

# FP11

MAINTENANCE INSTRUCTION  
**MD-11-DCFPM-B**  
TESTS

EP-DCFPM-B-DL  
COPYRIGHT 72-73  
FICHE 1 OF 1

AUG 1978  
**digital**  
MADE IN USA

The microfiche strip contains 15 frames of technical data. Each frame typically includes a title, a table of test parameters, and a list of test results. The data is organized into columns and rows, with some frames containing sub-sections or diagrams. The text is small and difficult to read due to the resolution of the scan.



.REPT 2

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

IDENTIFICATION  
-----

PRODUCT CODE: MAI DEC-11-DCFPM-B-D  
PRODUCT NAME: FP11 MAINTENANCE INSTRUCTION TESTS  
DATE CREATED: NOVEMBER 1973  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: BOB BRAIN

COPYRIGHT (c) 1972, 1973  
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.

45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94

MAINDEC-11-DCFPM-B-D FP11 MAINTENANCE TEST PAGE 2  
TABLE OF CONTENTS

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

MAINDEC-11-DCFPM-B-D FP11 MAINTENANCE TEST PAGE 3  
DESCRIPTION

95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148

1. ABSTRACT  
THE PURPOSE OF THIS PROGRAM IS THREE FOLD. FIRST TO TEST THE MAINTENANCE FEATURES, SECOND TO TRACE MULD THROUGH ITS STEPS, THIRD TO TRACE DIVD.
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
    - 4.3.1 BASIC MAINTENANCE TEST  
1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.

149  
150  
151  
152

- 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

153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206

MAINDEC-11-DCFPM-B-D FP11 MAINTENANCE TEST PAGE 4  
DESCRIPTION

6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.  
7) THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION CCUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

4.3.2 MULTIPLY TRACE (MUST RUN 1 PASS OF BASIC TEST)

- 1) LOAD ADDRESS 204
- 2) START
- 3) TYPE FIRST NUMBER (MUST BE IN OCTAL, 4 WORDS SEPERATED BY SPACES, FOLLOWED BY A RETURN)
- 4) TYPE SECOND NUMBER (LIKE FIRST)
- 5) THE PROGRAM WILL TRACE THE 71 STEPS

4.3.3 DIVIDE TRACE

SAME AS 4.3.2 EXCEPT START AT 210

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 .. ALL SWITCHES DOWN IS WORST CASE TESTING. IF AN ERROR OCCURS, THAT TEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF THE SUBTEST IF SW<9> SET TO A 1. 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 ..... LOOP ON ERROR
- SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>
- 0 ..... LOAD SW<7:0> INTO UB REGISTER

5.2 SUBROUTINE ABSTRACTS

237  
238  
239  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260

MAINDEC-11-DCFPM-B-D FP11 MAINTENANCE TEST PAGE 5  
DESCRIPTION

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.) IF A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 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 REACHED.

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)

THE FP11 INTERRUPT DISABLE BIT IS ALWAYS SET IN ALL OF THESE TESTS SO 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.

6. ERRORS

6.1 ERROR PRINTOUT

261  
262  
263  
264

THE FORMAT FOR THE BASIC TEST IS AS FOLLOWS:

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

265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318

MAINDEC-11-DCFPM-B-D FP11 MAINTENANCE TEST PAGE 6  
DESCRIPTION

WHERE:

ADR = ADDRESS OF ERROR WLT  
FPS = FLOATING POINT STATUS  
FEC = 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  
WLT; I.E., WLT+3 WOULD TYPE ANS1-ANS3.

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

THE TRACE TYPEOUT IS SIMILAR EXCEPT FPS IS THE COUNT (1 -  
71), ANS1 - ANS4 ARE THE AR, AND ANS5 - ANS8 ARE THE CR.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 15 SECONDS WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600

8.3 POWER FAIL

EACH 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

319  
320  
321  
322

THIS PROGRAM TESTS THE MAINTENANCE INSTRUCTIONS OF THE FP11.  
IT HAS MANY SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS)  
WHICH ARE RUN 256 TIMES BEFORE CONTINUING TO THE NEXT.

323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342

MAINDEC-11-DCFPM-B-D  
DESCRIPTION

FP11 MAINTENANCE TEST

PAGE 7

SW<11> ON A 1 CAUSES EACH SUBTEST TO BE RUN ONLY ONCE.  
SW<9> ON A 1 ENABLES LOOP ON ERROR. THE ADDRESS ICNT (LOC  
1000) AND DISPLAY REGISTER ON THE 11/45 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.

STARTING AT 204 OR 210 WILL TRACE THE AR AND OR THROUGH THE  
MULD OR DIVD INSTRUCTIONS.

.ENDR

343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383

.TITLE MAINDEC-11-DCFPM-B MAINTENANCE INSTRUCTION TEST  
PROGRAM BY BOB BRAIN  
.REM\*

SWITCH	USE
8	LOOP ON TEST IN SW(718)
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 FORM:

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*			

384			.ENABL	ABS	
385	000001		N=	1	
386	177776		PS=	177776	
387	177570		SWR=	177570	
388	177570		DISPLAY=	SWR	
389	104400		SCOPE=	TRAP	
390	104000		HLT=	EMT	
391	000004		TYPE=	IOT	
392	000007		BELL=	7	
393	000000		FPS=	X0	
394	000000		R0=	X0	
395	000001		R1=	X1	
396	000002		R2=	X2	
397	000003		R3=	X3	
398	000004		R4=	X4	
399	000005		R5=	X5	
400	000005		TTY=	X5	
401	000006		SP=	X6	
402	000007		PC=	X7	
403	000000		AC0=	X0	
404	000001		AC1=	X1	
405	000002		AC2=	X2	
406	000003		AC3=	X3	
407	000004		AC4=	X4	
408	000005		AC5=	X5	
409	100000		SW15=	100000	
410	040000		SW14=	40000	
411	020000		SW13=	20000	
412	010000		SW12=	10000	
413	004000		SW11=	4000	
414	002000		SW10=	2000	
415	001000		SW09=	1000	
416	000400		SW08=	400	
417	170003		LDUB=	170003	
418	170005		STAB=	170005	
419	170007		STOB=	170007	
420	170006		MRS=	170006	
421	170004		LOSC=	170004	
422					
423	000000		.S	0	
424	000200		.S	200	
425	000200	000622	JMP	BEG	
426	000204	003234	JMP	TSTMUL	
427	000210	003464	JMP	TSTDIV	
428					
429	000760		.S	760	
430	000760	170200	FLTRR:	STFPS	FPS
431	000762	170367		STST	FEC
432	000766	000000		HALT	
433	000770	000002		RTI	

;TRAP CATCHER FROM 0 - 770

```

434          001000          . =      1000
435
436 001000 000000          ICNT: 0          ; ITERATION COUNT = LH TEST NO. = RH
437 001002 000000          ANS1: 0          ; FIRST ANSWER (SEE CODE)
438 001004 000000          ANS2: 0
439 001006 000000          ANS3: 0
440 001010 000000          ANS4: 0
441 001012 000000          ANS5: 0
442 001014 000000          ANS6: 0
443 001016 000000          ANS7: 0
444 001020 000000          ANS8: 0
445 001022 000000          FEC: 0          ; FLOATING EXCEPTION CODES
446 001024 000000          FEA: 0          ; FLOATING EXECPTION ADDRESS
447
448 001026 012706 000600      BEG:  MOV  0600,SP          ; 00 STACK AT 000 00
449 001032 012737 001054 000004      MOV  0M1120,004          ; FIND OUT WHICH MACHINE THIS IS
450 001040 005737 177772          TST  00177772          ; IS PIRQ THERE?
451 001044 012767 000006 003476      MOV  06,YESRT          ; FUDGE IN RTT IF 11/49
452 001052 000403
453
454 001054 016737 004612 000010      M1120: MOV  FPTADR,0010          ; LOAD THE ILLEGAL INSTRUCTION VECTOR
455                                     ; WITH THE ADDRESS OF THE FPU,
456                                     ; THE FPU WILL HANDLE THE BAD OPCODES
457 001062 012737 000006 000004      BEGIN: MOV  06,004          ; RESET 4
458 001070 012706 000600      MOV  0600,SP
459 001074 012737 004550 000014      MOV  0YESRT,0014          ; SET TRACE TRAP VECTOR
460 001102 012777 005370 004570      MOV  0PONDWN,0DMNVEC
461 001110 012777 000340 004564      MOV  0340,0DMNVEC+2
462 001116 012737 005570 000020      MOV  0,IOT,0020          ; SET UP VECTOR 20
463 001124 012700 000030      MOV  030,R0          ; SET R0 TO VECTOR 30
464 001130 012720 004704      MOV  0,TRP,(0)+          ; SET EMT VECTOR
465 001134 012720 000340      MOV  0340,(0)+
466 001140 012720 004552      MOV  0,EMT,(0)+          ; SET TRAP VECTOR
467 001144 012710 000340      MOV  0340,(0)
468 001150 012777 000760 004516      MOV  0FLTERR,0FPVECT          ; LOAD INTERRUPT VECTOR
469 001156 012777 000340 004512      MOV  0340,0FPVECT+2          ; LOCK UP PROCESSOR
470 001164 005067 177610      CLR  ICNT
471 001170 005067 004520      CLR  LAD
472 001174 170127 040000      LDPPS 040000
473 001200 005003      CLR  R3          ; CLEAR
474 001202 170003      LDUB          ; BREAK REG

```

```

475 ;.....
476 ;TEST 1          TRAP BEFORE LOAD [5]
477 ;.....
478 TST1:  SCOPE
479
480 001204 104400          LDFPS  #40220          ;DOUBLE/MAINT BITS
481 001212 172467 004066  LDD    ALT1A,AC0      ;LOAD ALT 1'S
482 001216 012703 000005  MOV    #5,R3          ;LOAD 5 INTO
483 001222 170003          LDUB   ;UBREAK REG.
484 001224 172467 004064  LDD    ALT1B,AC0      ;LOAD OTHER 1'S
485 001230 170200          STFPS  FPS            ;STORE FLOATING POINT STATUS
486 001232 170367 177564  STST   FEC            ;STORE EXCEPTION CODES
487 001236 022700 140230  CMP    #140230,FPS    ;CHECK FLOATING POINT STATUS
488 001242 001401          BEQ    ,+4            ;BRANCH IF OK
489 001244 104000          HLT
490
491 001246 022767 000016 177546  CMP    #16,FEC        ;CHECK FLOATING EXCEPTION CODE
492 001254 001401          BEQ    ,+4            ;BRANCH IF OK
493 001256 104000          HLT                    ;FEC NOT EQUAL TO 16
494
495 001260 174067 177516  STD    AC0,ANS1       ;GET RESULT
496 001264 173467 004014  CMPD   ALT1A,AC0      ;CHECK RESULT
497 001270 170000          CFCC   ;GET CC BITS
498 001272 001401          BEQ    ,+4            ;SKIP IF OK
499 001274 104004          HLT+4                 ;SECOND LDD DIDN'T TRAP
500
501 001276 005003          CLR    R3             ;CLEAR
502 001300 170003          LDUB   ;UBREAK REG
503
504 ;.....
505 ;TEST 2          TRAP AFTER LOADING 32 BITS [21]
506 ;.....
507 TST2:  SCOPE
508
509 001304 170127 040220          LDFPS  #40220          ;DOUBLE/MAINT BITS
510 001310 012703 000021  MOV    #21,R3         ;LOAD 21 INTO
511 001314 170003          LDUB   ;UBREAK REG.
512 001316 172467 003772  LDD    ALT1B,AC0      ;LOAD FIRST 32 BITS
513 001322 170200          STFPS  FPS            ;STORE FLOATING POINT STATUS
514 001324 170367 177472  STST   FEC            ;STORE EXCEPTION CODES
515 001330 022700 140230  CMP    #140230,FPS    ;CHECK FLOATING POINT STATUS
516 001334 001401          BEQ    ,+4            ;BRANCH IF OK
517 001336 104000          HLT                    ;FEC NOT EQUAL TO 140230
518
519 001340 022767 000016 177454  CMP    #16,FEC        ;CHECK FLOATING EXCEPTION CODE
520 001346 001401          BEQ    ,+4            ;BRANCH IF OK
521 001350 104000          HLT                    ;FEC NOT EQUAL TO 16
522
523 001352 174067 177424          STD    AC0,ANS1       ;GET ANSWER
524 001356 173467 003736  CMPD   ALT1C,AC0      ;RESULT RIGHT?
525 001362 170000          CFCC   ;GET CC
526 001364 001401          BEQ    ,+4            ;SKIP IF OK
527 001366 104004          HLT+4                 ;WEIRD ANSWER
528

```

```

529 001370 005003          CLR      R3          ;CLEAR
530 001372 170003          LOUB     ;UBREAK REG
531
532
533          ;.....
533          ;TEST 3          TEST OF STAB AND STQB DOUBLE MODE
534          ;.....
535 001374 104400          TST3:   SCOPE
536
537 001376 170127 040220    LOFPS   #40220      ;DOUBLE/MAINT BITS
538 001402 172467 003722    LOD     ALTA,ACB   ;LOAD ALT 1'S
539 001406 012703 000115    MOV     #115,R3   ;LOAD 115 INTO
540 001412 170003          LOUB     ;UBREAK REG.
541 001414 172067 003720    ADD     ALTB,ACB  ;ADD ALT 1'S
542 001420 170200          STFPS   FPS       ;STORE FLOATING POINT STATUS
543 001422 170367 177374    STST   FEC       ;STORE EXCEPTION CODES
544 001426 022700 140220    CMP     #140220,FPS ;CHECK FLOATING POINT STATUS
545 001432 001401          BEO     ,+4       ;BRANCH IF OK
546 001434 104000          HLT
547
548 001436 022767 000016 177356    CMP     #16,FEC   ;CHECK FLOATING EXCEPTION CODE
549 001444 001401          BEO     ,+4       ;BRANCH IF OK
550 001446 104000          HLT
551
552 001450 170005          STAB
553 001452 174067 177324    STD     ACB,ANS1  ;GET AR INTO ACB
554 001456 042767 177600 177316    BIC     #177600,ANS1 ;GET FOR TYPING
555 001464 022767 000125 177310    CMP     #125,ANS1 ;CLEAR JUNK
556 001472 001401          BEO     ,+4       ;IS ANS1 = 125
557 001474 104004          HLT+4    ;SKIP IF SAME
558 001476 022767 052525 177300    CMP     #52525,ANS2 ;ANS1 NOT = 125
559 001504 001401          BEO     ,+4       ;IS ANS2 = 52525
560 001506 104004          HLT+4    ;SKIP IF SAME
561 001510 022767 052525 177270    CMP     #52525,ANS3 ;ANS2 NOT = 52525
562 001516 001401          BEO     ,+4       ;IS ANS3 = 52525
563 001520 104004          HLT+4    ;SKIP IF SAME
564 001522 022767 052525 177260    CMP     #52525,ANS4 ;ANS3 NOT = 52525
565 001530 001401          BEO     ,+4       ;IS ANS4 = 52525
566 001532 104004          HLT+4    ;SKIP IF SAME
567
568 001534 170007          STQB
569 001536 174067 177240    STD     ACB,ANS1  ;GET OR
570 001542 042767 177600 177232    BIC     #177600,ANS1 ;GET RESULT
571 001550 022767 000052 177224    CMP     #52,ANS1  ;CLEAR JUNK
572 001556 001401          BEO     ,+4       ;IS ANS1 = 52
573 001560 104004          HLT+4    ;SKIP IF SAME
574 001562 022767 125252 177214    CMP     #125252,ANS2 ;ANS1 NOT = 52
575 001570 001401          BEO     ,+4       ;IS ANS2 = 125252
576 001572 104004          HLT+4    ;SKIP IF SAME
577 001574 022767 125252 177204    CMP     #125252,ANS3 ;ANS2 NOT = 125252
578 001602 001401          BEO     ,+4       ;IS ANS3 = 125252
579 001604 104004          HLT+4    ;SKIP IF SAME
580 001606 022767 125252 177174    CMP     #125252,ANS4 ;ANS3 NOT = 125252
581 001614 001401          BEO     ,+4       ;IS ANS4 = 125252
582 001616 104004          HLT+4    ;SKIP IF SAME
583

```

```

583
584 001620 005003          CLR      R3          ;CLEAR
585 001622 170003          LDUB                     ;UBREAK REG.
586
587 ;.....
588 ;TEST 4          TEST OF STAB AND STQB FLOATING MODE
589 ;.....
590 001624 104400          TST4:  SCOPE
591
592 001626 170127 040020          LDFPS   #40020          ;FLOATING/MAINT BITS
593 001632 172467 003502          LDF     ALTB,ACB        ;LOAD ALT 1'S
594 001636 012703 000115          MOV     #115,R3        ;LOAD 115 INTO
595 001642 170003          LDUB                     ;UBREAK REG.
596 001644 172067 003460          ADDF   ALTA,ACB        ;ADD ALT 1'S
597 001650 170200          STFPS  FPS             ;STORE FLOATING POINT STATUS
598 001652 170367 177144          STST   FEC             ;STORE EXCEPTION CODES
599 001656 022700 140020          CMP     #140020,FPS     ;CHECK FLOATING POINT STATUS
600 001662 001401          BEQ    ,+4             ;BRANCH IF OK
601 001664 104000          HLT
602
603 001666 022767 000016 177126          CMP     #16,FEC        ;CHECK FLOATING EXCEPTION CODE
604 001674 001401          BEQ    ,+4             ;BRANCH IF OK
605 001676 104000          HLT
606
607 001700 170007          STQB
608 001702 174067 177074          STF     ACB,ANS1        ;GET OR INTO ACB
609 001706 042767 177600 177066          BIC     #177600,ANS1    ;GET FOR TYPING
610 001714 022767 000125 177060          CMP     #125,ANS1      ;CLEAR JUNK
611 001722 001401          BEQ    ,+4             ;IS ANS1 = 125
612 001724 104004          HLT+4                    ;SKIP IF SAME
613 001726 022767 052525 177050          CMP     #52525,ANS2    ;ANS1 NOT = 125
614 001734 001401          BEQ    ,+4             ;IS ANS2 = 52525
615 001736 104004          HLT+4                    ;SKIP IF SAME
616 001740 022767 125252 177040          CMP     #125252,ANS3   ;ANS2 NOT = 52525
617 001746 001401          BEQ    ,+4             ;IS ANS3 = 125252
618 001750 104004          HLT+4                    ;SKIP IF SAME
619 001752 022767 125252 177030          CMP     #125252,ANS4   ;ANS3 NOT = 125252
620 001760 001401          BEQ    ,+4             ;IS ANS4 = 125252
621 001762 104004          HLT+4                    ;SKIP IF SAME
622
623 001764 170005          STAB
624 001766 174067 177010          STF     ACB,ANS1        ;GET AR
625 001772 042767 177600 177002          BIC     #177600,ANS1    ;GET RESULT
626 002000 022767 000052 176774          CMP     #52,ANS1       ;CLEAR JUNK
627 002006 001401          BEQ    ,+4             ;IS ANS1 = 52
628 002010 104002          HLT+2                    ;SKIP IF SAME
629 002012 022767 125252 176764          CMP     #125252,ANS2   ;ANS1 NOT = 52
630 002020 001401          BEQ    ,+4             ;IS ANS2 = 125252
631 002022 104002          HLT+2                    ;SKIP IF SAME
632
633 002024 005003          CLR      R3          ;CLEAR
634 002026 170003          LDUB                     ;UBREAK REG.
635
636 ;.....

```

```

637                                     ;TEST 5          TEST OF MRS
638                                     ;.....
639 002030 104400                       TST5: SCOPE
640
641 002032 170127 040220                LDFPS  #40220      ;DOUBLE/MAINT BITS
642 002036 012703 000115                MOV     #115,R3    ;LOAD 115 INTO
643 002042 170003                       LOUB                      ;UBREAK REG.
644 002044 172467 003260                LOD     ALTA,AC0    ;LOAD ALT 1'S
645 002050 172067 003264                ADD    ALTB,AC0    ;ADD ALT 1'S
646 002054 170006                       MRS
647 002056 170007                       ST00                      ;GET OR INTO AC0
648 002060 174067 176716                STD     AC0,ANS1    ;GET RESULT
649 002064 170200                       STFPS  FPS          ;GET FPS
650 002066 042767 177600 176706        BIC    #177600,ANS1 ;CLEAR JUNK
651 002074 022767 000125 176700        CMP    #125,ANS1   ;IS IT 125?
652 002102 001401                       BEQ    ,+4          ;SKIP IF OK
653 002104 104004                       HLT+4                  ;FIRST WORD IS NOT 125
654
655 002106 170005                       STAB                      ;GET AR INTO AC0
656 002110 174067 176666                STD     AC0,ANS1    ;GET RESULT
657 002114 042767 177600 176660        BIC    #177600,ANS1 ;CLEAR JUNK
658 002122 022767 000152 176652        CMP    #152,ANS1   ;CHECK RESULT
659 002130 001401                       BEQ    ,+4          ;SKIP IF OK
660 002132 104004                       HLT+4                  ;AR NOT 152
661
662 002134 005003                       CLR    R3            ;CLEAR
663 002136 170003                       LOUB                      ;UBREAK REG
664
665                                     ;.....
666 ;TEST 6          TEST OF LOSC FOR MULD OF 1
667 ;.....
668 002140 104400                       TST6: SCOPE
669
670 002142 170127 040220                LDFPS  #40220      ;DOUBLE/MAINT BITS
671 002146 012703 000230                MOV     #230,R3    ;LOAD 230 INTO
672 002152 170003                       LOUB                      ;UBREAK REG.
673 002154 012704 000001                MOV     #1,R4      ;LOAD 1 INTO
674 002160 170004                       LOSC                      ;SHIFT COUNT REG.
675 002162 172467 000206                LOD     1$,AC0     ;LOAD INTO AC0 = 40000,0,0,0
676 002166 171067 000212                MULD   2$,AC0     ;MULD BY 40292,125292,125292,125292
677 002172 170200                       STFPS  FPS          ;STORE FLOATING POINT STATUS
678 002174 170367 176622                STST   FEC         ;STORE EXCEPTION CODES
679 002200 022700 140220                CMP    #140220,FPS ;CHECK FLOATING POINT STATUS
680 002204 001401                       BEQ    ,+4          ;BRANCH IF OK
681 002206 104000                       HLT                      ;FPS NOT EQUAL TO 140220
682
683 002210 022767 000016 176604        CMP    #16,FEC     ;CHECK FLOATING EXCEPTION CODE
684 002216 001401                       BEQ    ,+4          ;BRANCH IF OK
685 002220 104000                       HLT                      ;FEC NOT EQUAL TO 16
686
687 002222 170005                       STAB                      ;GET AR
688 002224 174067 176552                STD     AC0,ANS1    ;GET FOR TYPING
689 002230 042767 177600 176544        BIC    #177600,ANS1 ;CLEAR JUNK
690 002236 170007                       ST00                      ;GET OR

```

691	002240	174067	176546		STD	AC0,ANS5	:GET FOR TYPING
692	002244	042767	177600	176540	BIC	#177600,ANS5	:CLEAR JUNK
693	002252	022767	000000	176522	CMP	#0,ANS1	:IS ANS1 = 0
694	002260	001401			BEO	,+4	:SKIP IF SAME
695	002262	104010			HLT+0.		:ANS1 NOT = 0
696	002264	022767	000000	176512	CMP	#0,ANS2	:IS ANS2 = 0
697	002272	001401			BEO	,+4	:SKIP IF SAME
698	002274	104010			HLT+0.		:ANS2 NOT = 0
699	002276	022767	000000	176502	CMP	#0,ANS3	:IS ANS3 = 0
700	002304	001401			BEO	,+4	:SKIP IF SAME
701	002306	104010			HLT+0.		:ANS3 NOT = 0
702	002310	022767	000000	176472	CMP	#0,ANS4	:IS ANS4 = 0
703	002316	001401			BEO	,+4	:SKIP IF SAME
704	002320	104010			HLT+0.		:ANS4 NOT = 0
705	002322	022767	000125	176462	CMP	#125,ANS5	:IS ANS5 = 125
706	002330	001401			BEO	,+4	:SKIP IF SAME
707	002332	104010			HLT+0.		:ANS5 NOT = 125
708	002334	022767	052525	176452	CMP	#52525,ANS6	:IS ANS6 = 52525
709	002342	001401			BEO	,+4	:SKIP IF SAME
710	002344	104010			HLT+0.		:ANS6 NOT = 52525
711	002346	022767	052525	176442	CMP	#52525,ANS7	:IS ANS7 = 52525
712	002354	001401			BEO	,+4	:SKIP IF SAME
713	002356	104010			HLT+0.		:ANS7 NOT = 52525
714	002360	022767	052525	176432	CMP	#52525,ANS8	:IS ANS8 = 52525
715	002366	001401			BEO	,+4	:SKIP IF SAME
716	002370	104010			HLT+0.		:ANS8 NOT = 52525
717	002372	000410			BR	55	:SKIP JUNK
718							
719	002374	040000	000000	000000	15:	40000,0,0,0	
720	002402	000000					
721	002404	040252	125252	125252	25:	40252,125252,125252,125252	
722	002412	125252					
723							
724	002414	005003			55:	CLR R3	:CLEAR
725	002416	170003				LDUB	:UBREAK REG.
726							
727							
728							
729							
730							
731	002420	104400					
732							
733	002422	170127	040220		LDFPS	#40220	:DOUBLE/MAINT BITS
734	002426	012703	000230		MOV	#230,R3	:LOAD 230 INTO
735	002432	170003			LDUB		:UBREAK REG.
736	002434	012704	000071		MOV	#71,R4	:LOAD 71 INTO
737	002440	170004			LQSC		:SHIFT COUNT REG.
738	002442	172467	000206		LDD	15,AC0	:LOAD INTO AC0 = 40000,0,0,0
739	002446	171067	000212		MULD	25,AC0	:MULD BY 40252,125252,125252,125252
740	002452	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
741	002454	170367	176342		STST	FEC	:STORE EXCEPTION CODES
742	002460	022700	140220		CMP	#140220,FPS	:CHECK FLOATING POINT STATUS
743	002464	001401			BEO	,+4	:BRANCH IF OK
744	002466	104000			HLT		:FPS NOT EQUAL TO 140220

```

:.....
:TEST 7          TEST OF LQSC FOR MUL0 OF 71
:.....
TST7:  SCOPE

```

745											
746	002470	022767	000016	176324		CMP	#16,FEC				:CHECK FLOATING EXCEPTION CODE
747	002476	001401				BEG	,+4				:BRANCH IF OK
748	002500	104000				HLT					:FEC NOT EQUAL TO 16
749											
750	002502	170005				STAB					:GET AR
751	002504	174067	176272			STD	AC0,ANS1				:GET FOR TYPING
752	002510	042767	177600	176264		BIC	#177600,ANS1				:CLEAR JUNK
753	002516	170007				STQB					:GET OR
754	002520	174067	176266			STD	AC0,ANS5				:GET FOR TYPING
755	002524	042767	177600	176260		BIC	#177600,ANS5				:CLEAR JUNK
756	002532	022767	000125	176242		CMP	#125,ANS1				:IS ANS1 = 125
757	002540	001401				BEG	,+4				:SKIP IF SAME
758	002542	104010				HLT+0.					:ANS1 NOT = 125
759	002544	022767	052525	176232		CMP	#52525,ANS2				:IS ANS2 = 52525
760	002552	001401				BEG	,+4				:SKIP IF SAME
761	002554	104010				HLT+0.					:ANS2 NOT = 52525
762	002556	022767	052525	176222		CMP	#52525,ANS3				:IS ANS3 = 52525
763	002564	001401				BEG	,+4				:SKIP IF SAME
764	002566	104010				HLT+0.					:ANS3 NOT = 52525
765	002570	022767	052525	176212		CMP	#52525,ANS4				:IS ANS4 = 52525
766	002576	001401				BEG	,+4				:SKIP IF SAME
767	002600	104010				HLT+0.					:ANS4 NOT = 52525
768	002602	022767	000000	176202		CMP	#0,ANS5				:IS ANS5 = 0
769	002610	001401				BEG	,+4				:SKIP IF SAME
770	002612	104010				HLT+0.					:ANS5 NOT = 0
771	002614	022767	000000	176172		CMP	#0,ANS6				:IS ANS6 = 0
772	002622	001401				BEG	,+4				:SKIP IF SAME
773	002624	104010				HLT+0.					:ANS6 NOT = 0
774	002626	022767	000000	176162		CMP	#0,ANS7				:IS ANS7 = 0
775	002634	001401				BEG	,+4				:SKIP IF SAME
776	002636	104010				HLT+0.					:ANS7 NOT = 0
777	002640	022767	000000	176152		CMP	#0,ANS8				:IS ANS8 = 0
778	002646	001401				BEG	,+4				:SKIP IF SAME
779	002650	104010				HLT+0.					:ANS8 NOT = 0
780	002652	000410				BR	55				:SKIP JUNK
781											
782	002654	040000	000000	000000	15:		40000,0,0,0				
783	002662	000000									
784	002664	040252	125252	125252	25:		40252,125252,125252,125252				
785	002672	125252									
786											
787	002674	005003			55:	CLR	R3				:CLEAR
788	002676	170003				LOUB					:UBREAK REG.
789											
790											
791											
792											
793											
794	002700	104400									
795											
796	002702	170127	040220			LDPPS	#40220				:DOUBLE/MAINT BITS
797	002706	012703	000314			MOV	#314,R3				:LOAD 314 INTO
798	002712	170003				LOUB					:UBREAK REG.

799	002714	012704	000001		MOV	#1,R4	:LOAD 1 INTO
800	002720	170004			LDCS		:SHIFT COUNT REG.
801	002722	172467	000206		LDD	18,AC0	:LOAD INTO AC0 = 40000.0,0,0
802	002726	174467	000212		DIVD	28,AC0	:DIVD BY 40252,125252,125252,125252
803	002732	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
804	002734	170367	176062		STST	FEC	:STORE EXCEPTION CODES
805	002740	022700	140220		CMP	#140220,FPS	:CHECK FLOATING POINT STATUS
806	002744	001401			BEQ	,+4	:BRANCH IF OK
807	002746	104000			HLT		:FPS NOT EQUAL TO 140220
808							
809	002750	022767	000016	176044	CMP	#16,FEC	:CHECK FLOATING EXCEPTION CODE
810	002756	001401			BEQ	,+4	:BRANCH IF OK
811	002760	104000			HLT		:FEC NOT EQUAL TO 16
812							
813	002762	170005			STAB		:GET AR
814	002764	174067	176012		STD	AC0,ANS1	:GET FOR TYPING
815	002770	042767	177600	176004	BIC	#177600,ANS1	:CLEAR JUNK
816	002776	170007			STQB		:GET QR
817	003000	174067	176006		STD	AC0,ANS5	:GET FOR TYPING
818	003004	042767	177600	176000	BIC	#177600,ANS5	:CLEAR JUNK
819	003012	022767	000052	175762	CMP	#52,ANS1	:IS ANS1 = 52
820	003020	001401			BEQ	,+4	:SKIP IF SAME
821	003022	104010			HLT+0.		:ANS1 NOT = 52
822	003024	022767	125252	175752	CMP	#125252,ANS2	:IS ANS2 = 125252
823	003032	001401			BEQ	,+4	:SKIP IF SAME
824	003034	104010			HLT+0.		:ANS2 NOT = 125252
825	003036	022767	125252	175742	CMP	#125252,ANS3	:IS ANS3 = 125252
826	003044	001401			BEQ	,+4	:SKIP IF SAME
827	003046	104010			HLT+0.		:ANS3 NOT = 125252
828	003050	022767	125254	175732	CMP	#125254,ANS4	:IS ANS4 = 125254
829	003056	001401			BEQ	,+4	:SKIP IF SAME
830	003060	104010			HLT+0.		:ANS4 NOT = 125254
831	003062	022767	000125	175722	CMP	#125,ANS5	:IS ANS5 = 125
832	003070	001401			BEQ	,+4	:SKIP IF SAME
833	003072	104010			HLT+0.		:ANS5 NOT = 125
834	003074	022767	052525	175712	CMP	#52525,ANS6	:IS ANS6 = 52525
835	003102	001401			BEQ	,+4	:SKIP IF SAME
836	003104	104010			HLT+0.		:ANS6 NOT = 52525
837	003106	022767	052525	175702	CMP	#52525,ANS7	:IS ANS7 = 52525
838	003114	001401			BEQ	,+4	:SKIP IF SAME
839	003116	104010			HLT+0.		:ANS7 NOT = 52525
840	003120	022767	052524	175672	CMP	#52524,ANS8	:IS ANS8 = 52524
841	003126	001401			BEQ	,+4	:SKIP IF SAME
842	003130	104010			HLT+0.		:ANS8 NOT = 52524
843	003132	000410			BR	58	:SKIP JUNK
844							
845	003134	040000	000000	000000	18:	40000.0,0,0	
846	003142	000000					
847	003144	040252	125252	125252	28:	40252,125252,125252,125252	
848	003152	125252					
849							
850	003154	000003			58:	CLR R3	:CLEAR
851	003156	170003				LOUB	:UBREAK REG.
852							

```

853
854
855
856
857 003160 104400
858
859 003162 170127 040220 LDFPS 040220 ;DOUBLE/MAINT BITS
860 003166 012703 000314 MOV 0314,R3 ;LOAD 314 INTO
861 003172 170003 LDUB ;UBREAK REG,
862 003174 012704 000071 MOV 071,R4 ;LOAD 71 INTO
863 003200 170004 LDSC ;SHIFT COUNT REG,
864 003202 172467 000206 LDD 15,AC0 ;LOAD INTO AC0 = 40000,0,0,0
865 003206 174467 000212 DIVD 25,AC0 ;DIVD BY 40252,125252,125252,125252
866 003212 170200 STFPS FPS ;STORE FLOATING POINT STATUS
867 003214 170367 175602 STST FEC ;STORE EXCEPTION CODES
868 003220 022700 140220 CMP 0140220,FPS ;CHECK FLOATING POINT STATUS
869 003224 001401 BEO ,+4 ;BRANCH IF OK
870 003226 104000 HLT ;FPS NOT EQUAL TO 140220
871
872 003230 022767 000016 175564 CMP 016,FEC ;CHECK FLOATING EXCEPTION CODE
873 003236 001401 BEO ,+4 ;BRANCH IF OK
874 003240 104000 HLT ;FEC NOT EQUAL TO 16
875
876 003242 170005 STAB ;GET AR
877 003244 174067 175532 STD AC0,ANS1 ;GET FOR TYPING
878 003250 042767 177600 175524 BIC 0177600,ANS1 ;CLEAR JUNK
879 003256 170007 STOB ;GET OR
880 003260 174067 175526 STD AC0,ANS5 ;GET FOR TYPING
881 003264 042767 177600 175520 BIC 0177600,ANS5 ;CLEAR JUNK
882 003272 022767 000052 175502 CMP 052,ANS1 ;IS ANS1 = 52
883 003300 001401 BEO ,+4 ;SKIP IF SAME
884 003302 104010 HLT+0. ;ANS1 NOT = 52
885 003304 022767 125252 175472 CMP 0125252,ANS2 ;IS ANS2 = 125252
886 003312 001401 BEO ,+4 ;SKIP IF SAME
887 003314 104010 HLT+0. ;ANS2 NOT = 125252
888 003316 022767 125252 175462 CMP 0125252,ANS3 ;IS ANS3 = 125252
889 003324 001401 BEO ,+4 ;SKIP IF SAME
890 003326 104010 HLT+0. ;ANS3 NOT = 125252
891 003330 022767 125254 175452 CMP 0125254,ANS4 ;IS ANS4 = 125254
892 003336 001401 BEO ,+4 ;SKIP IF SAME
893 003340 104010 HLT+0. ;ANS4 NOT = 125254
894 003342 022767 000014 175442 CMP 014,ANS5 ;IS ANS5 = 14
895 003350 001401 BEO ,+4 ;SKIP IF SAME
896 003352 104010 HLT+0. ;ANS5 NOT = 14
897 003354 022767 000000 175432 CMP 00,ANS6 ;IS ANS6 = 0
898 003362 001401 BEO ,+4 ;SKIP IF SAME
899 003364 104010 HLT+0. ;ANS6 NOT = 0
900 003366 022767 000000 175422 CMP 00,ANS7 ;IS ANS7 = 0
901 003374 001401 BEO ,+4 ;SKIP IF SAME
902 003376 104010 HLT+0. ;ANS7 NOT = 0
903 003400 022767 000000 175412 CMP 00,ANS0 ;IS ANS0 = 0
904 003406 001401 BEO ,+4 ;SKIP IF SAME
905 003410 104010 HLT+0. ;ANS0 NOT = 0
906 003412 000410 BR 55 ;SKIP JUNK

```

907										
908	003414	040000	000000	000000	1S:				40000.0.0.0	
909	003422	000000								
910	003424	040252	125252	125252	2S:				40252.125252.125252.125252	
911	003432	125252								
912										
913	003434	005003			5S:	CLR	R3		:CLEAR	
914	003436	170003				LDUB			:UBREAK REG.	
915										
916										
917	003440	000167	001002			JMP	DONE			

918	003444	005067	175330		TSTMULI	CLR	ICNT	
919	003450	104400				SCOPE		
920	003452	004767	000456			JSR	PC,GETTY	;GET TTY INPUT
921	003456	170127	040220			LDFPS	040220	;DOUBLE/MAINT MODES
922	003462	012703	000230			MOV	0230,R3	
923	003466	170003				LOUB		
924	003470	005004				CLR	R4	
925	003472	012767	003510	002214		MOV	015,LAD	
926	003500	172767	175276			LDD	ANS1,AC3	;GET FIRST
927	003504	172567	175302			LDD	ANS5,AC1	;GET SECOND
928	003510	172603			15:	LDD	AC3,AC2	;RESTORE
929	003512	005204				INC	R4	;INC SHIFT COUNT
930	003514	170004				LOSC		
931	003516	171201				MULD	AC1,AC2	;MULTIPLY
932	003520	170005				STAB		
933	003522	174067	175254			STD	AC0,ANS1	;GET RESULT
934	003526	170007				STOB		
935	003530	174067	175256			STD	AC0,ANS5	;GET RESULT
936								
937	003534	042767	177600	175240		BIC	0177600,ANS1	
938	003542	042767	177600	175242		BIC	0177600,ANS5	
939	003550	170006				MRS		
940	003552	170006				MRS		
941	003554	170005				STAB		
942	003556	174067	000004			STD	AC0,INPUT	
943	003562	006167	000600			ROL	INPUT	
944	003566	006167	000574			ROL	INPUT	
945	003572	042767	177477	000566		BIC	0177477,INPUT	
946	003600	056767	000562	175174		BIS	INPUT,ANS1	
947	003606	170007				STOB		
948	003610	174067	000552			STD	AC0,INPUT	
949	003614	006167	000546			ROL	INPUT	
950	003620	006167	000542			ROL	INPUT	
951	003624	042767	177477	000534		BIC	0177477,INPUT	
952	003632	056767	000530	175152		BIS	INPUT,ANS5	
953	003640	010400				MOV	R4,R0	
954	003642	104010				HLT+0.		
955	003644	022704	000071			CHP	071,R4	
956	003650	001002				BNE	35	
957	003652	105067	175123			CLRB	ICNT+1	
958	003656	104400			35:	SCOPE		
959	003660	170127	040200			LDFPS	040200	
960	003664	172603				LDD	AC3,AC2	
961	003666	171201				MULD	AC1,AC2	
962	003670	170200				STFPS	FPS	
963	003672	104010				HLT+0.		
964	003674	000167	177544			JMP	TSTMUL	

965	003700	005067	175074		TSTDIV: CLR	ICNT	
966	003704	104400			SCOPE		
967	003706	004767	000222		JSR	PC,GETTY	;GET TTY INPUT
968	003712	170127	040220		LDFPS	040220	;DOUBLE/MAINT MODES
969	003716	012703	000314		MOV	0314,R3	
970	003722	170003			LDUB		
971	003724	005004			CLR	R4	
972	003726	012767	003744	001760	MOV	018,LAD	
973	003734	172767	175042		LDD	ANS1,AC3	;GET FIRST
974	003740	172567	175046		LDD	ANS5,AC1	;GET SECOND
975	003744	172603			LDD	AC3,AC2	;RESTORE
976	003746	005204			INC	R4	;INC SWIFT COUNT
977	003750	170004			LDSC		
978	003752	174601			DIVD	AC1,AC2	;DIVIDE
979	003754	170005			STAB		
980	003756	174067	175020		STD	AC0,ANS1	;GET RESULT
981	003762	170007			STOB		
982	003764	174067	175022		STD	AC0,ANS5	;GET RESULT
983							
984	003770	042767	177600	175004	BIC	0177600,ANS1	
985	003776	042767	177600	175006	BIC	0177600,ANS5	
986	004004	170006			MRS		
987	004006	170006			MRS		
988	004010	170005			STAB		
989	004012	174067	000350		STD	AC0,INPUT	
990	004016	006167	000344		ROL	INPUT	
991	004022	006167	000340		RCL	INPUT	
992	004026	042767	177477	000332	BIC	0177477,INPUT	
993	004034	056767	000326	174740	BIS	INPUT,ANS1	
994	004042	170007			STOB		
995	004044	174067	000316		STD	AC0,INPUT	
996	004050	006167	000312		ROL	INPUT	
997	004054	006167	000306		ROL	INPUT	
998	004060	042767	177477	000300	BIC	0177477,INPUT	
999	004066	056767	000274	174716	BIS	INPUT,ANS5	
1000	004074	010400			MOV	R4,R0	
1001	004076	104010			HLT+0.		
1002	004100	022704	000071		CMR	071,R4	
1003	004104	001002			BNE	38	
1004	004106	105067	174667		CLRB	ICNT+1	
1005	004112	104400			SCOPE		
1006	004114	170127	040200		LDFPS	040200	
1007	004120	172603			LDD	AC3,AC2	
1008	004122	174601			DIVD	AC1,AC2	
1009	004124	170200			STFPS	FPS	
1010	004126	104010			HLT+0.		
1011	004130	000167	177544		JMP	TSTDIV	

1012	004134			GETTY:			
1013	004134	000004	004140		TYPE	.,+2	;ASCIZ <15><12>"1)"
1014	004146	005027	000000	FLAG:	CLR	00	
1015	004152	012702	001002		MOV	RANS1,R2	
1016	004156	004567	000106	5S:	JSR	R9,GOGET	
1017	004162	004366			INPUT		
1018	004164	005012			CLR	(2)	
1019	004166	012701	004366		MOV	#INPUT,R1	
1020	004172	122711	000015	3S:	CMPB	#15,(1)	;IS IT A CR?
1021	004176	001420			BEO	15	
1022	004200	122711	000040		CMPB	#40,(1)	;HOW ABOUT A SPACE
1023	004204	001411			BEO	25	
1024	004206	106011			RORB	(1)	
1025	004210	106011			NORB	(1)	
1026	004212	106011			RORB	(1)	
1027	004214	006112			ROL	(2)	
1028	004216	106111			ROLB	(1)	
1029	004220	006112			ROL	(2)	
1030	004222	106121			ROLB	(1)+	
1031	004224	006112			ROL	(2)	
1032	004226	000761			BR	35	
1033	004230	105721		2S:	TSTB	(1)+	;TRY NEXT BYTE
1034	004232	005722			TST	(2)+	
1035	004234	005012			CLR	(2)	
1036	004236	000755			BR	35	
1037	004240	005767	177704	1S:	TST	FLAG+2	;FIRST TIME?
1038	004244	001010			BNE	65	
1039	004246	000004	004252		TYPE	.,+2	;ASCIZ "2)"
1040	004256	005722			TST	(2)+	
1041	004260	005167	177664		COM	FLAG+2	
1042	004264	000734			BR	55	
1043	004266	000207		6S:	RTS	PC	
1044							
1045	004270	012503		GOGET:	MOV	(5)+,R3	;GET ADDRESS
1046	004272	105767	173262	1S:	TSTB	177560	
1047	004276	100375			BPL	,+4	
1048	004300	116713	173256		MOVB	177562,(3)	;GET CHARACTER
1049	004304	142713	000200		BICB	#200,(3)	
1050	004310	111367	001374		MOVB	(3),,TYPE	;SET UP FOR TYPING
1051	004314	000004	005710		TYPE	.,TYPE	
1052	004320	122713	000177		CMPB	#177,(3)	
1053	004324	001411			BEO	25	
1054	004326	122723	000015		CMPB	#15,(3)+	;CHECK FOR RETURN
1055	004332	001357			BNE	15	
1056	004334	012767	000012 001346		MOV	#12,,TYPE	;TYPE A 12
1057	004342	000004	005710		TYPE	.,TYPE	
1058	004346	000205			RTS	R9	
1059	004350	000004	004362	2S:	TYPE	,QUES	
1060	004354	016503	177776		MOV	-2(5),R3	
1061	004360	000744			BR	15	
1062	004362	006477	000012	QUES:	,ASCIZ	"?"<15><12>	
1063	004366	000030		INPUT:	,BLKW	30	

1064	034446	104400			DONE:	SCOPE			
1265	034450	032737	002000	177570		BIT	0SW10,00SWR		;RING THE BELL?
1066	034456	001005				BNE	15		;NO!
1267	034460	012767	000007	001222		MOV	0BELL,,TYPE		;TYPE A BELL
1268	034466	000004	005710			TYPE	,,TYPE		
1069	034472	005046			15:	CLR	-(6)		;CLEAR TRACE TRAP
1070	034474	032737	010000	177570		BIT	0SW12,00SWR		;RUN WITH TRT?
1071	034502	001010				BNE	25		
1072	034504	005167	001202			COM	TRPB		
1073	034510	100005				BPL	25		
1074	034512	052716	000020			BIS	020,(6)		;SET TRACE TRAP
1075	034516	012746	001062			MOV	0BEGIN,-(6)		;JUMP TO START OF TEST
1076	034522	000412				OR	YESRT		
1077	034524	012746	001062		25:	MOV	0BEGIN,-(6)		;JUMP TO START OF TEST
1078	034530	013700	000042			MOV	0042,R0		;GET MONITOR ADDRESS
1079	034534	001404				BEO	35		;IF NONE
1080	034536	004710				JSR	7,(0)		;GO TO MONITOR
1081	034540	000240				NOP			
1082	034542	000240				NOP			
1083	034544	000240				NOP			
1084	034546	000002			35:	RTI			
1085	034550	000002			YESRT:	RTI			;RETURN TO PROGRAM FROM TRAP
1086									
1087	034552	032737	000400	177570	.ENT:	BIT	0SW00,00SWR		;KILL LOUB OR LOOP ON SPEC. TEST
1088	034560	001404				BEO	15		
1089	034562	123767	177570	174210		CMPS	00SWR,ICNT		;ON RIGHT TEST? 0SW7-00
1090	034570	001434				BEO	OVER		
1091	034572	032737	040000	177570	15:	BIT	0SW14,00SWR		;LOOP ON TEST
1092	034600	001026				BNE	KIT		
1093	034602	032737	004000	177570		BIT	0SW11,00SWR		;KILL ITERATIONS
1094	034610	001012				BNE	SAVLAD		
1095	034612	105767	174163			TSTB	ICNT+1		
1096	034616	001404				BEO	25		;BRANCH IF FIRST
1097	034620	126767	001074	174153		CMPS	TIMES,ICNT+1		;DONE?
1098	034626	001013				BNE	KIT		;BRANCH IF NOT
1099	034630	112767	000001	174143	25:	MOVB	01,ICNT+1		;FIRST ITERATION
1100	034636	105267	174136		SAVLAD:	INCB	ICNT		;COUNT TEST NUMBERS
1101	034642	011667	001046			MOV	(0),LAD		;SAVE LOOP ADDRESS
1102	034646	016737	174126	177570		MOV	ICNT,00DISPLAY		;DISPLAY TEST NO. AND ITERATION COUNT
1103	034654	000002				RTI			;RETURN
1104									
1105	034656	105267	174117		KIT:	INCB	ICNT+1		
1106	034662	016737	174112	177570	OVER:	MOV	ICNT,00DISPLAY		;SET UP DISPLAY
1107	034670	005767	001020			TST	LAD		;FIRST ONE?
1108	034674	001760				BEO	SAVLAD		
1109	034676	016716	001012			MOV	LAD,(6)		;FUDGE RETURN ADDRESS
1110	034702	000002				RTI			;FIXES PS

1111	034724	032737	002000	177570	.TRP:	BIT	#SW17,0#SWR	:BELL ON ERROR?
1112	034712	001405				REQ	15	:NO - SKIP
1113	034714	012767	000007	000766		MOV	#BELL,TYPE	:TYPE & BELL
1114	034722	000004	005710			TYPE	,TYPE	
1115	034726	005267	000764		15:	INC	ERRORS	:COUNT THE NUMBER OF ERRORS
1116	034732	010446				MOV	R4,-(6)	
1117	034734	032737	020000	177570		BIT	#SW13,0#SWR	:SKIP TYPEOUT IF SET
1118	034742	001072				RNE	45	
1119	034744	000004	005656			TYPE	,RETJRN	
1120	034750	016646	000002			MOV	2(6),-(6)	:PUT ADDRESS OF INSTRUCTION ON STACK
1121	034754	162716	000002			SUB	#2,(6)	
1122	034760	011605				MOV	(6),TTY	:TYPE (6) IN OCTAL
1123	034762	004767	000156			JSR	X7,PRINTR	:TYPE LEADING ZERO'S
1124	034766	000004	005664			TYPE	,SPACE+3	
1125	034772	010005				MOV	R0,TTY	:TYPE R0 IN OCTAL
1126	034774	004767	000144			JSR	X7,PRINTR	:TYPE LEADING ZERO'S
1127	005000	000004	005665			TYPE	,SPACE+4	
1128	005004	012703	001002			MOV	#ANS1,R3	:ADDRESS OF DATA
1129	005010	113604				MCVB	0(6)+,R4	:AMOUNT OF DATA IN TABLE
1130	005012	001426				BEO	35	
1131	005014	100016				BPL	25	:TYPE STACK?
1132	005016	016667	000006	173756		MOV	6(6),ANS1	
1133	005024	016667	000010	173752		MOV	10(6),ANS2	
1134	005032	016667	000012	173746		MOV	12(6),ANS3	
1135	005040	016667	000014	173742		MOV	14(6),ANS4	
1136	005046	042704	177600			BIC	#177600,R4	:CLEAR SIGN
1137	005052	000004	005665		25:	TYPE	,SPACE+4	
1138	005056	012305				MOV	(3)+,TTY	:TYPE (3)+ IN OCTAL
1139	005060	004767	000060			JSR	X7,PRINTR	:TYPE LEADING ZERO'S
1140	005064	005304				DEC	R4	
1141	005066	001371				RNE	25	
1142	005070	005700			35:	TST	FPS	
1143	005072	100016				BPL	45	
1144	005074	000004	005661			TYPE	,SPACE	
1145	005100	170367	173716			STST	FEC	
1146	005104	016705	173712			MOV	FEC,TTY	:TYPE FEC IN OCTAL
1147	005110	004767	000030			JSR	X7,PRINTR	:TYPE LEADING ZERO'S
1148	005114	000004	005664			TYPE	,SPACE+3	
1149	005120	016705	173700			MOV	FEA,TTY	:TYPE FEA IN OCTAL
1150	005124	004767	000014			JSR	X7,PRINTR	:TYPE LEADING ZERO'S
1151	005130	012604			45:	MOV	(6)+,R4	
1152	005132	005737	177570			TST	0#SWR	:HALT ON ERROR
1153	005136	100001				BPL	,+4	:SKIP IF CONTINUE
1154	005140	000000				HALT		:HALT ON ERROR!
1155	005142	000002				RTI		

1156	005144	112767	000001	000130	PRINTR:	MOVB	#1,A4S		;SET ZERO FILL SWITCH
1157	005152	000402				BR	,+6		
1158	005154	005067	000122		PRINTS:	CLR	A4S		;SUPRESS LEADING ZERO'S
1159	005160	112767	177772	000115		MOVB	#-6,A4S+1		;SET COUNT
1160	005166	010446				MOV	R4,-(6)		;SAVE R4
1161	005170	012704	005272			MOV	#3S,R4		;SET POINTER TO FIRST ASCII CHAR.
1162	005174	105014				CLRB	(4)		;CLEAR FIRST BYTE
1163	005176	000405				BR	25		;ROTATE FIRST BIT
1164	005200	105014			1S:	CLRB	(4)		;CLEAR BYTE OF CHARACTER
1165	005202	006105				ROL	TTY		;ROTATE BIT INTO C
1166	005204	106114				ROLB	(4)		;PACK IT
1167	005206	006105				ROL	TTY		;ROTATE BIT INTO C
1168	005210	106114				ROLB	(4)		;PACK IT
1169	005212	006105			2S:	ROL	TTY		;ROTATE BIT INTO C
1170	005214	106114				ROLB	(4)		;PACK IT
1171	005216	105714				TSTB	(4)		
1172	005220	001402				BEQ	,+6		
1173	005222	105267	000054			INCB	A4S		
1174	005226	105767	000050			TSTB	A4S		;CHECK FILL SWITCH
1175	005232	001402				BEQ	,+6		
1176	005234	152724	000060			BISB	#'0,(4)+		;MAKE INTO ASCII CHAR
1177	005240	105267	000037			INCB	A4S+1		
1178	005244	001355				BNC	1S		;REPEAT
1179	005246	022704	005272			CMP	#3S,R4		
1180	005252	001002				BNE	,+6		
1181	005254	112724	000060			MOVB	#'0,(4)+		
1182	005260	105014				CLRB	(4)		
1183	005262	000004	005272			TYPE	,3S		;TYPE IT
1184	005266	012604				MOV	(6)+,R4		;RESTORE R4
1185	005270	000207				RTS	PC		
1186									
1187	005272	000004			3S:	,BLKW	4		
1188	005302	000000			A4S:	0			
1189									
1190	005304	052525	052525	052525	ALT1A:	52525,52525,52525,52525			
1191	005312	052525							
1192	005314	125252	125252		ALT1B:	125252,125252			
1193	005320	125252	125252	052525	ALT1C:	125252,125252,52525,52525			
1194	005326	052525							
1195	005330	040125	052525	052525	ALTA:	40125,52525,52525,52525			
1196	005336	052525							
1197	005340	040052	125252	125252	ALTB:	40052,125252,125252,125252			
1198	005346	125252							
1199	005350	000125	052525	052525	ANSA:	125,52525,52525,52525			
1200	005356	052525							
1201	005360	000052	125252	125252	ANSB:	52,125252,125252,125252			
1202	005366	125252							
1203									

1204	005370	012777	005564	000306	POWDWN:	MOV	#ILLUP,#UPVEC	;SET FOR FAST UP
1205	005376	012777	000340	000302		MOV	#340,#UPVEC+2	;PRIO:7
1206	005404	170246				STFPS	-(6)	;GET THE FPS
1207	005406	170011				SETD		
1208	005410	174046				STD	AC0,-(6)	;SAVE AC'S
1209	005412	174146				STD	AC1,-(6)	
1210	005414	174246				STD	AC2,-(6)	
1211	005416	174346				STD	AC3,-(6)	
1212	005420	172404				LDD	AC4,AC0	
1213	005422	174046				STD	AC0,-(6)	
1214	005424	172405				LDD	AC5,AC0	
1215	005426	174046				STD	AC0,-(6)	
1216	005430	010046				MOV	R0,-(6)	;SAVE REGISTERS
1217	005432	010146				MOV	R1,-(6)	
1218	005434	010246				MOV	R2,-(6)	
1219	005436	010346				MOV	R3,-(6)	
1220	005440	010446				MOV	R4,-(6)	
1221	005442	010546				MOV	R5,-(6)	
1222	005444	010667	000220			MOV	SP,SAVE6	;SAVE SP
1223	005450	012777	005460	000226		MOV	#POWUP,#UPVEC	;SET UP VECTOR
1224	005456	000000				HALT		
1225								
1226	005460	016706	000204		POWUP:	MOV	SAVE6,SP	;GET SP
1227	005464	005001				CLR	R1	;WAIT LOOP FOR THE TTY
1228	005466	005201			1S:	INC	R1	
1229	005470	001376				BNE	1S	
1230	005472	012605				MOV	(6)+,R5	;GET THE REGISTERS
1231	005474	012604				MOV	(6)+,R4	
1232	005476	012603				MOV	(6)+,R3	
1233	005500	012602				MOV	(6)+,R2	
1234	005502	012601				MOV	(6)+,R1	
1235	005504	012600				MOV	(6)+,R0	
1236	005506	170011				SETD		
1237	005510	172426				LDD	(6)+,AC0	;RESTORE THE AC'S
1238	005512	174005				STD	AC0,AC5	
1239	005514	172426				LDD	(6)+,AC0	
1240	005516	174004				STD	AC0,AC4	
1241	005520	172726				LDD	(6)+,AC3	
1242	005522	172626				LDD	(6)+,AC2	
1243	005524	172526				LDD	(6)+,AC1	
1244	005526	172426				LDD	(6)+,AC0	
1245	005530	170126				LDFPS	(6)+	;RESTORE FPS
1246	005532	012777	005370	000140		MOV	#POWDWN,#DOWNVEC	;SET UP THE POWER DOWN VECTOR
1247	005540	012777	000340	000134		MOV	#340,#DOWNVEC+2	
1248	005546	000004	005552			TYPE	,,+2	;ASCIZ <15><12>"POWER"
1249	005562	000002				RTI		
1250								
1251	005564	000000			ILLUP:	HALT		;THE POWER UP SEQUENCE WAS STARTED
1252	005566	000776				BR	,=2	;BEFORE THE POWER DOWN WAS COMPLETE

```

1253 005570 010546          .IOT:  MOV    TTY,-(6)      ;SAVE TTY
1254 005572 017605 000002    MOV    02(6),TTY    ;GET ADDRESS TO BE TYPED
1255 005576 105715          1S:   TSTB   (TTY)      ;TERMINATOR?
1256 005600 001400          BEQ    2S           ;
1257 005602 112537 177566    MOVB  (TTY)+,0#177566 ;LOAD AND TYPE THE CHARACTER
1258 005606 105737 177564    TSTB  0#177564     ;IS THE PRINTER READY
1259 005612 100375          BPL    ,=4         ;
1260 005614 000770          BR     1S          ;GET THE NEXT CHARACTER
1261 005616 017646 000002    2S:   MOV    02(6),-(6) ;GET ADDRESS TO BE TYPED
1262 005622 062766 000002 000004    ADD   #2,4(6)      ;ADD 2 TO THE ADDRESS
1263 005630 022666 000002    CMP   (6)+,2(6)   ;IS IT ,+2?
1264 005634 001000          BNE   3S          ;NO
1265 005636 062705 000002    ADD   #2,TTY      ;ADD 2 TO THE ADDRESS
1266 005642 042705 000001    BIC   01,TTY      ;BACK UP TO AN EVEN BYTE
1267 005646 010566 000002    MOV   TTY,2(6)   ;RESTORE ADDRESS
1268 005652 012605          3S:   MOV   (6)+,TTY  ;RESTORE TTY
1269 005654 000002          RTI                    ;RETURN
1270
1271 005656 005015 000          RETURN: .ASCIZ <15><12> ;RETURN AND LINEFEED
1272 005661 015 020012 020040 SPACE: .ASCIZ <15><12>" " ;RETURN AND 3 SPACES
1273 005666 000
1274 005670
1275 005670 000000          .EVEN
1276 005672 172160          SAVE6: 0
1277 005674 000244 000246    FPTADR: 172160     ;FLOATING POINT ADDRESS ON THE 11/20
1278 005700 000024 000026    FPVECT: 244,246   ;FLOATING POINT VECTOR ADDRESS
1279 005704 000024 000026    DMNVEC: 24,26     ;POWER DOWN VECTOR ADDRESS
1280 005710 000000          UPVEC: 24,26     ;POWER UP VECTOR ADDRESS
1281 005712 000000          .TYPE: 0
1282 005714 000000          TRPB: 0
1283 005716 000000          LAD: 0            ;LOOP ADDRESS
1284 005720 000377          ERRORS: 0         ;ERROR COUNT
1295 000001          TIMES: 377      ;ITERATION COUNT
          .END

```

AC0	=X000000	4030	4810	4840	495	496	5120	523	524	5380	5410	553	565	5530
		5960	608	624	6440	6450	648	656	6750	6760	688	691	7380	7390
		751	754	8010	8020	814	817	8640	8650	877	880	933	935	942
		948	980	982	989	995	1208	12120	1213	12140	1215	12370	1238	12390
		1240	12440											
AC1	=X000001	4040	9270	931	961	9740	978	1008	1209	12430				
AC2	=X000002	4050	9280	9310	9600	9610	9750	9780	10070	10080	1210	12420		
AC3	=X000003	4060	9260	928	960	9730	975	1007	1211	12410				
AC4	=X000004	4070	1212	12400										
AC5	=X000005	4080	1214	12380										
ALTA	005330	538	596	644	11950									
ALTB	005340	541	593	645	11970									
ALT1A	005304	481	496	11900										
ALT1B	005314	484	512	11920										
ALT1C	005320	524	11930											
ANSA	005350	11990												
ANSB	005360	12010												
ANS1	001002	4370	4950	5230	5530	5540	555	5690	5700	571	6080	6090	610	6240
		6250	626	6480	6500	651	6560	6570	658	6880	6890	693	7510	7520
		756	8140	8150	819	8770	8780	882	926	9330	9370	9460	973	9800
		9840	9930	1015	1120	11320								
ANS2	001004	4380	558	574	613	629	696	759	822	885	11330			
ANS3	001006	4390	561	577	616	699	762	825	888	11340				
ANS4	001010	4400	564	580	619	702	765	828	891	11350				
ANS5	001012	4410	6910	6920	705	7540	7550	768	8170	8180	831	8800	8810	894
		927	9350	9380	9520	974	9820	9850	9990					
ANS6	001014	4420	708	771	834	897								
ANS7	001016	4430	711	774	837	900								
ANS8	001020	4440	714	777	840	903								
A4S	005302	11560	11580	11590	11730	1174	11770	11800						
BEG	001026	425	4480											
BEGIN	001062	452	4570	1075	1077									
BELL	= 000007	3920	1067	1113										
DISPLA	= 177570	3080	11020	11060										
DONE	004446	917	10640											
DWNVEC	005700	4600	4610	12460	12470	12700								
ERRORS	005716	11150	12030											
FEA	001024	4460	1149											
FEC	001022	4310	4450	4860	491	5140	519	5430	548	5980	603	6780	683	7410
		746	8040	809	8670	872	11450	1146						
FLAG	004146	10140	1037	10410										
FLTERR	000760	4300	468											
FPS	=X000000	3930	4300	4850	487	5130	515	5420	544	5970	599	6490	6770	679
		7400	742	8030	805	8660	868	9620	10090	1142				
FPTAJR	005672	454	12760											
FPVECT	005674	4680	4690	12770										
GETTY	004134	920	967	10120										
GOGET	004270	1016	10450											
HLT	= 104000	3900	409	493	499	517	521	527	546	550	557	560	563	566
		573	576	579	582	601	605	612	615	618	621	620	631	653
		660	681	685	695	698	701	704	707	710	713	716	744	748
		750	761	764	767	770	773	776	779	807	811	821	824	827
		830	833	836	839	842	870	874	884	887	890	893	896	899
		902	905	954	963	1001	1010							

ICNT	001000	4360	4700	9180	9570	9650	10040	1009	1095	1097	10990	11000	1102	11050
		1106												
ILLUP	005564	1204	12510											
INPUT	004366	9420	9430	9440	9450	946	9480	9490	9520	9510	952	9890	9900	9510
		9920	993	9950	9960	9970	9980	999	1017	1019	10030			
KIT	004656	1092	1098	11050										
LAD	005714	4710	9250	9720	11010	1107	1109	12020						
LDSC	= 170004	4210												
LDUB	= 170003	4170												
MRS	= 170006	4200	646	939	940	986	987							
M1120	001054	449	4540											
N	= 000012	3050	475	4790	504	5000	532	5360	507	5910	636	6400	665	6090
		720	7320	791	7950	854	8500							
OVER	004662	1090	11060											
PC	=X000007	4020	9200	9670	10430	11050								
POWDN	005370	460	12040	1246										
POWUP	005460	1223	12260											
PRINTR	005144	1123	1126	1139	1147	1150	11560							
PRINTS	005154	11500												
PS	= 177776	3060												
QUES	004362	1059	10620											
RETURN	005656	1119	12710											
R0	=X000000	3940	4630	9530	10000	10780	1125	1216	12350					
R1	=X000001	3950	10190	1217	12270	12280	12340							
R2	=X000002	3960	10150	1210	12330									
R3	=X000003	3970	4730	4820	5010	5100	5290	5390	5040	5940	6330	6420	6620	6710
		7240	7340	7870	7970	8500	8600	9130	9220	9690	10450	10600	11200	1219
		12320												
R4	=X000004	3980	6730	7360	7990	8620	9240	9290	953	955	9710	9760	1000	1002
		1116	11290	11360	11400	11510	1160	11610	1179	11840	1220	12310		
R5	=X000005	3990	10160	10500	1221	12300								
SAVE6	005670	12220	1226	12750										
SAVLAD	004636	1094	11000	1100										
SCOPE	= 104409	3090	470	507	535	590	639	660	731	794	857	919	950	966
		1005	1064											
SP	=X000006	4010	4400	4500	1222	12260								
SPACE	005661	1124	1127	1137	1144	1140	12720							
STAG	= 170005	4100												
ST00	= 170007	4190												
SWR	= 177570	3070	300	1065	1070	1007	1009	1091	1093	1111	1117	1152		
SW00	= 000400	4160	1007											
SW09	= 001000	4150												
SW10	= 002000	4140	1065	1111										
SW11	= 004000	4130	1093											
SW12	= 010000	4120	1070											
SW13	= 020000	4110	1117											
SW14	= 040000	4100	1091											
SW15	= 100000	4090												
TIMES	005720	1097	12040											
TRPB	005712	10720	12010											
TSTDIV	003700	427	9650	1011										
TSTMUL	003444	426	9100	964										
TST1	001204	4700												
TST10	002700	7940												

TST11	003160	8570																		
TST2	001302	5070																		
TST3	001374	5350																		
TST4	001624	5900																		
TST5	002030	6390																		
TST6	002140	6680																		
TST7	002420	7310																		
TTY	=X000005	4000	11220	11250	11380	11460	11490	11650	11670	11690	1253	12540	1255	1257						
		12650	12660	1267	12680															
TYPE	= 000004	3910	1013	1039	1051	1057	1059	1060	1114	1119	1124	1127	1137	1144						
		1140	1183	1248																
UPVEC	005704	12040	12050	12230	12790															
YESRT	004550	4510	459	1076	10850															
.	= 005722	4230	4240	4290	4340	488	492	498	516	520	526	545	549	556						
		559	562	565	572	575	578	581	600	604	611	614	617	620						
		627	630	652	659	680	684	694	697	700	703	706	709	712						
		715	743	747	757	760	763	766	769	772	775	778	806	810						
		820	823	826	829	832	835	838	841	869	873	883	886	889						
		892	895	898	901	904	1013	1039	1047	10630	1153	1157	1172	1175						
		1100	11070	1248	1252	1259	12740													
.EMT	004552	466	10070																	
.IOT	005570	462	12530																	
.TRP	004704	464	11110																	
.TYPE	005710	10500	1051	10560	1057	10670	1068	11130	1114	12000										

CHECK	384#	555	558	561	564	571	574	577	580	610	613	616	619	626	629
	693	696	699	702	705	708	711	714	756	759	762	765	768	771	774
	777	819	822	825	828	831	834	837	840	882	885	888	891	894	897
	900	903													
DUMP	384#	1122	1125	1138	1146	1149									
NUMBER	384#	475	504	532	587	636	665	728	791	854					
PRINT	384#	1012	1039	1248											
SCOPEX	384#	475	504	532	587	636	665	728	791	854					
SDUMP	384#														
STATUS	384#	485	513	542	597	677	742	803	866						
TSTSC	384#	665	728	791	854										
TYPEM	384#	1056	1067	1113											

ADD	1262	1265													
ADDD	541	645													
ADDF	596														
BEQ	488	492	498	516	528	526	545	549	556	559	562	565	572	575	578
	581	608	604	611	614	617	622	627	632	652	659	682	684	694	697
	788	783	786	789	712	715	743	747	757	762	763	766	769	772	775
	778	806	810	820	823	826	829	832	835	838	841	869	873	883	886
	889	892	895	898	901	904	1021	1023	1053	1079	1088	1092	1096	1108	1112
	1130	1172	1175	1256											
BIC	554	578	609	625	658	657	689	692	752	759	815	818	878	881	937
	938	945	951	984	985	992	998	1136	1266						
BICB	1049														
BIS	946	952	993	999	1074										
BISB	1176														
BIT	1065	1070	1087	1091	1093	1111	1117								
BNE	956	1003	1030	1055	1066	1071	1092	1094	1098	1110	1141	1178	1181	1229	1264
BPL	1047	1073	1131	1143	1153	1259									
BR	452	717	788	843	906	1032	1036	1042	1061	1076	1157	1163	1252	1262	
CFCC	497	525													
CLR	478	471	473	581	529	584	633	662	724	787	858	913	918	924	965
	971	1014	1018	1035	1069	1158	1227								
CLRB	957	1084	1162	1164	1182										
CMP	487	491	515	519	544	548	555	558	561	564	571	574	577	588	599
	683	618	613	616	619	626	629	651	658	679	683	693	696	699	702
	785	788	711	714	742	746	756	759	762	765	768	771	774	777	805
	829	819	822	825	828	831	834	837	842	868	872	882	885	888	891
	894	897	908	903	955	1002	1179	1263							
CMPB	1028	1022	1052	1054	1089	1097									
CMPD	496	524													
COM	1041	1072													
DEC	1140														
DIVD	802	865	978	1088											
EMT	398														
HALT	424	432	1154	1224	1251										
INC	929	976	1115	1228											
INCB	1108	1185	1173	1177											
IOT	391														
JMP	425	426	427	917	964	1011									
JSR	920	967	1016	1088	1123	1126	1139	1147	1158						
LDD	481	484	512	538	644	675	738	801	864	926	927	928	961	973	974
	975	1007	1212	1214	1237	1239	1241	1242	1243	1244					
LDF	593														
LDFPS	472	488	589	537	592	641	678	733	796	859	921	959	968	1086	1245
LDSC	674	737	888	863	938	977									
LDUB	474	483	582	511	538	548	585	595	634	643	663	672	725	735	788
	798	851	861	914	923	978									
MOV	448	449	451	454	457	458	459	468	461	462	463	464	465	466	467
	468	469	482	518	539	594	642	671	673	734	736	797	799	868	862
	922	925	953	969	972	1088	1015	1019	1045	1056	1068	1067	1075	1077	1078
	1101	1102	1106	1109	1113	1116	1120	1122	1125	1128	1132	1133	1134	1135	1138
	1146	1149	1151	1168	1161	1164	1204	1205	1216	1217	1218	1219	1221	1221	1222
	1223	1226	1238	1231	1232	1233	1234	1235	1246	1247	1253	1254	1261	1267	1268
MOVB	1048	1058	1099	1129	1156	1159	1181	1257							
MULD	676	739	931	961											

NOP	1081	1082	1083											
ROL	943	944	949	958	998	991	996	997	1027	1029	1031	1165	1167	1169
ROLB	1028	1030	1166	1168	1170									
RORB	1024	1029	1026											
RTI	433	1084	1085	1103	1110	1155	1249	1269						
RTS	1043	1058	1105											
SETD	1207	1236												
STAB	552	623	655	687	758	813	876	932	941	979	988			
STD	495	523	553	569	648	656	688	691	751	754	814	817	877	888
	935	942	948	988	982	989	995	1208	1289	1218	1211	1213	1215	1238
														1242
STF	688	624												
STFPS	438	485	513	542	597	649	677	748	883	866	962	1089	1286	
STQB	568	687	647	698	753	816	879	934	947	981	994			
STST	431	486	514	543	598	678	741	884	867	1145				
SUB	1121													
TRAP	389													
TST	458	1034	1037	1048	1107	1142	1152							
TSTB	1033	1046	1095	1171	1174	1255	1258							
.ASCIZ	1014	1048	1062	1249	1271	1272								
.BLKW	1063	1187												
.ENABL	384													
.END	1285													
.ENDC	487	495	515	523	544	552	599	687	672	679	687	735	742	758
	885	813	861	868	876									
.EVEN	1014	1048	1249	1274										
.IF	486	514	543	598	671	678	734	741	797	884	868	867		
.IFF	672	735	797	868										
.IFNZ	491	495	519	523	548	552	683	687	683	687	746	758	889	813
	876													872
.LIST	344	384	424	434	475	479	584	588	532	536	587	591	636	648
	669	728	732	791	795	854	858	1014	1048	1064	1111	1156	1284	1249
														1253
.MACR	384													
.MACRO	384													
.NLIST	344	384	424	434	475	479	584	588	532	536	587	591	636	648
	669	728	732	791	795	854	858	1014	1048	1064	1111	1156	1284	1249
														1253
.RE4	345													
.REPT	2	424												
.SBTTL	344	384	434	475	584	532	587	636	665	728	791	854	1064	1111
	1284	1253												1256
.TITLE	343													

ERRORS DETECTED: 0

MAINDEC-11-DCFPM-B  
DCFPM.P11

MAINTENANCE INSTRUCTION TEST

MACY11.624 6-MAR-74 10:19 PAGE 37

•DCFPM,DCFPM/SOL/CRF-DCFPM.P11  
RUN-TIME: 6 10 1 SECONDS  
CORE USED: 7K