
(2)
(2) 010342 104400
1646 010344 170127 047400
1647 010350 177327 000005
1648 010354 012702 001012
1649 010360 177227 000007
1650 010364 174312
1651 010366 173737 001012
1652 010372 170200
1653 010374 170000
1654 010376 001401
1655 010400 104001
1656 010402 171212
1657 010404 012703 000006
1658 010410 173212
1659 010412 005303
1660 010414 001375
1661 010416 175637 001012
1662 010422 022737 000005 001012
1663 010430 170200
1664 010432 001401
1665 010434 104001
1666

```
.....  
|TEST 44      MODE ONE TEST  
|.....  
TST44: SCOPE  
LDFPS      047400  
LDCIP      09,AC3      ;AC309  
MOV        0ANS1,R2   ;R2=ANS1  
LDCIP      07,AC2     ;LOAD 7 INTO AC2  
STP        AC3,(R2)   ;ANS1 SHOULD =9,R2  
CHPF      00ANS1,AC3 ;DOES 0ANS1 = 9  
STFPS      FPS  
CFCC  
BEO        ,04       ;YES BRANCH  
HLT        1         ;ANS1 SHOULD CONTAIN 9  
MULF      (R2),AC2   ;MUL 9 X 7,AC2 = 35  
MOV        06,R3     ;SET UP COUNTER  
SUBFM: SUBF   (R2),AC2 ;SUB 9 FROM 35  
DEC        R3        ;DO 7 TIMES  
BNE        SUBFM  
STCF:     AC2,0ANS1  ;0ANS1 SHOULD =5  
CMP        05,0ANS1 ;DOES 0ANS1 =5  
STFPS      FPS  
BEO        ,04       ;BRANCH IF YES  
HLT        1         ;ANS SHOULD =5  
.....
```

1667
(2)
(2)
(2) 010436 104400
1668 010440 170127 047600
1669 010444 177027 000252
1670 010450 177227 052525
1671 010454 174204
1672 010456 171002
1673 010460 012702 000251
1674 010464 172204
1675 010466 005302
1676 010470 001375
1677 010472 174237 001012
1678 010476 170200
1679 010500 173402
1680 010502 170000
1681 010504 001401
1682 010506 104004

```
.....  
|TEST 45      ADD EXERCISER  
|.....  
TST45: SCOPE  
LDFPS      047600  
LDCID      0252,AC0   ;LOAD AC0 WITH 252  
LDCID      052525,AC2 ;LOAD AC2 WITH 52525  
STD        AC2,AC4   ;AC4=52525  
MULD      AC2,AC0    ;AC0=52525 X 252  
MOV        0251,R2   ;SET UP COUNTER  
AADDH: ADD   AC4,AC2 ;ADD 52525 TO 252  
DEC        R2        ;DO 252 TIMES  
BNE        AADDH     ;DONE?  
STD        AC2,0ANS1 ;GET FOR PRINTING  
STFPS      FPS  
CHPD      AC2,AC0    ;DOES AC2 = AC0?  
CFCC  
BEO        ,04       ;BRANCH IF EQUAL  
HLT        4         ;ANS SHOULD BE.....  
.....
```

```

1684
(2)
(2)
(2) 010510 104400
1685 010512 170127 047600
1686 010516 177027 077777
1687 010522 177127 000000
1688 010526 174401
1689 010530 175437 001012
1690 010534 170200
1691 010536 170337 001032
1692 010542 022737 077777 001012
1693 010550 001401
1694 010552 104001
1695 010554 022737 000004 001032
1696 010562 001401
1697 010564 104000
1698
(2)
(2)
(2) 010566 104400
1699 010570 170127 047400
1700 010574 170011
1701 010576 177027 077777
1702 010602 177127 002525
1703 010606 012702 000012
1704 010612 174401
1705 010614 171001
1706 010616 172001
1707 010620 173001
1708 010622 009302
1709 010624 001372
1710 010626 175437 001012
1711 010632 170200
1712 010634 022737 077777 001012
1713 010642 001401
1714 010644 104001
1715 010646 172437 011102
1716 010652 012702 001012
1717 010656 174022
1718 010660 172537 011102
1719 010664 174122
1720 010666 172737 001022
1721 010672 170200
1722 010674 173737 001012
1723 010700 170000
1724 010702 001401
1725 010704 104010
1726 010706 172537 011104
1727 010712 174142
1728 010714 172012
1729 010716 173042
1730 010720 173001
1731 010722 175437 001012

```

```

.....
;TEST 46 TEST DIVDE BY 0
.....
TST46; SCOPE
LDPPS 047600
LDCID 077777,AC0
LDCID 00,AC1
DIVD AC1,AC0 ;DIVIDE 0 INTO 77777
STCDI AC0,0ANS1 ;LOAD 0ANS1 WITH 77777
STFPS FPS
STST 00FEC ;GET 00FEC STATUS
CMP 077777,0ANS1
BEQ ,04
HLT 1 ;ANS SHOULD =77777
CMP 04,00FEC ;DID WE TRY TO DIV BY 0?
BEQ ,04 ;YES
HLT ;FEC SHOULD =4
.....
;TEST 47 EXERCISER FOR ADD0,SUB0,MUL0 AND DIV0
.....
TST47; SCOPE
LDPPS 047400
SETD ;SET DOUBLE MODE
LDCID 077777,AC0 ;LOAD AC0 WITH 77777
LDCID 02525,AC1 ;LOAD AC1 WITH 2525
MOV 012,R2 ;SET UP COUNTER
EXLOP; DIVD AC1,AC0 ;DIVIDE 2525 INTO 77777
MUL0 AC1,AC0 ;MUL 2525 X ANS
ADD0 AC1,AC0 ;ADD 2525 TO ANS
SUB0 AC1,AC0 ;SUB 2525 FROM ANS
DEC R2 ;DO 12 TIMES
BNE EXLOP ;DONE?
STCDI AC0,0ANS1 ;LOAD ANS INTO 0ANS1
STFPS FPS
CMP 077777,0ANS1 ;IS 0ANS1 CORRECT
BEQ ,04 ;BRANCH IF CORRECT
HLT 1
EX4; LDD 0001010,AC0 ;GET DATA
MOV 0ANS1,R2
STD AC0,(R2)0
LDD 0001010,AC1
STD AC1,(R2)0
LDD 00ANS5,AC3
STFPS FPS
CMPD 00ANS1,AC3
CFCC
BEQ ,04
HLT 0 ;ANS1 AND ANS5 SHOULD =1010
LDD 0000101,AC1 ;STORED AC1 IN ANS5-0
STD AC1,-(R2)
ADD0 (R2),AC0
SUB0 -(R2),AC0
SUB0 AC1,AC0
STCDI AC0,0ANS1

```

1732	010726	022737	000000	001012	CHP	00,00ANS1	
1733	010734	170200			STPPS	FPS	
1734	010736	001401			BEG	,+4	
1735	010740	104001			HLT	1	
1736	010742	170001			SETF		;SET FLOATING MODE
1737	010744	177027	000525		LDCIF	0525,AC0	;LOAD ACP WITH 525
1738	010750	177127	000252		LDCIF	0252,AC1	;LOAD AC1 WITH 252
1739	010754	174104			STP	AC1,AC4	;LOAD AC4 WITH 252
1740	010756	172104			ADDF	AC4,AC1	;ADD 252 TO 252
1741	010760	172701			LDF	AC1,AC3	;PUT ANS IN AC3=924
1742	010762	173003			SUBF	AC3,AC0	;SUB 924 FROM 925
1743	010764	175037	001012		STEXP	AC0,00ANS1	;1 IN ANS1
1744	010770	170200			STPPS	FPS	
1745	010772	022737	000001	001012	CHP	01,00ANS1	;CORRECT ANS SHOULD BE 1
1746	011000	001401			BEG	,+4	
1747	011002	104001			HLT	1	;ANS SHOULD BE 9
1748	011004	177027	000021		LDCIF	021,AC0	;LOAD AC0 WITH 21
1749	011010	171000			MULF	AC0,AC0	;21 TIMES 21 AC0 = 441
1750	011012	174427	040400		DIVF	02,AC0	;DIV BY 2
1751	011016	171427	040200		MODF	01,,AC0	;MUL 1 TIMES 441
1752	011022	170200			STPPS	FPS	;STOR FPS
1753	011024	179537	001012		STCFI	AC1,00ANS1	;PUT ANS IN ANS1
1754	011030	022737	000220	001012	CHP	0220,00ANS1	;IS IT EQUAL?
1755	011036	001401			BEG	,+4	;YES
1756	011040	104001			HLT	1	;SHOULD = 220
1757	011042	171427	041040		MODF	010,,AC0	;GET FRACTION
1758	011046	179537	001012		STCFI	AC1,00ANS1	;PUT AC0 INTO ANS1
1759	011052	022737	000005	001012	CHP	05,00ANS1	;ANS1 SHOULD = 9
1760	011060	001401			BEG	,+4	
1761	011062	104001			HLT	1	;ANS1 SHOULD = 9
1762	011064	104400			SCOPE		
1763	011066	109037	001001		CLRD	00ICNT=1	
1764	011072	000207			RTS	PC	

Line	Code	Value	Code	Value	Code	Value	Code	Value	Code	Value
1766	011074	012701	177777		CHKERR:	MOV	0-1,R1		ISEY FLAG	
1767	011100	000002				RTI			IRETURN	
1768										
1769	011102	177777			D1010:	-1				
1770	011104	000000	177777	000000	D0101:	0,-1,0				
1771	011112	177777	000000	000000	D1001:	-1,0,0				
1772	011120				D0MALL:					
1773	011120	177777	000001	000000	WEIRD:	-1,1,0,-1				
	011126	177777								
1774	011130	011104			ADD101:	D0101				
1775	011132	011112			AD1001:	D1001				
1776	011134	011252			AD1000:	D1000				
1777	011136	077777	000000	177777	D01G:	77777,0,-1,0				
	011144	000000								
1778	011146	100000	000000	000000	DMEERO:	100000,0,0,0				
	011154	000000								
1779	011156	001012			AANS1:	ANS1				
1780	011160	040252	125252	125252	DALTA:	40252,125252,125252,125252				
	011166	125252								
1781	011170	040325	052525	052525	DALTB:	40325,52525,52525,52525				
	011176	052525								
1782	011200	040325	052525	052525	DALTC:	40325,52525,52525,52526				
	011206	052526								
1783	011210	040000	000000	000000	D40:	40000,0,0,0				
	011216	000000								
1784	011220	037400	000000	000000	D37:	37400,0,0,0				
	011226	000000								
1785	011230	040600	000000	000000	D46:	40600,0,0,0				
	011236	000000								
1786	011240	020000	000000	000000	D20:	20000,0,0,0				
	011246	000000								
1787	011250	000000			D0100:	0				
1788	011252	177777	000000	000000	D1000:	-1,0,0,0				
	011260	000000								
1789	011262	000100	000000	000000	D0100X:	100,0,0				
1790	011270	000000	177777	177777	D0111:	0,-1,-1,-1				
	011276	177777								
1791	011300	000000	054321		D5T01:	0,54321				
1792	011304	011300			ADDT01:	D5T01				
1793	011306	043661	121000	000000	F5T01:	43661,121000,0,0				
	011314	000000								
1794	011316	000000			SAVADR:	0				
1795	011320	000000			SAVSTS:	0				

1026	011322	104400			DONEI	SCOPE			
1028	011324	032737	002000	177570		BIT	0SW10,00SWR		;RING THE BELL? 0SW100
1029	011332	001002				BNE	15		;NOI
1010	011334	000004	000007			TYPE	,BELL		
1014									
1015	011340	005046			1SI	CLR	-(6)		;CLEAR TRACE TRAP
1016	011342	032737	010000	177570		BIT	0SW12,00SWR		;RUN WITH TRT? 0SW120
1017	011350	001010				BNE	25		
1018	011352	005167	001256			COM	TRPB		
1019	011356	100005				BPL	25		
1020	011360	052716	000020			BIS	020,(6)		;SET TRACE TRAP
1021	011364	012746	001100			MOV	0BEGIN,-(6)		;JUMP TO START OF TEST
1022	011370	000002				RTI			
1023	011372	012746	001100		2SI	MOV	0BEGIN,-(6)		;JUMP TO START OF TEST
1024	011376	013700	000042			MOV	0042,R0		;RERUN?
1025	011402	001405				BEO	RERUN		
1026	011404	000005				RESET			
1027	011406	004710			LOGICALI	JBR	7,(0)		;JMP TO MONITOR
1028	011410	000240				NOP			
1029	011412	000240				NOP			
1030	011414	000240				NOP			
1031	011416	000002			RERUNI	RTI			
1032									
1036	011420	032737	000400	177570	.ENTI	BIT	0400,00SWR		;KILL LOUB OR LOOP ON SPEC. TEST 0SW00
1037	011426	001404				BEO	15		
1038	011430	126767	166134	167342		CHPB	SWR,ICNT		;ON RIGHT TEST? 0SW7-00
1039	011436	001437				BEO	OVER		
1040	011440	116703	166124		1SI	MOV0	SWR,R3		;GET US BITS
1041	011444	170003				LOUB			
1042	011446	032737	040000	177570		BIT	0SW14,00SWR		;LOOP ON TEST 0SW140
1043	011454	001026				BNE	KIT		
1044	011456	032737	004000	177570		BIT	0SW11,00SWR		;KILL ITERATIONS 0SW110
1045	011464	001012				BNE	SAVLAD		
1046	011466	105767	167307			TST0	ICNT-1		
1048	011472	001404				BEO	25		;BRANCH IF FIRST
1049	011474	126767	001136	167277		CHPB	TIMES,ICNT-1		;DONE?
1050	011502	001013				BNE	KIT		;BRANCH IF NOT
1051	011504	112767	000001	167267	2SI	MOV0	01,ICNT-1		;FIRST ITERATION
1052	011512	105267	167262		SAVLAD:	INCB	ICNT		;COUNT TEST NUMBERS
1053	011516	011667	167266			MOV	(6),LAD		;SAVE LOOP ADDRESS
1054	011522	016737	167252	177570		MOV	ICNT,00SWR		;DISPLAY TEST NO. AND ITERATION COUNT
1055	011530	000002				RTI			;RETURN
1056									
1057	011532	105267	167243		KIT:	INCB	ICNT-1		
1059	011536	016737	167236	177570	OVER:	MOV	ICNT,00SWR		;SET UP DISPLAY
1060	011544	005767	167240			TST	LAD		;FIRST ONE?
1064	011550	001760				BEO	SAVLAD		
1065	011552	016716	167232			MOV	LAD,(6)		;FUDGE RETURN ADDRESS
1066	011556	000002				RTI			;FIXES PS

1876	011560	032737	002000	177570	,TRPI	BIT	0SW10,0,SHR	;BELL ON ERROR?
1877	011566	001402				BEO	15	;NO - SKIP
1878	011570	000004	000007			TYPE	,BELL	;RING BELL
1879	011574	004767	000444		15I	JSR	PC,ERROR	;COUNT THE NUMBER OF ERRORS
1880	011600	010446				MOV	R4,-(6)	
1881	011602	032737	020000	177570		BIT	0SW13,0,SHR	;SKIP TYPEOUT IF SET
1882	011610	001074				BNE	45	
1883	011612	000004	011616			TYPE	,,02	;ASCIZ <19><12>
1884	011622	016646	000002			MOV	2(6),-(6)	;PUT ADDRESS OF INSTRUCTION ON STACK
1885	011626	102716	000002			SUB	02,(6)	
1886	011632	011605				MOV	(6),TTY	;TYPE (6) IN OCTAL
(1)	011634	004767	000222			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
1887	011640	000004	012617			TYPE	,SPACE*3	
1888	011644	010005				MOV	R0,TTY	;TYPE R0 IN OCTAL
(1)	011646	004767	000210			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
1889	011652	000004	012620			TYPE	,SPACE*4	
1890	011656	012703	001012			MOV	0ANS1,R3	;ADDRESS OF DATA
1891	011662	113604				MOV0	0(6)*,R4	;AMOUNT OF DATA IN TABLE
1892	011664	001426				BEO	35	
1893	011666	100016				BPL	25	;TYPE STACK?
1894	011670	016667	000006	167114		MOV	6(6),ANS1	
1895	011676	016667	000010	167110		MOV	10(6),ANS2	
1896	011704	016667	000012	167104		MOV	12(6),ANS3	
1897	011712	016667	000014	167100		MOV	14(6),ANS4	
1898	011720	042704	177600			BIC	0177600,R4	;CLEAR SIGN
1899	011724	000004	012620		25I	TYPE	,SPACE*4	
1900	011730	012305				MOV	(3)*,TTY	;TYPE (3)* IN OCTAL
(1)	011732	004767	000124			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
1901	011736	005304				DEC	R4	
1902	011740	001371				BNE	25	
1903	011742	005700			35I	TST	FPS	
1904	011744	100016				BPL	45	
1905	011746	000004	012614			TYPE	,SPACE	
1906	011752	170337	001032			STST	00FEC	
1907	011756	013705	001032			MOV	00FEC,TTY	;TYPE 00FEC IN OCTAL
(1)	011762	004767	000074			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
1908	011766	000004	012617			TYPE	,SPACE*3	
1909	011772	016705	167036			MOV	FEA,TTY	;TYPE FEA IN OCTAL
(1)	011776	004767	000060			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
1910	012002	012604			45I	MOV	(6)*,R4	
1911	012004	005737	177570			TST	00SHR	;HALT ON ERROR
1912	012010	100001				BPL	,*4	;SKIP IF CONTINUE
1913	012012	000000				HALT		;HALT ON ERROR!
1914	012014	032737	001000	177570		BIT	0SW09,0,SHR	;CHECK FOR INHIBIT LOOP ON ERROR
1915	012022	001001				BNE	,*4	;SKIP IF LOOP ON ERROR
1916	012024	000002				RTI		
1917	012026	105067	166747			CLRB	ICNT*1	
1918	012032	032737	000400	177570		BIT	0SW08,0,SHR	;CHECK FOR LOAD MICROBREAK
1919	012040	001234				BNE	KIT	;BRANCH IF NOT
1920	012042	113703	177570			MOV0	00SHR,R3	;PUT MICROBREAK ADDRESS IN R3
1921	012046	170003				LDUB		;LOAD MICROBREAK
1922	012050	000630				BR	KIT	;LOOP ON TEST UNTIL NO ERRORS

```

1924          ;          SOCIAL          OCTAL TYPEOUT ROUTINE
(1)
(1)          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
(1)          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 10 BIT ADDRESS, OR TYPE
(1)          ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP10, OR BITYPE MACRO'S.
(1)          ;
(1) 012052 012767 170101 000140 BITYPS: MOV      0170101,,PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
(1) 012060 000411          BR          ,PTIT          ;NOW TYPE IT IN BIT FORM
(1) 012062 112767 000001 000130 PRINTR: MOVB   01,,PR          ;SET ZERO FILL SWITCH
(1) 012070 000402          BR          ,06          ;SKIP
(1) 012072 005067 000122          PRINTS: CLR      ,PR          ;SUPPRESS LEADING ZERO'S
(1) 012076 112767 177772 000119          MOVB   0-6,,PR+1      ;SET COUNT
(1) 012104 010446          .PTIT:  MOV      R4,-(0)          ;SAVE R4
(1) 012106 012704 012222          MOV      0,PR+2,R4        ;SET POINTER TO FIRST ASCII CHAR.
(1) 012112 105014          CLR      (4)          ;CLEAR FIRST BYTE
(1) 012114 000411          BR          ,PRF          ;ROTATE FIRST BIT
(1) 012116 105014          .PRLI  CLR      (4)          ;CLEAR BYTE OF CHARACTER
(1) 012120 032767 000100 000072          BIT      0100,,PR        ;BIT TYPING MODE?
(1) 012126 001004          BNE     ,PRF          ;YES - SKIP 2 ROTATES
(1) 012130 006105          ROL     TTY          ;ROTATE BIT INTO C
(1) 012132 106114          ROL     (4)          ;PACK IT
(1) 012134 006105          ROL     TTY          ;ROTATE BIT INTO C
(1) 012136 106114          ROL     (4)          ;PACK IT
(1) 012140 006105          .PRFI  ROL     TTY          ;ROTATE BIT INTO C
(1) 012142 106114          ROL     (4)          ;PACK IT
(1) 012144 105714          TSTB   (4)          ;IS IT ZERO?
(1) 012146 001402          BEQ    ,06          ;SKIP INC
(1) 012150 105267 000044          INCB   ,PR          ;SET FILL SWITCH
(1) 012154 105767 000040          TSTB   ,PR          ;CHECK FILL SWITCH
(1) 012160 001402          BEQ    ,06          ;SKIP BITSET
(1) 012162 102724 000060          DISB   010,(4)0     ;MAKE INTO ASCII CHAR
(1) 012166 105267 000027          INCB   ,PR+1        ;INC COUNT
(1) 012172 001391          BNE     ,PRL        ;REPEAT
(1) 012174 022704 012222          CMP    0,PR+2,R4    ;EMPTY BUFFER?
(1) 012200 001002          BNE     ,06          ;SKIP IF NOT
(1) 012202 112724 000060          MOVB   010,(4)0     ;LOAD 1 ZERO
(1) 012206 105014          CLR      (4)          ;NULL TERMINATOR
(1) 012210 000004 012222          TYPE   ,,PR+2        ;TYPE IT
(1) 012214 012604          MOV      (0),R4      ;RESTORE R4
(1) 012216 000207          RTS     PC          ;RETURN
(1) 012220 000012          .PRI   ,BLKW      12 ;COUNT, SWITCH, AND OUTPUT BUFFER
1925
1926 012244 005267 166532          ERROR: INC      ERRORS          ;COUNT ERRORS
1927 012250 132737 000001 000041          BITB   01,0041      ;AUTO MODE?
1928 012256 001412          BEQ    15          ;NO!
1929 012260 022767 000010 166514          CMP    010,ERRORS   ;TOO MANY?
1930 012266 001004          BNE     15          ;NOT YET
1931 012270 013700 000042          MOV    0042,R0      ;GET ADDRESS
1932 012274 001403          BEQ    15          ;FORGET IT IF ZERO
1933 012276 005037 000042          CLR    0042         ;EAP 42
1934 012302 004710          JSR    PC,(0)       ;CALL THE MONITOR
1935 012304 000207          15:    RTS     PC          ;RETURN

```

1940	012306	012777	012502	000314	POWDN:	MOV	0ILLUP,0UPVEC	;SET FOR FAST UP
1941	012314	012777	000340	000310		MOV	0340,0UPVEC+2	;PRI017
1942	012322	170246				STFPS	=(0)	;GET THE FPS
1943	012324	170011				SETD		
1944	012326	174046				STD	AC0,=(0)	;SAVE AC'S
1945	012330	174146				STD	AC1,=(0)	
1946	012332	174246				STD	AC2,=(0)	
1947	012334	174346				STD	AC3,=(0)	
1948	012336	172404				LDD	AC4,AC0	
1949	012340	174046				STD	AC0,=(0)	
1950	012342	172405				LDD	AC5,AC0	
1951	012344	174046				STD	AC0,=(0)	
1952	012346	010046				MOV	R0,=(0)	;SAVE REGISTERS
1953	012350	010146				MOV	R1,=(0)	
1954	012352	010246				MOV	R2,=(0)	
1955	012354	010346				MOV	R3,=(0)	
1956	012356	010446				MOV	R4,=(0)	
1957	012360	010546				MOV	R5,=(0)	
1958	012362	010667	000234			MOV	SP,SAVE6	;SAVE SP
1959	012366	012777	012376	000234		MOV	0PONUP,0UPVEC	;SET UP VECTOR
1960	012374	000000				HALT		
1961								
1962	012376	016706	000220		POWUP:	MOV	SAVE6,SP	;GET SP
1963	012402	005001				CLR	R1	;WAIT LOOP FOR THE TTY
1964	012404	005201			10:	INC	R1	
1965	012406	001376				BNE	10	
1966	012410	012605				MOV	(0)+,R5	;GET THE REGISTERS
1967	012412	012604				MOV	(0)+,R4	
1968	012414	012603				MOV	(0)+,R3	
1969	012416	012602				MOV	(0)+,R2	
1970	012420	012601				MOV	(0)+,R1	
1971	012422	012600				MOV	(0)+,R0	
1972	012424	170011				SETD		
1973	012426	172426				LDD	(0)+,AC0	;RESTORE THE AC'S
1974	012430	174005				STD	AC0,AC0	
1975	012432	172426				LDD	(0)+,AC0	
1976	012434	174004				STD	AC0,AC4	
1977	012436	172726				LDD	(0)+,AC3	
1978	012440	172626				LDD	(0)+,AC2	
1979	012442	172526				LDD	(0)+,AC1	
1980	012444	172426				LDD	(0)+,AC0	
1981	012446	170126				LDFPS	(0)+	;RESTORE FPS
1982	012450	012777	012306	000146		MOV	0PONDOWN,0DOWNVEC	;SET UP THE POWER DOWN VECTOR
1983	012456	012777	000340	000142		MOV	0340,0DOWNVEC+2	
1984	012464	000004	012470			TYPE	,,02	;ASCIZ <13><12>"POWER"
1985	012500	000002				RTI		
1986								
1987	012502	000000			ILLUP:	HALT		;THE POWER UP SEQUENCE WAS STARTED
1988	012504	000776				OR	,-2	;BEFORE THE POWER DOWN WAS COMPLETE

```

1998      )      STYPE      MESSAGE TYPEOUT ROUTINE
(1)
(1)      )THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
(1)      )CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
(1)      )MESSAGE STARTIN IN LOCATION "ADR", 2) "TYPE ,CHAR" - TYPES
(1)      )THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE" - TYPES
(1)      )THE MESSAGE WHICH IS INLINE ASCII.
(1)
(1)      012506 010546      IOTS:  MOV      TTY, -(6)      )SAVE TTY
(1)      012510 017605 000002      MOV      @2(6), TTY      )GET ADDRESS TO BE TYPED
(1)      012514 032705 177400      BIT      @177400, TTY      )IS IT A TYPEN?
(1)      012520 001004      BNE      IS      )NO
(1)      012522 010567 000064      MOV      TTY, ,TYPE      )GET THE CHARACTER
(1)      012526 012705 012612      MOV      @,TYPE, TTY      )JUDGE THE ADDRESS
(1)      012532 105715      ISI      TSTB      (TTY)      )TERMINATOR?
(1)      012534 001406      BEQ      ZS      )GET OUT IF SO
(1)      012536 112537 177566      MOVB     (TTY)+, @0177566 )LOAD AND TYPE THE CHARACTER
(1)      012542 105737 177564      TSTB     @0177564      )IS THE PRINTER READY
(1)      012546 100375      SPL      ,=4      )WAIT UNTIL IT IS
(1)      012550 000770      BR       IS      )GET THE NEXT CHARACTER
(1)      012552 017646 000002      2SI      MOV      @2(6), -(6)      )GET ADDRESS TO BE TYPED
(1)      012556 062766 000002 000004      ADD      @2, 4(6)      )ADD 2 TO THE ADDRESS
(1)      012564 022666 000002      CMP      (6)+, 2(6)      )IS IT ,=2?
(1)      012570 001006      BNE      3S      )NO
(1)      012572 062705 000002      ADD      @2, TTY      )ADD 2 TO THE ADDRESS
(1)      012576 042705 000001      BIC      @1, TTY      )BACK UP TO AN EVEN BYTE
(1)      012602 010566 000002      MOV      TTY, 2(6)      )RESTORE ADDRESS
(1)      012606 012605      3SI      MOV      (6)+, !TY      )RESTORE TTY
(1)      012610 000002      RTI      )RETURN
(1)      012612 000000      )CHARACTER TYPE LOCATION
1991      012614 005015 020040 000040      )TYPE) 0
1992      012622 000000      SPACE) ,ASCII <15><12>" "
1993      012624 000024 000026      SAVE6) 0
1994      012630 000024 000026      DWNVEC) 24,26      )POWER DOWN VECTOR ADDRESS
1996      012634 000000      UPVEC) 24,26      )POWER UP VECTOR ADDRESS
1997      012636 000377      TRPB) 0
2000      012640 172160      TIMES) 377      )ITERATION COUNT
2001      012642 000244 000246      FPTADR) 172160      )FLOATING POINT ADDRESS ON THE 11/20
2002      )FPVECT) 244,246      )FLOATING POINT VECTOR ADDRESS
2016
2018      000001      )END

```

AAD	210212	AADDH	010464	AANS1	011150	AC0	=X000000
AC1	=X000001	AC2	=X000002	AC3	=X000003	AC4	=X000004
AC5	=X000005	ADANS1	002136	ADP1R1	011130	AD1000	011134
AD1021	011132	AD9T01	011304	ANS1	001012	ANS2	001014
ANS3	001016	ANS4	001020	ANS5	001022	ANS0	001024
ANS7	001026	ANS0	001030	BEGIN	001100	BELL	= 000007
BITYPS	012652	CHKERR	011074	CODE	010000	DALTA	011160
DALTE	011170	DALYC	011200	DBIG	011130	DDIV	010314
DMZERO	011146	DONE	011322	DSMALL	011120	DHNVEC	012624
DB102	011250	DB100X	011262	DB101	011104	DB111	011270
D1202	011252	D1001	011112	D1010	011102	D20	011240
D37	011220	D40	011210	D46	011230	D9T01	011300
ERROR	012244	ERRORS	001002	EXLOP	010012	EX4	010646
FEA	001034	FEC	001032	FLTERR	000760	FPS	=X000000
FPTACR	012640	FPVECT	012642	FST01	011300	ICNT	001000
ILLUP	012502	IOTS	012506	KIT	011532	LAD	001010
LDSC	= 170004	LDUB	= 170003	LOGICA	011400	MHUL	010252
MORE	010742	MOVE	006710	MRS	= 170006	M120	001072
N	= 200050	OVER	011536	PC	=X000007	PCNT	001004
POWDN	012306	POWUP	012376	PRINTR	012062	PRINTS	012072
PS	= 177776	RELEX	006564	RERUN	011410	RETURN	=X000001
RE	=X000000	R1	=X000001	R2	=X000002	R3	=X000003
R4	=X000004	R5	=X000005	SAVADR	011310	SAVE6	012622
SAVLAD	011512	SAVSTS	011320	SP	=X000006	SPACE	012614
START	001036	STAB	= 170005	ST00	= 170007	SUBFM	010410
SUBT	010136	SWR	= 177970	SW00	= 000400	SW09	= 001000
SW10	= 002000	SW11	= 004000	SW12	= 010000	SW13	= 020000
SW14	= 040000	TIMES	012636	TRAP4	006672	TRPB	012634
TST1	001172	TST10	002300	TST11	002424	TST12	002600
TST13	003004	TST14	003174	TST15	003422	TST16	003524
TST17	003666	TST2	001310	TST20	003770	TST21	004072
TST22	004310	TST23	004424	TST24	004660	TST25	005104
TST26	005302	TST27	005444	TST3	001414	TST30	005556
TST31	005654	TST32	005746	TST33	006034	TST34	006122
TST35	006206	TST36	006326	TST37	006424	TST4	001000
TST40	010006	TST41	010050	TST42	010100	TST43	010164
TST44	010342	TST45	010436	TST46	010510	TST47	010566
TST5	001722	TST6	002044	TST7	002224	TTY	=X000005
TYPE	= 000004	UPVEC	012630	WEIRD	011120	X2X	006676
,BIT	= 176777	,ENT	011420	,PR	012220	,PRF	012140
,PRL	012116	,PTIT	012104	,TRP	011562	,TYPE	012612
,	= 212646						

ERRORS DETECTED: 0

•DCFPD,DCFPD•DCFPD.P11
 RUN-TIME: 13 19 0 SECONDS
 CORE USED: 0K