

FP11

MODF. MODD
MD-11-DCFPL-B

EP-DCFPL B DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made In U.S.A.

REF: 2

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPL
 PRODUCT NAME: P11 BASIC INSTRUCTION TESTS
 DATE CREATED: MARCH 12, 1973
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHORS: BOB BRAIN & KEN CHAPMAN

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION 1973

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.

<u>MAINDEC NO.</u>	<u>INSTRUCTIONS TESTED</u>
DCFPA	LDFPS, STFPS, SETX, SETL SETF, SETD, CFCC
DCFPB	STST
DCFPC	LOF, LDD, STF, STC
DCFPD	ADDF, ADDD, SUBF, SUBD
DCFDE	CMDF, CMPD
DCFPF	MULF, MULD
DCFPG	DIVF, DIVD
DCFPH	CLRF, CLRD, TSTF, TSTD ABSF, ABSD, NEGF, NEGD
DCFPI	LCCFD, LCCDF, STCFD, STCF
DCFPJ	LCCIF, LDCLF, LCCID, LCCID STCFI, STCFL, STCFI, STCFL
DCFPK	LCEXP, STEXP
DCFPL	MOVF, MOVD

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
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

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
 DESCRIPTION

PAGE 3

1. ABSTRACT

THESE PROGRAMS TEST THE FP11 IN ALL MODES WITH FIXED NUMBER PATTERNS. THE PROGRAMS SHOULD BE RUN IN ORDER 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 C - 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
- 6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

MACY11-27(732) 03-SEP-76 14:57 PAGE 4
MACY11-27(732) 03-SEP-76 14:57 PAGE 4

TEST OF MODF AND MODD

MACY11 27(732) 03-SEP-76 14:57 PAGE 4

E01

162
163

THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN
THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE

FF11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

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 LB 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 LOOPEL 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 LOOPEL UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

Vertical column of characters on the left margin, possibly a scan artifact or a specific test sequence indicator.

GO1

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

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 "HLT" 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 C - 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 (EXCEPT DCFPA) 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 RC.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

ADR = ADDRESS OF ERROR HLT
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 HLT; I.E., HLT+3 WOULD TYPE ANS1-ANS3.

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

UNCLASSIFIED//FOR OFFICIAL USE ONLY

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL
DESCRIPTION

PAGE 6

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

THESE PROGRAMS TEST ALL THE INSTRUCTIONS ON THE FP11 IN ALL
MODES. EACH PROGRAM HAS MANY 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. 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 LOOPEO
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.

.ENDP

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

.TITLE MAINDEC-11-DCFP--B TEST OF MODF AND MODD
 :COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
 :PROGRAM BY BOB BRAIN & KEN CHAPMAN
 .REM*

SWITCH	USE
8	0 - LOAD UB REGISTER WITH SW<7:0> 1 - LOOP ON TEST IN SW<7:0>
9	LOOP ON ERROR
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	8		OVERFLOW
5		TRUNCATE MODE	10		UNDERFLOW
6		LONG INTEGER MODE	14		UNDEFINED VARIABLE (-C)
7		DOUBLE PRECISION MODE	16		JBREAK 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*			

```

      .ENABL ABS
000001 N= 1
177776 PS= 177776
177570 SWR= 177570
177570 DISPLAY=SWP
104400 SCOPE= TRAP
104000 HLT= EMT
000004 TYPE= IOT
000007 BELL= 7
000000 FPS= %0
000000 PO= %0
000001 R1= %1
000002 R2= %2
000003 R3= %3
000004 R4= %4
000005 R5= %5
000005 TTY= %5
000006 SP= %6
000007 PC= %7
000000 ACO= %0
000001 AC1= %1
000002 AC2= %2
000003 AC3= %3
000004 AC4= %4
000005 AC5= %5
100000 SW15= 100000
040000 SW14= 40000
020000 SW13= 20000
010000 SW12= 10000
004000 SW11= 4000
002000 SW10= 2000
001000 SW09= 1000
000400 SW08= 400
170003 LD08= 170003
170005 STA0= 170005
170007 ST0C= 170007
170006 MRS= 170006
170004 LD0C= 170004

000000 .= 0 ;TRAP CATCHER FROM 0 - 776
000200 .= 200

000200 000167 000622 JMP BEG

000760 000760 .= 760
000762 170200 FILTER: STFPS FPS
000766 170367 000034 STST FEC
000770 000000 HALT
000002 000002 RTI

```

```

001000      001000      . =      1000
001000 000000      ICNT:      0      ; ITERATION COUNT - LH TEST NO. - PH
001002 000000      ANS1:      0      ; FIRST ANSWER (SEE CODE)
001004 000000      ANS2:      0
001006 000000      ANS3:      0
001010 000000      ANS4:      0
001012 000000      ANS5:      0
001014 000000      ANS6:      0
001016 000000      ANS7:      0
001020 000000      ANS8:      0
001022 000000      FEC:      0      ; FLOATING EXCEPTION CODES
001024 000000      FEA:      0      ; FLOATING EXECPTION ADDRESS

001026 012706 000600      BEG:  MOV      #600,SP      ; ** STACK AT 600 **
001032 012737 001054 000004      MOV      #M1120, @#4      ; FIND OUT WHICH MACHINE THIS IS
001040 005737 177772      TST      @#177772      ; IS PIRQ THERE?
001044 012767 000006 013634      MOV      #6, YESRT      ; FUDGE IN RTT IF 11/45
001052 000403      BR

001054 016737 014770 000010 M1120: MOV      FPTADR, @#10      ; LOAD THE ILLEGAL INSTRUCTION VECTOR
; WITH THE ADDRESS OF THE FPU.
; THE FPU WILL HANDLE THE BAD OPCODES
; RESET 4

001062 012737 000006 000004 BEGIN: MOV      #6, @#4
001070 012706 000600      MOV      #600, SP
001074 012737 014706 000014      MOV      #YESRT, @#14      ; SET TRACE TRAP VECTOR
001102 012777 015546 014746      MOV      #POWDWN, @DWNVEC
001110 012777 000340 014742      MOV      #340, @DWNVEC+2
001116 012737 015746 000020      MOV      #.IOT, @#20      ; SET UP VECTOR 20
001124 012700 000030      MOV      #30, R0      ; SET R0 TO VECTOR 30
001130 012720 015050      MOV      #.TRAP, (0)+      ; SET EMT VECTOR
001134 012720 000340      MOV      #340, (0)+
001140 012720 014710      MOV      #.EMT, (0)+      ; SET TRAP VECTOR
001144 012710 000340      MOV      #340, (0)
001150 012777 000760 014674      MOV      #FLTERR, @FPVECT      ; LOAD INTERRUPT VECTOR
001156 012777 000340 014670      MOV      #340, @FPVECT+2      ; LOCK UP PROCESSOR
001164 005067 177610      CLR      ICNT
001170 005067 014676      CLR      LAD

```

:TEST 1 TEST OF MODF FPU INSTRUCTION
: 0 0 * 0 0 = 0 0 & 0 0
: USING ACO FPS = 47404 FEC = N/A
:*****

001174	104400			SCOPE			
001176	170127	047400		LDFPS	#47404&57760		
001202	172467	000030		LDF	N1,0	;LOAD 0 INTO ACO	
001206	171467	000030		MODF	M1,0	;MODIFY 0 BY 0	
001212	170200			STFPS	FPS	;STORE FLOATING POINT STATUS	
001214	022700	047404		CMP	#47404.FPS	;CHECK FLOATING POINT STATUS	
001220	001401			BEQ	+.4	;BRANCH IF OK	
001222	104000			HLT		;FPS NOT EQUAL TO 47404	
001224	174067	177552		STF	0,ANS1	;STORE FRACTIONAL PART	
001230	174167	177552		STF	0!1,ANS3	;STORE INTEGER PART	
001234	000404			BR	01		
001236	000000	000000	N1:	0.0			
001242	000000	000000	M1:	0.0			
001246	022767	000000	177526	01:	CMP	#0,ANS1	;CHECK LEFT HALF OF FRACTION
001254	001401				BEQ	+.4	
001256	104004				HLT+4		;LEFT HALF #0
001260	022767	000000	177516		CMP	#0,ANS2	;CHECK RIGHT HALF OF FRACTION
001266	001401				BEQ	+.4	
001270	104004				HLT+4		;RIGHT HALF #0
001272	022767	000000	177506		CMP	#0,ANS3	;CHECK LEFT HALF OF INTEGER
001300	001401				BEQ	+.4	
001302	104004				HLT+4		;LEFT HALF #0
001304	022767	000000	177476		CMP	#0,ANS4	;CHECK RIGHT HALF OF INTEGER
001312	001401				BEQ	+.4	
001314	104004				HLT+4		;RIGHT HALF #0


```

*****
:TEST 3          TEST OF MODF FPU INSTRUCTION
:40200 0 * 0 0 = 0 0 & 0 0
:USING AC2      FPS = 47404      FEC = N/A
*****

```

001440	104400			SCOPE		
001442	170127	047400		LDFPS	#47404&57760	
001446	172667	000030		LDF	M3,2	;LOAD 40200 INTO AC2
001452	171667	000030		MODF	M3,2	;MODIFY 40200 BY C
001456	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
001460	022700	047404		CMP	#47404,FPS	;CHECK FLOATING POINT STATUS
001464	001401			BEQ	+.4	;BRANCH IF OK
001466	104000			HLT		;FPS NOT EQUAL TO 47404
001470	174267	177306		STF	2,ANS1	;STORE FRACTIONAL PART
001474	174367	177306		STF	2!1,ANS3	;STORE INTEGER PART
001500	000404			BR	03	
001502	040200	000000	N2:	40200.0		
001506	000000	000000	M3:	0,0		
001512	022767	000000	177262	03:	CMP	#0,ANS1
001520	001401				BEQ	+.4
001522	104004				HLT+4	
						;CHECK LEFT HALF OF FRACTION
						;LEFT HALF #0
001524	022767	000000	177252		CMP	#0,ANS2
001532	001401				BEQ	+.4
001534	104004				HLT+4	
						;CHECK RIGHT HALF OF FRACTION
						;RIGHT HALF #0
001536	022767	000000	177242		CMP	#0,ANS3
001544	001401				BEQ	+.4
001546	104004				HLT+4	
						;CHECK LEFT HALF OF INTEGER
						;LEFT HALF #0
001550	022767	000000	177232		CMP	#0,ANS4
001556	001401				BEQ	+.4
001560	104004				HLT+4	
						;CHECK RIGHT HALF OF INTEGER
						;RIGHT HALF #0

```

.....
TEST 4      TEST OF MODF FPU INSTRUCTION
40200 0 + 40200 0 = 40200 0 & 0 0
USING AC2   FPS = 47404     FEC = 4/A
.....

```

Address	Instruction	PC	PSW	Accumulator	Register	Comments
001610	LD 40200, AC2	04000	00000000	40200.00	AC2	: LOAD 40200 INTO AC2
001612	MODF 40200, AC2	174267	177164	40200.00	AC2	: MODIFY 40200 BY 40200
001614	ST 40200, 000000	174367	177164	40200.00	000000	: STORE FLOATING POINT
001616	ST 40200, 000000	174467	177164	40200.00	000000	: CHECK FLOATING POINT
001618	BR 04	000404	000000	40200.00	04	: BRANCH IF OK
001620	LD 40200, AC2	04000	00000000	40200.00	AC2	: FPS NOT EQUAL TO 47404
001622	ST 40200, 000000	174267	177164	40200.00	000000	: STORE FRACTIONAL PART
001624	ST 40200, 000000	174367	177164	40200.00	000000	: STORE INTEGER PART
001626	LD 40200, AC2	04000	00000000	40200.00	AC2	
001628	LD 40200, AC2	04000	00000000	40200.00	AC2	
001630	LD 40200, AC2	04000	00000000	40200.00	AC2	
001632	LD 40200, AC2	04000	00000000	40200.00	AC2	
001634	LD 40200, AC2	04000	00000000	40200.00	AC2	
001636	LD 40200, AC2	04000	00000000	40200.00	AC2	
001638	LD 40200, AC2	04000	00000000	40200.00	AC2	
001640	LD 40200, AC2	04000	00000000	40200.00	AC2	
001642	LD 40200, AC2	04000	00000000	40200.00	AC2	
001644	LD 40200, AC2	04000	00000000	40200.00	AC2	
001646	LD 40200, AC2	04000	00000000	40200.00	AC2	
001648	LD 40200, AC2	04000	00000000	40200.00	AC2	
001650	LD 40200, AC2	04000	00000000	40200.00	AC2	
001652	LD 40200, AC2	04000	00000000	40200.00	AC2	
001654	LD 40200, AC2	04000	00000000	40200.00	AC2	
001656	LD 40200, AC2	04000	00000000	40200.00	AC2	
001658	LD 40200, AC2	04000	00000000	40200.00	AC2	
001660	LD 40200, AC2	04000	00000000	40200.00	AC2	
001662	LD 40200, AC2	04000	00000000	40200.00	AC2	
001664	LD 40200, AC2	04000	00000000	40200.00	AC2	
001666	LD 40200, AC2	04000	00000000	40200.00	AC2	
001668	LD 40200, AC2	04000	00000000	40200.00	AC2	
001670	LD 40200, AC2	04000	00000000	40200.00	AC2	
001672	LD 40200, AC2	04000	00000000	40200.00	AC2	
001674	LD 40200, AC2	04000	00000000	40200.00	AC2	
001676	LD 40200, AC2	04000	00000000	40200.00	AC2	
001678	LD 40200, AC2	04000	00000000	40200.00	AC2	
001680	LD 40200, AC2	04000	00000000	40200.00	AC2	
001682	LD 40200, AC2	04000	00000000	40200.00	AC2	
001684	LD 40200, AC2	04000	00000000	40200.00	AC2	
001686	LD 40200, AC2	04000	00000000	40200.00	AC2	
001688	LD 40200, AC2	04000	00000000	40200.00	AC2	
001690	LD 40200, AC2	04000	00000000	40200.00	AC2	
001692	LD 40200, AC2	04000	00000000	40200.00	AC2	
001694	LD 40200, AC2	04000	00000000	40200.00	AC2	
001696	LD 40200, AC2	04000	00000000	40200.00	AC2	
001698	LD 40200, AC2	04000	00000000	40200.00	AC2	
001700	LD 40200, AC2	04000	00000000	40200.00	AC2	

```

:*****
:TEST 5 TEST OF MODF FPU INSTRUCTION
: 40214 146315 * 40200 C = 40200 0 & 37314 146320
: USING ACC FPS = 47400 FEC = N/A
:*****

```

001704	104400				SCOPE		
001706	170127	047400			LDFPS	#47400&57760	
001712	172467	000030			LDF	M5,0	:LOAD 40214 INTO ACC
001716	171467	000030			MODF	M5,0	:MODIFY 40214 BY 40200
001722	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
001724	022700	047400			CMP	#47400,FPS	:CHECK FLOATING POINT STATUS
001730	001401				BEG	+.4	:BRANCH IF OK
001732	104000				HLT		:FPS NOT EQUAL TO 47400
001734	174067	177042			STF	0,ANS1	:STORE FRACTIONAL PART
001740	174167	177042			STF	0!1,ANS3	:STORE INTEGER PART
001744	000404				BR	05	
001746	040214	146315		MS:	40214,146315		
001752	040200	000000		MS:	40200,0		
001756	022767	037314	177016	05:	CMP	#37314,ANS1	:CHECK LEFT HALF OF FRACTION
001764	001401				BEG	+.4	
001766	104004				HLT+4		:LEFT HALF #37314
001770	022767	146320	177006		CMP	#146320,ANS2	:CHECK RIGHT HALF OF FRACTION
001776	001401				BEG	+.4	
002000	104004				HLT+4		:RIGHT HALF #146320
002002	022767	040200	176776		CMP	#40200,ANS3	:CHECK LEFT HALF OF INTEGER
002010	001401				BEG	+.4	
002012	104004				HLT+4		:LEFT HALF #40200
002014	022767	000000	176766		CMP	#0,ANS4	:CHECK RIGHT HALF OF INTEGER
002022	001401				BEG	+.4	
002024	104004				HLT+4		:RIGHT HALF #0

:TEST 6 TEST OF MODF FPU INSTRUCTION
: 40414 146315 * 40200 0 = 40400 0 & 37514 146320
: USING AC2 FPS = 47400 FEC = N/A
:*****

002026	104400				SCOPE		
002030	170127	047400			LDFPS	#47400&57760	
002034	172667	000030			LDF	M6,2	:LOAD 40414 INTO AC2
002040	171667	000030			MODF	M6,2	:MODIFY 40414 BY 40200
002044	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
002046	022700	047400			CMP	#47400.FPS	:CHECK FLOATING POINT STATUS
002052	001401				BEQ	.+4	:BRANCH IF OK
002054	104000				HLT		:FPS NOT EQUAL TO 47400
002056	174267	176720			STF	2,ANS1	:STORE FRACTIONAL PART
002062	174367	176720			STF	2,1,ANS3	:STORE INTEGER PART
002066	000404				BR	06	
002070	040414	146315		NS:	40414,146315		
002074	040200	000000		MS:	40200,0		
002100	022767	037514	176674	06:	CMP	#37514,ANS1	:CHECK LEFT HALF OF FRACTION
002106	001401				BEQ	.+4	
002110	104004				HLT+4		:LEFT HALF #37514
002112	022767	146320	176664		CMP	#146320,ANS2	:CHECK RIGHT HALF OF FRACTION
002120	001401				BEQ	.+4	
002122	104004				HLT+4		:RIGHT HALF #146320
002124	022767	040400	176654		CMP	#40400,ANS3	:CHECK LEFT HALF OF INTEGER
002132	001401				BEQ	.+4	
002134	104004				HLT+4		:LEFT HALF #40400
002136	022767	000000	176644		CMP	#0,ANS4	:CHECK RIGHT HALF OF INTEGER
002144	001401				BEQ	.+4	
002146	104004				HLT+4		:RIGHT HALF #0

```

*****
:TEST 7          TEST OF MODF FPU INSTRUCTION
: 40523 31463 * 40200 0 = 40500 0 & 37631 114630
: USING AC2          FPS = 47400      FEC = N/A
*****

```

002150	104400			SCOPE			
002152	170127	047400		LDFPS	#47400&57760		
002156	172667	000030		LDF	M7.2		:LOAD 40523 INTO AC2
002162	171667	000030		MODF	M7.2		:MODIFY 40523 BY 40200
002166	170200			STFPS	FPS		:STORE FLOATING POINT STATUS
002170	022700	047400		CMP	#47400.FPS		:CHECK FLOATING POINT STATUS
002174	001401			BEG	.+4		:BRANCH IF OK
002176	104000			HLT			:FPS NOT EQUAL TO 47400
002200	174267	176576		STF	2.ANS1		:STORE FRACTIONAL PART
002204	174367	176576		STF	2!1.ANS3		:STORE INTEGER PART
002210	000404			BR	07		
002212	040523	031463	N7:	40523.31463			
002216	040200	000000	M7:	40200.0			
002222	022767	037631	07:	CMP	#37631.ANS1		:CHECK LEFT HALF OF FRACTION
002230	001401			BEG	.+4		
002232	104004			HLT+4			:LEFT HALF #37631
002234	022767	114630		CMP	#114630.ANS2		:CHECK RIGHT HALF OF FRACTION
002242	001401			BEG	.+4		
002244	104004			HLT+4			:RIGHT HALF #114630
002246	022767	040500		CMP	#40500.ANS3		:CHECK LEFT HALF OF INTEGER
002254	001401			BEG	.+4		
002256	104004			HLT+4			:LEFT HALF #40500
002260	022767	000000		CMP	#0.ANS4		:CHECK RIGHT HALF OF INTEGER
002266	001401			BEG	.+4		
002270	104004			HLT+4			:RIGHT HALF #0

F02

MAINDEC-11-DCFPL-B
DCFPLE.F11

TEST OF MODF AND MODD MACY1: 27(732) 03-SEP-76 14:57 PAGE 18

```

*****
:EST 10 TEST OF MODF FPU INSTRUCTION
: 40660 0 * 40200 0 = 40640 0 & 40000 0
: USING ACO FPS = 47400 FEC = N/A
*****
  
```

002272	104400				SCOPE		
002274	170127	047400			LDFPS	#47400&57760	
002300	172467	000030			LDF	M10,0	:LOAD 40660 INTO ACO
002304	171467	000030			MODF	M10,0	:MODIFY 40660 BY 40200
002310	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
002312	022700	047400			CMP	#47400,FPS	:CHECK FLOATING POINT STATUS
002316	001401				BEQ	.+4	:BRANCH IF OK
002320	104000				HLT		:FPS NOT EQUAL TO 47400
002322	174067	176454			STF	0,ANS1	:STORE FRACTIONAL PART
002326	174167	176454			STF	0!1,ANS3	:STORE INTEGER PART
002332	000404				BR	010	
002334	040660	000000			M10:	40660,0	
002340	040200	000000			M10:	40200,0	
002344	022767	040000	176430	010:	CMP	#40000,ANS1	:CHECK LEFT HALF OF FRACTION
002352	001401				BEQ	.+4	
002354	104004				HLT+4		:LEFT HALF #40000
002356	022767	000000	176420		CMP	#0,ANS2	:CHECK RIGHT HALF OF FRACTION
002364	001401				BEQ	.+4	
002366	104004				HLT+4		:RIGHT HALF #0
002370	022767	040640	176410		CMP	#40640,ANS3	:CHECK LEFT HALF OF INTEGER
002376	001401				BEQ	.+4	
002400	104004				HLT+4		:LEFT HALF #40640
002402	022767	000000	176400		CMP	#0,ANS4	:CHECK RIGHT HALF OF INTEGER
002410	001401				BEQ	.+4	
002412	104004				HLT+4		:RIGHT HALF #0

```

*****
:TEST 11 TEST OF MODF FPU INSTRUCTION
: 140200 0 * 40200 0 = 140200 0 & 0 0
: USING AC2 FPS = 47404 FEC = N/A
*****

```

002414	104400			SCOPE		
002416	170127	047400		LDFPS	#47404&57760	
002422	172667	000030		LDF	N11.2	:LOAD 140200 INTO AC2
002426	171667	000030		MODF	M11.2	:MODIFY 140200 BY 40200
002432	170200			STFPS	FPS	:STORE FLOATING POINT STATUS
002434	022700	047404		CMP	#47404.FPS	:CHECK FLOATING POINT STATUS
002440	001401			BEQ	.+4	:BRANCH IF OK
002442	104000			HLT		:FPS NOT EQUAL TO 47404
002444	174267	176332		STF	2,ANS1	:STORE FRACTIONAL PART
002450	174367	176332		STF	2!1,ANS3	:STORE INTEGER PART
002454	000404			BR	011	
002456	140200	000000	N11:	140200.0		
002462	040200	000000	M11:	40200.0		
002466	022767	000000	176306	011:	CMP #0,ANS1	:CHECK LEFT HALF OF FRACTION
002474	001401			BEQ	.+4	
002476	104004			HLT+4		:LEFT HALF #0
002500	022767	000000	176276	CMP	#0,ANS2	:CHECK RIGHT HALF OF FRACTION
002506	001401			BEQ	.+4	
002510	104004			HLT+4		:RIGHT HALF #0
002512	022767	140200	176266	CMP	#140200,ANS3	:CHECK LEFT HALF OF INTEGER
002520	001401			BEQ	.+4	
002522	104004			HLT+4		:LEFT HALF #140200
002524	022767	000000	176256	CMP	#0,ANS4	:CHECK RIGHT HALF OF INTEGER
002532	001401			BEQ	.+4	
002534	104004			HLT+4		:RIGHT HALF #0

H02

MAINDEC-11-CCFPL-B
CCFPLB.P11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 20

```
*****  
:TEST 12 TEST OF MODF FPU INSTRUCTION  
: 140214 146315 * 40200 0 = 140200 0 & 137314 146320  
: USING AC2 FPS = 47410 FEC = N/A  
:*****
```

Address	Instruction	Operand 1	Operand 2	Operand 3	Operand 4	Comment
002536	LD	104400				
002540	LDFPS	170127	047400			
002544	LDF	172667	000030			
002550	MODF	171667	000030			
002554	STFPS	170200				
002556	CMP	022700	047410			
002562	BEG	001401				
002564	HLT	104000				
002566	STF	174267	176210			:STORE FRACTIONAL PART
002572	STF	174367	176210			:STORE INTEGER PART
002576	BR	000404				
002600		140214	146315			
002604		040200	000000			
002610		022767	137314	176164	012:	
002616		001401				
002620		104004				
002622		022767	146320	176154		
002630		001401				
002632		104004				
002634		022767	140200	176144		
002642		001401				
002644		104004				
002646		022767	000000	176134		
002654		001401				
002656		104004				

```

*****
:TEST 13 TEST OF MODF FPU INSTRUCTION
: 140414 146315 * 40200 0 = 140400 0 & 137514 146320
: USING AC2 FPS = 47410 FEC = N/A
*****

```

002660	104400				SCOPE		
002662	170127	047400			LDFPS	#47410&57760	
002666	172667	000030			LDF	M13.2	;LOAD 140414 INTO AC2
002672	171667	000030			MODF	M13.2	;MODIFY 140414 BY 40200
002676	170200				STFPS	FPS	;STORE FLOATING POINT STATUS
002700	022700	047410			CMP	#47410.FPS	;CHECK FLOATING POINT STATUS
002704	001401				BEQ	.+4	;BRANCH IF OK
002706	104000				HLT		;FPS NOT EQUAL TO 47410
002710	174267	176066			STF	2,ANS1	;STORE FRACTIONAL PART
002714	174367	176066			STF	2,1,ANS3	;STORE INTEGER PART
002720	000404				BR	013	
002722	140414	146315					
002726	040200	000000			M13:	140414,146315	
					M13:	40200,0	
002732	022767	137514	176042	013:	CMP	#137514,ANS1	;CHECK LEFT HALF OF FRACTION
002740	001401				BEQ	.+4	
002742	104004				HLT+4		;LEFT HALF #137514
002744	022767	146320	176032		CMP	#146320,ANS2	;CHECK RIGHT HALF OF FRACTION
002752	001401				BEQ	.+4	
002754	104004				HLT+4		;RIGHT HALF #146320
002756	022767	140400	176022		CMP	#140400,ANS3	;CHECK LEFT HALF OF INTEGER
002764	001401				BEQ	.+4	
002766	104004				HLT+4		;LEFT HALF #140400
002770	022767	000000	176012		CMP	#0,ANS4	;CHECK RIGHT HALF OF INTEGER
002776	001401				BEQ	.+4	
002780	104004				HLT+4		;RIGHT HALF #0

```

*****
:TEST 14 TEST OF MODF FPU INSTRUCTION
: 140523 31463 * 40200 0 = 140500 0 & 137631 114630
: USING ACO FPS = 47410 FEC = N/A
*****

```

003002	104400				SCOPE		
003004	170127	047400			LDFPS	#47410&57760	
003010	172467	000030			LDF	N14,0	:LOAD 140523 INTO ACO
003014	171467	000030			MODF	M14,0	:MODIFY 140523 BY 40200
003020	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
003022	022700	047410			CMP	#47410.FPS	:CHECK FLOATING POINT STATUS
003026	001401				BEQ	+.4	:BRANCH IF OK
003030	104000				HLT		:FPS NOT EQUAL TO 47410
003032	174067	175744			STF	0,ANS1	:STORE FRACTIONAL PART
003036	174167	175744			STF	0:1,ANS3	:STORE INTEGER PART
003042	000404				BR	014	
003044	140523	031463		N14:	140523,31463		
003050	040200	000000		M14:	40200,0		
003054	022767	137631	175720	014:	CMP	#137631,ANS1	:CHECK LEFT HALF OF FRACTION
003062	001401				BEQ	+.4	
003064	104004				HLT+4		:LEFT HALF #137631
003066	022767	114630	175710		CMP	#114630,ANS2	:CHECK RIGHT HALF OF FRACTION
003074	001401				BEQ	+.4	
003076	104004				HLT+4		:RIGHT HALF #114630
003100	022767	140500	175700		CMP	#140500,ANS3	:CHECK LEFT HALF OF INTEGER
003106	001401				BEQ	+.4	
003110	104004				HLT+4		:LEFT HALF #140500
003112	022767	000000	175670		CMP	#0,ANS4	:CHECK RIGHT HALF OF INTEGER
003120	001401				BEQ	+.4	
003122	104004				HLT+4		:RIGHT HALF #0

K02

MAINDEC-11-DCFPL-B
DCFPLB.F11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 23

```

*****
:TEST 15 TEST OF MODF FPU INSTRUCTION
: 140660 0 * 40200 0 = 140640 0 & 140000 0
: USING ACO FPS = 47410 FEC = N/A
*****

```

003124	104400			SCOPE		
003126	170127	047400		LDFPS	#47410&57760	
003132	172467	000030		LDF	M15,0	;LOAD 140660 INTO ACO
003136	171467	000030		MODF	M15,0	;MODIFY 140660 BY 40200
003142	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
003144	022700	047410		CMP	#47410,FPS	;CHECK FLOATING POINT STATUS
003150	001401			BEQ	.+4	;BRANCH IF OK
003152	104000			HLT		;FPS NOT EQUAL TO 47410
003154	174067	175622		STF	0,ANS1	;STORE FRACTIONAL PART
003160	174167	175622		STF	0!1,ANS3	;STORE INTEGER PART
003164	000404			BR	015	
003166	140660	000000		M15:	140660,0	
003172	040200	000000		M15:	40200,0	
003176	022767	140000	175576	015:	CMP	#140000,ANS1
003204	001401				BEQ	.+4
003206	104004				HLT+4	
003210	022767	000000	175566		CMP	#0,ANS2
003216	001401				BEQ	.+4
003220	104004				HLT+4	
003222	022767	140640	175556		CMP	#140640,ANS3
003230	001401				BEQ	.+4
003232	104004				HLT+4	
003234	022767	000000	175546		CMP	#0,ANS4
003242	001401				BEQ	.+4
003244	104004				HLT+4	

```

:*****
:TEST 16 TEST OF MODF FPU INSTRUCTION
: 52525 52525 * 40200 0 = 52525 52525 & 0 0
: USING ACO FPS = 47404 FEC = N/A
:*****

```

003246	104400				SCOPE			
003250	170127	047400			LDFPS	#47404&57760		
003254	172467	000030			LDF	N16,0		;LOAD 52525 INTO ACO
003260	171467	000030			MODF	M16,0		;MODIFY 52525 BY 40200
003264	170200				STFPS	FPS		;STORE FLOATING POINT STATUS
003266	022700	047404			CMP	#47404,FPS		;CHECK FLOATING POINT STATUS
003272	001401				BEQ	+.4		;BRANCH IF OK
003274	104000				HLT			;FPS NOT EQUAL TO 47404
003276	174067	175500			STF	0,ANS1		;STORE FRACTIONAL PART
003302	174167	175500			STF	0!1,ANS3		;STORE INTEGER PART
003306	000404				BR	016		
003310	052525	052525			M16	52525,52525		
003314	040200	000000			M16:	40200,0		
003320	022767	000000	175454	016:	CMP	#0,ANS1		;CHECK LEFT HALF OF FRACTION
003326	001401				BEQ	+.4		
003330	104004				HLT+4			;LEFT HALF #0
003332	022767	000000	175444		CMP	#0,ANS2		;CHECK RIGHT HALF OF FRACTION
003340	001401				BEQ	+.4		
003342	104004				HLT+4			;RIGHT HALF #0
003344	022767	052525	175434		CMP	#52525,ANS3		;CHECK LEFT HALF OF INTEGER
003352	001401				BEQ	+.4		
003354	104004				HLT+4			;LEFT HALF #52525
003356	022767	052525	175424		CMP	#52525,ANS4		;CHECK RIGHT HALF OF INTEGER
003364	001401				BEQ	+.4		
003366	104004				HLT+4			;RIGHT HALF #52525

M02

MAINDEC-11-DCFPL-B
DCFPLB.F11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 25

```

*****
TEST 17          TEST OF MODF FFU INSTRUCTION
37600 0 * 40200 0 = 0 0 & 37600 0
USING ACO      FPS = 47400      FEC = N/A
*****
    
```

003370	104400			SCOPE		
003372	170127	047400		LDFPS	#47400&57760	
003376	172467	000030		LDF	N17,0	;LOAD 37600 INTO ACO
003402	171467	000030		MODF	M17,0	;MODIFY 37600 BY 40200
003406	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
003410	022700	047400		CMP	#47400,FPS	;CHECK FLOATING POINT STATUS
003414	001401			BEQ	.+4	;BRANCH IF OK
003416	104000			HLT		;FPS NOT EQUAL TO 47400
003420	174067	175356		STF	0,ANS1	;STORE FRACTIONAL PART
003424	174167	175356		STF	0!1,ANS3	;STORE INTEGER PART
003430	000404			BR	017	
003432	037600	000000	N17:	37600,0		
003436	040200	000000	M17:	40200,0		
003442	022767	037600	175332	017:	CMP #37600,ANS1	;CHECK LEFT HALF OF FRACTION
003450	001401			BEQ	.+4	
003452	104004			HLT+4		;LEFT HALF #37600
003454	022767	000000	175322		CMP #0,ANS2	;CHECK RIGHT HALF OF FRACTION
003462	001401			BEQ	.+4	
003464	104004			HLT+4		;RIGHT HALF #0
003466	022767	000000	175312		CMP #0,ANS3	;CHECK LEFT HALF OF INTEGER
003474	001401			BEQ	.+4	
003476	104004			HLT+4		;LEFT HALF #0
003500	022767	000000	175302		CMP #0,ANS4	;CHECK RIGHT HALF OF INTEGER
003506	001401			BEQ	.+4	
003510	104004			HLT+4		;RIGHT HALF #0

```

*****
TEST 20 TEST OF MODF FPU INSTRUCTION
40200 0 * 20000 0 = 0 0 & 20000 0
USING ACO FPS = 47400 FEC = N/A
*****

```

003512	104400			SCOPE		
003514	170127	047400		LOFPS	#47400&57760	
003520	172467	000030		LOF	N20,0	;LOAD 40200 INTO ACO
003524	171467	000030		MODF	M20,0	;MODIFY 40200 BY 20000
003530	170200			STFPS	FPS	;STORE FLOATING POINT STATUS
003532	022700	047400		CMP	#47400,FPS	;CHECK FLOATING POINT STATUS
003536	001401			BEQ	+.4	;BRANCH IF OK
003540	104000			HLT		;FPS NOT EQUAL TO 47400
003542	174067	175234		STF	0,ANS1	;STORE FRACTIONAL PART
003546	174167	175234		STF	0!1,ANS3	;STORE INTEGER PART
003552	000404			BR	020	
003554	040200	000000	N20:	40200,0		
003560	020000	000000	M20:	20000,0		
003564	022767	020000	175210	020:	CMP #20000,ANS1	;CHECK LEFT HALF OF FRACTION
003572	001401			BEQ	+.4	
003574	104004			HLT+4		;LEFT HALF #20000
003576	022767	000000	175200		CMP #0,ANS2	;CHECK RIGHT HALF OF FRACTION
003604	001401			BEQ	+.4	
003606	104004			HLT+4		;RIGHT HALF #0
003610	022767	000000	175170		CMP #0,ANS3	;CHECK LEFT HALF OF INTEGER
003616	001401			BEQ	+.4	
003620	104004			HLT+4		;LEFT HALF #0
003622	022767	000000	175160		CMP #0,ANS4	;CHECK RIGHT HALF OF INTEGER
003630	001401			BEQ	+.4	
003632	104004			HLT+4		;RIGHT HALF #0


```

.....
TEST 2: TEST OF MOOF FPU INSTRUCTION
40600 0 + 37400 0 = 0 0 & 40000 0
USING ACC FPS = 47400 REC = N/A
.....

```

003662	104000	000000			SC00P					
003664	174067	175112			STP	00,ANS1				:LOAD 40600 INTO ACC
003670	174167	175112			STP	01,ANS3				:MODIFY 40600 BY 37400
003674	000404				BR	02:				:STORE FLOATING POINT STATUS
003676	040600	000000								:CHECK FLOATING POINT STATUS
003702	037400	000000								:BRANCH IF OK
003706	022767	040000	175066	021:	CMP	840000,ANS1				:FPS NOT EQUAL TO 47400
003714	001401				BEG	+.4				:STORE FRACTIONAL PART
003716	104004				HLT	+.4				:STORE INTEGER PART
003720	022767	000000	175056		CMP	80,ANS2				
003726	001401				BEG	+.4				
003730	104004				HLT	+.4				
003732	022767	000000	175046		CMP	80,ANS3				
003740	001401				BEG	+.4				
003742	104004				HLT	+.4				
003744	022767	000000	175036		CMP	80,ANS4				
003752	001401				BEG	+.4				
003754	104004				HLT	+.4				

:TEST 22 TEST OF MODF FPU INSTRUCTION
: 40740 0 + 40500 0 = 41250 0 & 0 0
: USING ACO FPS = 47404 FEC = N/A

003756	104400			SCOPE			
003760	170127	047400		LDFPS	#47404&57760		
003764	172467	000030		LDF	M22.0	:LOAD 40740 INTO ACO	
003770	171467	000030		MODF	M22.0	:MODIFY 40740 BY 40500	
003774	170200			STFPS	FPS	:STORE FLOATING POINT STATUS	
003776	022700	047404		CMP	#47404.FPS	:CHECK FLOATING POINT STATUS	
004002	001401			BEG	+.4	:BRANCH IF OK	
004004	104000			HLT		:FPS NOT EQUAL TO 47404	
004006	174067	174770		STF	0.ANS1	:STORE FRACTIONAL PART	
004012	174167	174770		STF	0!1.ANS3	:STORE INTEGER PART	
004016	000404			BR	022		
004020	040740	000000	M22:	40740.C			
004024	040500	000000	M22:	40500.0			
004030	022767	000000	174744	022:	CMP	#0.ANS1	:CHECK LEFT HALF OF FRACTION
004036	001401			BEG	+.4		
004040	104004			HLT	+.4	:LEFT HALF #0	
004042	022767	000000	174734	CMP	#0.ANS2	:CHECK RIGHT HALF OF FRACTION	
004050	001401			BEG	+.4		
004052	104004			HLT	+.4	:RIGHT HALF #0	
004054	022767	041250	174724	CMP	#41250.ANS3	:CHECK LEFT HALF OF INTEGER	
004062	001401			BEG	+.4		
004064	104004			HLT	+.4	:LEFT HALF #41250	
004066	022767	000000	174714	CMP	#0.ANS4	:CHECK RIGHT HALF OF INTEGER	
004074	001401			BEG	+.4		
004076	104004			HLT	+.4	:RIGHT HALF #0	

```

*****
:TEST 23 TEST OF MODD FPU INSTRUCTION
:      0 0 0 0 * 0 0 0 0 =
:      0 0 0 0 & 0 0 0 0
:      USING ACO      FPS = 47604      FEC = N/A
*****

```

```

004100 104400
004102 170127 047600
004106 172467 000030
004112 171467 000034
004116 170200
004120 022700 047604
004124 001401
004126 104000
SCOPE
LDFPS #47604&57760
LDD N23.0 ;LOAD 0 INTO ACO
MODD M23.0 ;MODIFY 0 BY 0
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #47604.FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 47604

004130 174067 174646
004134 174167 174652
004140 000410
STD 0,ANS1 ;STORE FRACTIONAL PART
STD 0,ANS5 ;STORE INTEGER PART
BR 023

004142 000000 000000 000000 N23: 0,0,0,0
004150 000000
004152 000000 000000 000000 M23: 0,0,0,0
004160 000000
004162 022767 000000 174612 023: CMP #0,ANS1 ;CHECK FIRST WORD OF FRACTION
004170 001401 BEQ .+4 ;FIRST # 0
004172 104010 HLT+8.

004174 022767 000000 174602 CMP #0,ANS2 ;CHECK SECOND WORD
004202 001401 BEQ .+4 ;SECOND # 0
004204 104010 HLT+8.

004206 022767 000000 174572 CMP #0,ANS3 ;CHECK THIRD WORD
004214 001401 BEQ .+4 ;THIRD # 0
004216 104010 HLT+8.

004220 022767 000000 174562 CMP #0,ANS4 ;CHECK FORTH
004226 001401 BEQ .+4 ;FORTH # 0
004230 104010 HLT+8.

004232 022767 000000 174552 CMP #0,ANS5 ;CHECK FIRST WORD OF INTEGER
004240 001401 BEQ .+4 ;FIRST # 0
004242 104010 HLT+8.

004244 022767 000000 174542 CMP #0,ANS6 ;CHECK SECOND WORD
004252 001401 BEQ .+4 ;SECOND # 0
004254 104010 HLT+8.

004256 022767 000000 174532 CMP #0,ANS7 ;CHECK THIRD WORD
004264 001401 BEQ .+4 ;THIRD # 0
004266 104010 HLT+8.

004270 022767 000000 174522 CMP #0,ANS8 ;CHECK FORTH WORD
004276 001401 BEQ .+4 ;FORTH # 0
004280 104010 HLT+8.

```

E03

NOEC-...-00FP-B
00FP.B.F... EST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 30

```

*****
TEST 24 TEST OF MODD FPU INSTRUCTION
0 0 0 0 * 40200 0 0 0 =
0 0 0 0 & 0 0 0 0
USING ACO FPS = 47604 FEC = N/A
*****

```

004302	104400				SCOPE		
004304	170127	047600			LDFPS	#47604&5776C	
004310	172467	000030			LDD	M24,0	:LOAD 0 INTO ACO
004314	171467	000034			MODD	M24,0	:MODIFY 0 BY 40200
004320	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
004322	022700	047604			CMP	#47604.FPS	:CHECK FLOATING POINT STATUS
004326	001401				BEG	+.4	:BRANCH IF OK
004330	104000				HLT		:FPS NOT EQUAL TO 47604
004332	174067	174444			STD	0,ANS1	:STORE FRACTIONAL PART
004336	174167	174450			STI	0!1,ANS5	:STORE INTERGER PART
004342	000410				BR	024	
004344	000000	000000	000000	M24:	0,0,0,0		
004352	000000						
004354	040200	000000	000000	M24:	40200,0,0,0		
004362	000000						
004364	022767	000000	174410	024:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
004372	001401				BEG	+.4	
004374	104010				HLT+8.		:FIRST # 0
004376	022767	000000	174400		CMP	#0,ANS2	:CHECK SECOND WORD
004404	001401				BEG	+.4	
004406	104010				HLT+8.		:SECOND # 0
004410	022767	000000	174370		CMP	#0,ANS3	:CHECK THIRD WORD
004416	001401				BEG	+.4	
004420	104010				HLT+8.		:THIRD # 0
004422	022767	000000	174360		CMP	#0,ANS4	:CHECK FORTH
004430	001401				BEG	+.4	
004432	104010				HLT+8.		:FORTH # 0
004434	022767	000000	174350		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER
004442	001401				BEG	+.4	
004444	104010				HLT+8.		:FIRST # 0
004446	022767	000000	174340		CMP	#0,ANS6	:CHECK SECOND WORD
004454	001401				BEG	+.4	
004456	104010				HLT+8.		:SECOND # 0
004460	022767	000000	174330		CMP	#0,ANS7	:CHECK THIRD WORD
004466	001401				BEG	+.4	
004470	104010				HLT+8.		:THIRD # 0
004472	022767	000000	174320		CMP	#0,ANS8	:CHECK FORTH WORD
004500	001401				BEG	+.4	
004502	104010				HLT+8.		:FORTH # 0

TEST OF MODF AND MODD

F03

```

*****
TEST 25          TEST OF MODD FPU INSTRUCTION
40200 0 0 0 * 0 0 0 0 =
0 0 0 0 & 0 0 0 0
USING ACC      FPS = 47604      FEC = N/A
*****

```

Address	Instruction	Operand 1	Operand 2	Operand 3	Operand 4	Comment
004504	LD	40200				:LOAD 40200 INTO ACC
004506	LD	0				:MODIFY 40200 BY 0
004512	ST	0				:STORE FLOATING POINT STATUS
004516	ST	0				:CHECK FLOATING POINT STATUS
004522	BR	0				:BRANCH IF OK
004524	LD	47604				:FPS NOT EQUAL TO 47604
004530	ST	0				:STORE FRACTIONAL PART
004532	ST	0				:STORE INTERGER PART
004534	BR	025				
004546	LD	40200	000000	000000	N25:	40200,0,0,0
004554	LD	0	000000	000000	M25:	0,0,0,0
004556	LD	0	000000	000000		
004564	LD	0	000000	000000		
004566	LD	022767	000000	174206	O25:	CMP #0,ANS1
004574	LD	001401				:CHECK FIRST WORD OF FRACTION
004576	LD	104010				:FIRST # 0
004600	LD	022767	000000	174176		CMP #0,ANS2
004606	LD	001401				:CHECK SECOND WORD
004610	LD	104010				:SECOND # 0
004612	LD	022767	000000	174166		CMP #0,ANS3
004620	LD	001401				:CHECK THIRD WORD
004622	LD	104010				:THIRD # 0
004624	LD	022767	000000	174156		CMP #0,ANS4
004632	LD	001401				:CHECK FORTH
004634	LD	104010				:FORTH # 0
004636	LD	022767	000000	174146		CMP #0,ANS5
004644	LD	001401				:CHECK FIRST WORD OF INTEGER
004646	LD	104010				:FIRST # 0
004650	LD	022767	000000	174136		CMP #0,ANS6
004656	LD	001401				:CHECK SECOND WORD
004660	LD	104010				:SECOND # 0
004662	LD	022767	000000	174126		CMP #0,ANS7
004670	LD	001401				:CHECK THIRD WORD
004672	LD	104010				:THIRD # 0
004674	LD	022767	000000	174116		CMP #0,ANS8
004702	LD	001401				:CHECK FORTH WORD
004704	LD	104010				:FORTH # 0

```

*****
TEST 26 TEST OF MODD FPU INSTRUCTION
40200 0 0 0 * 40200 0 0 0 =
40200 0 0 0 & 0 0 0 0
USING ACC FPS = 47604 FEC = N/A
*****

```

004706	104400				SCOPE		
004710	170127	047600			LDFPS	#47604&57760	
004714	172467	000030			LDD	M26,0	:LOAD 40200 INTO ACC
004720	171467	000034			MODD	M26,0	:MODIFY 40200 BY 40200
004724	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
004726	022700	047604			CMP	#47604.FPS	:CHECK FLOATING POINT STATUS
004732	001401				BEG	+.4	:BRANCH IF OK
004734	104000				HLT		:FPS NOT EQUAL TO 47604
004736	174067	174040			STD	0,ANS1	:STORE FRACTIONAL PART
004742	174167	174044			STD	0!1,ANS5	:STORE INTERGER PART
004746	000410				BR	026	
004750	040200	000000	000000	N26:	40200,0,0,0		
004756	000000						
004760	040200	000000	000000	M26:	40200,0,0,0		
004766	000000						
004770	022767	000000	174004	026:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
004776	001401				BEG	+.4	
005000	104010				HLT+8.		:FIRST # 0
005002	022767	000000	173774		CMP	#0,ANS2	:CHECK SECOND WORD
005010	001401				BEG	+.4	
005012	104010				HLT+8.		:SECOND # 0
005014	022767	000000	173764		CMP	#0,ANS3	:CHECK THIRD WORD
005022	001401				BEG	+.4	
005024	104010				HLT+8.		:THIRD # 0
005026	022767	000000	173754		CMP	#0,ANS4	:CHECK FORTH
005034	001401				BEG	+.4	
005036	104010				HLT+8.		:FORTH # 0
005040	022767	040200	173744		CMP	#40200,ANS5	:CHECK FIRST WORD OF INTEGER
005046	001401				BEG	+.4	
005050	104010				HLT+8.		:FIRST # 40200
005052	022767	000000	173734		CMP	#0,ANS6	:CHECK SECOND WORD
005060	001401				BEG	+.4	
005062	104010				HLT+8.		:SECOND # 0
005064	022767	000000	173724		CMP	#0,ANS7	:CHECK THIRD WORD
005072	001401				BEG	+.4	
005074	104010				HLT+8.		:THIRD # 0
005076	022767	000000	173714		CMP	#0,ANS8	:CHECK FORTH WORD
005084	001401				BEG	+.4	
005086	104010				HLT+8.		:FORTH # 0

H03

NOEC-11-DCFP-8
DCFP-8-11-EST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 33

```

*****
TEST 27          TEST OF MODD FPU INSTRUCTION
40214 146314 146314 146315 * 40200 0 0 0 =
40200 0 0 0 & 37314 146314 146314 146320
USING ACO      FPS = 47600      FEC = N/A
*****

```

005110	104400				SCOPE		
005112	170127	047600			LDFPS	#47600&57760	
005116	172467	000030			LDC	M27,0	:LOAD 40214 INTO ACO
005122	171467	000034			MODD	M27,0	:MODIFY 40214 BY 40200
005126	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
005130	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
005134	001401				BEG	+.4	:BRANCH IF OK
005136	104000				HLT		:FPS NOT EQUAL TO 47600
005140	174067	173636			STD	0,ANS1	:STORE FRACTIONAL PART
005144	174167	173642			STC	0!1,ANS5	:STORE INTERGER PART
005150	000410				BR	027	
005152	040214	146314	146314	M27:	40214,146314,146314,146315		
005160	146315						
005162	040200	000000	000000	M27:	40200,0,0,0		
005170	000000						
005172	022767	037314	173602	027:	CMP	#37314,ANS1	:CHECK FIRST WORD OF FRACTION
005200	001401				BEG	+.4	
005202	104010				HLT+8.		:FIRST # 37314
005204	022767	146314	173572		CMP	#146314,ANS2	:CHECK SECOND WORD
005212	001401				BEG	+.4	
005214	104010				HLT+8.		:SECOND # 146314
005216	022767	146314	173562		CMP	#146314,ANS3	:CHECK THIRD WORD
005224	001401				BEG	+.4	
005226	104010				HLT+8.		:THIRD # 146314
005230	022767	146320	173552		CMP	#146320,ANS4	:CHECK FORTH
005236	001401				BEG	+.4	
005240	104010				HLT+8.		:FORTH # 146320
005242	022767	040200	173542		CMP	#40200,ANS5	:CHECK FIRST WORD OF INTEGER
005250	001401				BEG	+.4	
005252	104010				HLT+8.		:FIRST # 40200
005254	022767	000000	173532		CMP	#0,ANS6	:CHECK SECOND WORD
005262	001401				BEG	+.4	
005264	104010				HLT+8.		:SECOND # 0
005266	022767	000000	173522		CMP	#0,ANS7	:CHECK THIRD WORD
005274	001401				BEG	+.4	
005276	104010				HLT+8.		:THIRD # 0
005300	022767	000000	173512		CMP	#0,ANS8	:CHECK FORTH WORD
005306	001401				BEG	+.4	
005310	104010				HLT+8.		:FORTH # 0


```

*****
:TEST 30 TEST OF MODD FPU INSTRUCTION
: 40414 146314 146314 146315 * 40200 0 0 0 =
: 40400 0 0 0 & 37514 146314 146314 146320
: USING ACO FPS = 47600 FEC = N/A
*****

```

005312	104400				SCOPE		
005314	170127	047600			LDFPS	#47600&57760	
005320	172467	00003C			LDD	M30,0	:LOAD 40414 INTO ACO
005324	171467	000034			MODD	M30,0	:MODIFY 40414 BY 40200
005330	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
005332	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
005336	001401				BEG	+.4	:BRANCH IF OK
005340	104000				HLT		:FPS NOT EQUAL TO 47600
005342	174067	173434			STD	0,ANS1	:STORE FRACTIONAL PART
005346	174167	173440			STD	0!1,ANS5	:STORE INTERGER PART
005352	000410				BR	030	
005354	040414	146314	146314	N30:		40414,146314,146314,146315	
005362	146315						
005364	040200	000000	000000	M30:		40200,0,0,0	
005372	000000						
005374	022767	037514	173400	030:	CMP	#37514,ANS1	:CHECK FIRST WORD OF FRACTION
005402	001401				BEG	+.4	
005404	104010				HLT+8.		:FIRST # 37514
005406	022767	146314	173370		CMP	#146314,ANS2	:CHECK SECOND WORD
005414	001401				BEG	+.4	
005416	104010				HLT+8.		:SECOND # 146314
005420	022767	146314	173360		CMP	#146314,ANS3	:CHECK THIRD WORD
005426	001401				BEG	+.4	
005430	104010				HLT+8.		:THIRD # 146314
005432	022767	146320	173350		CMP	#146320,ANS4	:CHECK FORTH
005440	001401				BEG	+.4	
005442	104010				HLT+8.		:FORTH # 146320
005444	022767	040400	173340		CMP	#40400,ANS5	:CHECK FIRST WORD OF INTEGER
005452	001401				BEG	+.4	
005454	104010				HLT+8.		:FIRST # 40400
005456	022767	000000	173330		CMP	#0,ANS6	:CHECK SECOND WORD
005464	001401				BEG	+.4	
005466	104010				HLT+8.		:SECOND # 0
005470	022767	000000	173320		CMP	#0,ANS7	:CHECK THIRD WORD
005476	001401				BEG	+.4	
005500	104010				HLT+8.		:THIRD # 0
005502	022767	000000	173310		CMP	#0,ANS8	:CHECK FORTH WORD
005510	001401				BEG	+.4	
005512	104010				HLT+8.		:FORTH # 0

```

*****
:TEST 31 TEST OF MODD FPU INSTRUCTION
: 40523 31463 31463 31463 * 40200 0 0 0 =
: 40500 0 0 0 & 37631 114631 114631 114630
: USING ACO FPS = 47600 FEC = N/A
*****

```

005514	104400				SCOPE		
005516	170127	047600			LDFPS	#47600&57760	
005522	172467	000030			LDD	M31,0	:LOAD 40523 INTO ACO
005526	171467	000034			MODD	M31,0	:MODIFY 40523 BY 40200
005532	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
005534	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
005540	001401				BEQ	+.4	:BRANCH IF OK
005542	104000				HLT		:FPS NOT EQUAL TO 47600
005544	174067	173232			STD	0,ANS1	:STORE FRACTIONAL PART
005550	174157	173236			STD	0!1,ANS5	:STORE INTERGER PART
005554	000410				BR	031	
005556	040523	031463	031463	M31:		40523,31463,31463,31463	
005564	031463						
005566	040200	000000	000000	M31:		40200,0,0,0	
005574	000000						
005576	022767	037631	173176	031:	CMP	#37631,ANS1	:CHECK FIRST WORD OF FRACTION
005604	001401				BEQ	+.4	
005606	104010				HLT+8.		:FIRST # 37631
005610	022767	114631	173166		CMP	#114631,ANS2	:CHECK SECOND WORD
005616	001401				BEQ	+.4	
005620	104010				HLT+8.		:SECOND # 114631
005622	022767	114631	173156		CMP	#114631,ANS3	:CHECK THIRD WORD
005630	001401				BEQ	+.4	
005632	104010				HLT+8.		:THIRD # 114631
005634	022767	114630	173146		CMP	#114630,ANS4	:CHECK FORTH
005642	001401				BEQ	+.4	
005644	104010				HLT+8.		:FORTH # 114630
005646	022767	040500	173136		CMP	#40500,ANS5	:CHECK FIRST WORD OF INTEGER
005654	001401				BEQ	+.4	
005656	104010				HLT+8.		:FIRST # 40500
005660	022767	000000	173126		CMP	#0,ANS6	:CHECK SECOND WORD
005666	001401				BEQ	+.4	
005670	104010				HLT+8.		:SECOND # 0
005672	022767	000000	173116		CMP	#0,ANS7	:CHECK THIRD WORD
005700	001401				BEQ	+.4	
005702	104010				HLT+8.		:THIRD # 0
005704	022767	000000	173106		CMP	#0,ANS8	:CHECK FORTH WORD
005712	001401				BEQ	+.4	
005714	104010				HLT+8.		:FORTH # 0

```

*****
:TEST 32 TEST OF MODD FPU INSTRUCTION
: 40660 0 0 0 * 40200 0 0 0 =
: 40640 0 0 0 & 40000 0 0 0
: USING ACO FPS = 47600 FEC = N/A
*****

```

005716	104400				SCOPE		
005720	170127	047600			LDFPS	#47600&57760	
005724	172467	000030			LDD	N32,0	:LOAD 40660 INTO ACO
005730	171467	000034			MODD	M32,0	:MODIFY 40660 BY 40200
005734	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
005736	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
005742	001401				BEG	+.4	:BRANCH IF OK
005744	104007				HLT		:FPS NOT EQUAL TO 47600
005746	174067	173030			STD	C,ANS1	:STORE FRACTIONAL PART
005752	174167	173034			STD	0!1,ANS5	:STORE INTERGER PART
005756	000410				BR	032	
005760	040660	000000	000000	N32:		40660,0,0,0	
005766	000000						
005770	040200	000000	000000	M32:		40200,0,0,0	
005776	000000						
006000	022767	040000	172774	032:	CMP	#40000,ANS1	:CHECK FIRST WORD OF FRACTION
006006	001401				BEG	+.4	
006010	104010				HLT+8.		:FIRST # 40000
006012	022767	000000	172764		CMP	#0,ANS2	:CHECK SECOND WORD
006020	001401				BEG	+.4	
006022	104010				HLT+8.		:SECOND # 0
006024	022767	000000	172754		CMP	#0,ANS3	:CHECK THIRD WORD
006032	001401				BEG	+.4	
006034	104010				HLT+8.		:THIRD # 0
006036	022767	000000	172744		CMP	#0,ANS4	:CHECK FORTH
006044	001401				BEG	+.4	
006046	104010				HLT+8.		:FORTH # 0
006050	022767	040640	172734		CMP	#40640,ANS5	:CHECK FIRST WORD OF INTEGER
006056	001401				BEG	+.4	
006060	104010				HLT+8.		:FIRST # 40640
006062	022767	000000	172724		CMP	#0,ANS6	:CHECK SECOND WORD
006070	001401				BEG	+.4	
006072	104010				HLT+8.		:SECOND # 0
006074	022767	000000	172714		CMP	#0,ANS7	:CHECK THIRD WORD
006102	001401				BEG	+.4	
006104	104010				HLT+8.		:THIRD # 0
006106	022767	000000	172704		CMP	#0,ANS8	:CHECK FORTH WORD
006114	001401				BEG	+.4	
006116	104010				HLT+8.		:FORTH # 0

```

*****
:TEST 33 TEST OF MODD FPU INSTRUCTION
: 140200 0 0 0 * 40200 0 0 0 =
: 140200 0 0 0 & 0 0 0 0
: USING ACO FPS = 47604 FEC = N/A
*****

```

```

006120 104400
006122 170127 047600
006126 172467 000030
006132 171467 000034
006136 170200
006140 022700 047604
006144 001401
006146 104000
SCOPE
LDFPS #47604&57760
LDD N33,0 ;LOAD 140200 INTO ACO
MODD M33,0 ;MODIFY 140200 BY 40200
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #47604,FPS ;CHECK FLOATING PCINT STATJS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 47604

006150 174067 172626
006154 174167 172632
006160 000410
STD 0,ANS1 ;STORE FRACTIONAL PART
STD 0!1,ANS5 ;STORE INTERGER PART
BR 033

006162 140200 000000 000000 N33: 140200,0,0,0
006170 000000
006172 040200 000000 000000 M33: 40200,0,0,0
006200 000000
006202 022767 000000 172572 033: CMP #0,ANS1 ;CHECK FIRST WORD OF FRACTION
006210 001401 BEQ .+4 ;FIRST # 0
006212 104010 HLT+8.

006214 022767 000000 172562 CMP #0,ANS2 ;CHECK SECOND WORD
006222 001401 BEQ .+4 ;SECOND # 0
006224 104010 HLT+8.

006226 022767 000000 172552 CMP #0,ANS3 ;CHECK THIRD WORD
006234 001401 BEQ .+4 ;THIRD # 0
006236 104010 HLT+8.

006240 022767 000000 172542 CMP #0,ANS4 ;CHECK FORTH
006246 001401 BEQ .+4 ;FORTH # 0
006250 104010 HLT+8.

006252 022767 140200 172532 CMP #140200,ANS5 ;CHECK FIRST WORD OF INTEGER
006260 001401 BEQ .+4 ;FIRST # 140200
006262 104010 HLT+8.

006264 022767 000000 172522 CMP #0,ANS6 ;CHECK SECOND WORD
006272 001401 BEQ .+4 ;SECOND # 0
006274 104010 HLT+8.

006276 022767 000000 172512 CMP #0,ANS7 ;CHECK THIRD WORD
006304 001401 BEQ .+4 ;THIRD # 0
006306 104010 HLT+8.

006310 022767 000000 172502 CMP #0,ANS8 ;CHECK FORTH WORD
006316 001401 BEQ .+4 ;FORTH # 0
006320 104010 HLT+8.

```

```

*****
:TEST 34 TEST OF MODD FPU INSTRUCTION
:140214 146314 146314 146315 * 40200 0 0 0 =
:140200 0 0 0 & 137314 146314 146314 146320
:USING ACO FPS = 47610 FEC = N/A
*****

```

```

006322 104400 SCOPE
006324 170127 047600 LDFPS #47610&57760
006330 172467 000030 LDD N34,0 ;LOAD 140214 INTO ACO
006334 171467 000034 MODD M34,0 ;MODIFY 140214 BY 40200
006340 170200 STFPS FPS ;STORE FLOATING POINT STATUS
006342 022700 047610 CMP #47610,FPS ;CHECK FLOATING POINT STATUS
006346 001401 BEQ .+4 ;BRANCH IF OK
006350 104000 HLT ;FPS NOT EQUAL TO 47610

006352 174067 172424 STD 0,ANS1 ;STORE FRACTIONAL PART
006356 174167 172430 STD 0!1,ANS5 ;STORE INTERGER PART
006362 000410 BR 034

006364 140214 146314 146314 N34: 140214,146314,146314,146315
006372 146315
006374 040200 000000 000000 M34: 40200,0,0,0
006402 000000
006404 022767 137314 172370 034: CMP #137314,ANS1 ;CHECK FIRST WORD OF FRACTION
006412 001401 BEQ .+4 ;FIRST # 137314
006414 104010 HLT+8.

006416 022767 146314 172360 CMP #146314,ANS2 ;CHECK SECOND WORD
006424 001401 BEQ .+4 ;SECOND # 146314
006426 104010 HLT+8.

006430 022767 146314 172350 CMP #146314,ANS3 ;CHECK THIRD WORD
006436 001401 BEQ .+4 ;THIRD # 146314
006440 104010 HLT+8.

006442 022767 146320 172340 CMP #146320,ANS4 ;CHECK FORTH
006450 001401 BEQ .+4 ;FORTH # 146320
006452 104010 HLT+8.

006454 022767 140200 172330 CMP #140200,ANS5 ;CHECK FIRST WORD OF INTEGER
006462 001401 BEQ .+4 ;FIRST # 140200
006464 104010 HLT+8.

006466 022767 000000 172320 CMP #0,ANS6 ;CHECK SECOND WORD
006474 001401 BEQ .+4 ;SECOND # 0
006476 104010 HLT+8.

006500 022767 000000 172310 CMP #0,ANS7 ;CHECK THIRD WORD
006506 001401 BEQ .+4 ;THIRD # 0
006510 104010 HLT+8.

006512 022767 000000 172300 CMP #0,ANS8 ;CHECK FORTH WORD
006520 001401 BEQ .+4 ;FORTH # 0
006522 104010 HLT+8.

```

```

*****
TEST 35 TEST OF MODD FPU INSTRUCTION
140414 146314 146314 146315 * 40200 0 0 0 =
140400 0 0 0 & 137514 146314 146314 146320
USING ACO FPS = 47610 FEC = N/A
*****

```

006524	104400				SCOPE		
006526	170127	047600			LDFPS	#47610&57760	
006532	172467	000030			LDD	N35,0	;LOAD 140414 INTO ACO
006536	171467	000034			MODD	M35,0	;MODIFY 140414 BY 40200
006542	170200				STFPS	FPS	;STORE FLOATING POINT STATUS
006544	022700	047610			CMP	#47610,FPS	;CHECK FLOATING POINT STATUS
006550	001401				BEQ	+.4	;BRANCH IF OK
006552	104000				HLT		;FPS NOT EQUAL TO 47610
006554	174067	172222			STD	0,ANS1	;STORE FRACTIONAL PART
006560	174167	172226			STD	0!1,ANS5	;STORE INTERGER PART
006564	000410				BR	035	
006566	140414	146314	146314	N35:		140414,146314,146314,146315	
006574	146315						
006576	040200	000000	000000	M35:		40200,0,0.0	
006604	000000						
006606	022767	137514	172166	035:	CMP	#137514,ANS1	;CHECK FIRST WORD OF FRACTION
006614	001401				BEQ	+.4	
006616	104010				HLT+8.		;FIRST # 137514
006620	022767	146314	172156		CMP	#146314,ANS2	;CHECK SECOND WORD
006626	001401				BEQ	+.4	
006630	104010				HLT+8.		;SECOND # 146314
006632	022767	146314	172146		CMP	#146314,ANS3	;CHECK THIRD WORD
006640	001401				BEQ	+.4	
006642	104010				HLT+8.		;THIRD # 146314
006644	022767	146320	172136		CMP	#146320,ANS4	;CHECK FORTH
006652	001401				BEQ	+.4	
006654	104010				HLT+8.		;FORTH # 146320
006656	022767	140400	172126		CMP	#140400,ANS5	;CHECK FIRST WORD OF INTEGER
006664	001401				BEQ	+.4	
006666	104010				HLT+8.		;FIRST # 140400
006670	022767	000000	172116		CMP	#0,ANS6	;CHECK SECOND WORD
006676	001401				BEQ	+.4	
006700	104010				HLT+8.		;SECOND # 0
006702	022767	000000	172106		CMP	#0,ANS7	;CHECK THIRD WORD
006710	001401				BEQ	+.4	
006712	104010				HLT+8.		;THIRD # 0
006714	022767	000000	172076		CMP	#0,ANS8	;CHECK FORTH WORD
006722	001401				BEQ	+.4	
006724	104010				HLT+8.		;FORTH # 0

```

*****
TEST 36          TEST OF 4000 FPL INSTRUCTION
:140523 3:463 31463 31463 * 40200 0 0 0 =
:140500 0 0 0 8 :137631 114631 114631 114630
USING ACC      FPS = 47610      FEC = N/A
*****

```

006750	000000	047600			SCOPE		
006751	000000	000030			FPS	047610857760	
006752	000000	000024			FPS	M36.0	:LOAD 140523 INTO ACC
006753	000000	047610			4000	M36.0	:MODIFY 140523 BY 40200
006754	104000				SFPS	FPS	:STORE FLOATING POINT STATUS
					CMP	047610.FPS	:CHECK FLOATING POINT STATUS
					BEG	.+4	:BRANCH IF OK
					HLT		:FPS NOT EQUAL TO 47610
006756	174067	172020			STD	0,ANS1	:STORE FRACTIONAL PART
006762	174167	172024			STD	0.1,ANS5	:STORE INTEGER PART
006766	000410				BR	036	
006770	140523	031463	031463	M36:		140523,3:463,31463,31463	
006776	031463						
007000	040200	000000	000000	M36:		40200,0,0,0	
007006	000000						
007010	022767	137631	171764	036:	CMP	0137631,ANS1	:CHECK FIRST WORD OF FRACTION
007016	001401				BEG	.+4	
007020	104010				HLT	+8.	:FIRST # 137631
007022	022767	114631	171754		CMP	0114631,ANS2	:CHECK SECOND WORD
007030	001401				BEG	.+4	
007032	104010				HLT	+8.	:SECOND # 114631
007034	022767	114631	171744		CMP	0114631,ANS3	:CHECK THIRD WORD
007042	001401				BEG	.+4	
007044	104010				HLT	+8.	:THIRD # 114631
007046	022767	114630	171734		CMP	0114630,ANS4	:CHECK FORTH
007054	001401				BEG	.+4	
007056	104010				HLT	+8.	:FORTH # 114630
007060	022767	140500	171724		JMP	0140500,ANS5	:CHECK FIRST WORD OF INTEGER
007066	001401				BEG	.+4	
007070	104010				HLT	+8.	:FIRST # 140500
007072	022767	000000	171714		CMP	00,ANS6	:CHECK SECOND WORD
007100	001401				BEG	.+4	
007102	104010				HLT	+8.	:SECOND # 0
007104	022767	000000	171704		CMP	00,ANS7	:CHECK THIRD WORD
007112	001401				BEG	.+4	
007114	104010				HLT	+8.	:THIRD # 0
007116	022767	000000	171674		CMP	00,ANS8	:CHECK FORTH WORD
007124	001401				BEG	.+4	
007126	104010				HLT	+8.	:FORTH # 0


```

*****
TEST 37          TEST OF MODD FPU INSTRUCTION
140660 0 0 0 + 40200 0 0 0 =
140640 0 0 0 & 140000 0 0 0
USING ACO      FPS = 47610      FEC = N/A
*****

```

```

007130 104400
007132 170127 047600
007136 172467 000030
007142 171467 000034
007146 170200
007150 022700 047610
007154 001401
007156 104000

SCOPE
LDFPS #47610&57760
LDD M37.0 :LOAD 140660 INTO ACO
MODD M37.0 :MODIFY 140660 BY 40200
STFPS FPS :STORE FLOATING POINT STATUS
CMP #47610.FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 47610

007160 174067 171616 STD 0,ANS1 :STORE FRACTIONAL PART
007164 174167 171522 STD 0!1,ANS5 :STORE INTERGER PART
007170 000410 BP 037

007172 140660 000000 000000 N37: 140660.0.0.0
007200 000000
007202 040200 000000 000000 M37: 40200.0.0.0
007210 000000
007212 022767 140000 171562 037: CMP #140000,ANS1 :CHECK FIRST WORD OF FRACTION
007220 001401 BEQ .+4
007222 104010 HLT+8. :FIRST # 140000

007224 022767 000000 171552 CMP #0,ANS2 :CHECK SECOND WORD
007232 001401 BEQ .+4
007234 104010 HLT+8. :SECOND # 0

007236 022767 000000 171542 CMP #0,ANS3 :CHECK THIRD WORD
007244 001401 BEQ .+4
007246 104010 HLT+8. :THIRD # 0

007250 022767 000000 171532 CMP #0,ANS4 :CHECK FORTH
007256 001401 BEQ .+4
007260 104010 HLT+8. :FORTH # 0

007262 022767 140640 171522 CMP #140640,ANS5 :CHECK FIRST WORD OF INTEGER
007270 001401 BEQ .+4
007272 104010 HLT+8. :FIRST # 140640

007274 022767 000000 171512 CMP #0,ANS6 :CHECK SECOND WORD
007302 001401 BEQ .+4
007304 104010 HLT+8. :SECOND # 0

007306 022767 000000 171502 CMP #0,ANS7 :CHECK THIRD WORD
007314 001401 BEQ .+4
007316 104010 HLT+8. :THIRD # 0

007320 022767 000000 171472 CMP #0,ANS8 :CHECK FORTH WORD
007326 001401 BEQ .+4
007330 104010 HLT+8. :FORTH # 0

```

```

*****
TEST 40 TEST OF MODD FPU INSTRUCTION
52525 52525 52525 52525 * 40200 0 0 0 =
52525 52525 52525 40000 & 37652 124000 0 0
USING ACO FPS = 47600 FEC = N/A
*****

```

007332	104400				SCOPE		
007334	170127	047600			LDFPS	#47600&57760	
007340	172467	000030			LDD	M40,0	:LOAD 52525 INTO ACO
007344	171467	000034			MODD	M40,0	:MODIFY 52525 BY 40200
007350	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
007352	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
007356	001401				BEQ	+.4	:BRANCH IF OK
007360	104000				HLT		:FPS NOT EQUAL TO 47600
007362	174067	171414			STD	0,ANS1	:STORE FRACTIONAL PART
007366	174167	171420			STD	0!1,ANS5	:STORE INTEGER PART
007372	000410				BR	040	
007374	052525	052525	052525	N40:	52525,52525,52525,52525		
007402	052525						
007404	040200	000000	000000	M40:	40200,0,0,0		
007412	000000						
007414	022767	037652	171360	040:	CMP	#37652,ANS1	:CHECK FIRST WORD OF FRACTION
007422	001401				BEQ	+.4	
007424	104010				HLT+8.		:FIRST # 37652
007426	022767	124000	171350		CMP	#124000,ANS2	:CHECK SECOND WORD
007434	001401				BEQ	+.4	
007436	104010				HLT+8.		:SECOND # 124000
007440	022767	000000	171340		CMP	#0,ANS3	:CHECK THIRD WORD
007446	001401				BEQ	+.4	
007450	104010				HLT+8.		:THIRD # 0
007452	022767	000000	171330		CMP	#0,ANS4	:CHECK FORTH
007460	001401				BEQ	+.4	
007462	104010				HLT+8.		:FORTH # 0
007464	022767	052525	171320		CMP	#52525,ANS5	:CHECK FIRST WORD OF INTEGER
007472	001401				BEQ	+.4	
007474	104010				HLT+8.		:FIRST # 52525
007476	022767	052525	171310		CMP	#52525,ANS6	:CHECK SECOND WORD
007504	001401				BEQ	+.4	
007506	104010				HLT+8.		:SECOND # 52525
007510	022767	052525	171300		CMP	#52525,ANS7	:CHECK THIRD WORD
007516	001401				BEQ	+.4	
007520	104010				HLT+8.		:THIRD # 52525
007522	022767	040000	171270		CMP	#40000,ANS8	:CHECK FORTH WORD
007530	001401				BEQ	+.4	
007532	104010				HLT+8.		:FORTH # 40000

```

*****
:TEST 41 TEST OF MODD FPU INSTRUCTION
: 125252 125252 125252 125252 * 40200 0 0 0 =
: 0 0 0 0 & 125252 125252 125252 125252
: USING ACO FPS = 47610 FEC = N/A
*****

```

007534	104400				SCOPE		
007536	170127	047600			LDFPS	#47610&57760	
007542	172467	000030			LDD	M41,0	:LOAD 125252 INTO ACO
007546	171467	000034			MODD	M41,0	:MODIFY 125252 BY 40200
007552	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
007554	022700	047610			CMP	#47610,FPS	:CHECK FLOATING POINT STATUS
007560	001401				BEG	+.4	:BRANCH IF OK
007562	104000				HLT		:FPS NOT EQUAL TO 47610
007564	174067	171212			STD	0,ANS1	:STORE FRACTIONAL PART
007570	174167	171216			STD	0!1,ANSS	:STORE INTERGER PART
007574	000410				BR	041	
007576	125252	125252	125252	N41:		125252,125252,125252,125252	
007604	125252						
007606	040200	000000	000000	M41:		40200,0,0,0	
007614	000000						
007616	022767	125252	171156	041:	CMP	#125252,ANS1	:CHECK FIRST WORD OF FRACTION
007624	001401				BEG	+.4	
007626	104010				HLT+8.		:FIRST # 125252
007630	022767	125252	171146		CMP	#125252,ANS2	:CHECK SECOND WORD
007636	001401				BEG	+.4	
007640	104010				HLT+8.		:SECOND # 125252
007642	022767	125252	171136		CMP	#125252,ANS3	:CHECK THIRD WORD
007650	001401				BEG	+.4	
007652	104010				HLT+8.		:THIRD # 125252
007654	022767	125252	171126		CMP	#125252,ANS4	:CHECK FORTH
007662	001401				BEG	+.4	
007664	104010				HLT+8.		:FORTH # 125252
007666	022767	000000	171116		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER
007674	001401				BEG	+.4	
007676	104010				HLT+8.		:FIRST # 0
007700	022767	000000	171106		CMP	#0,ANS6	:CHECK SECOND WORD
007706	001401				BEG	+.4	
007710	104010				HLT+8.		:SECOND # 0
007712	022767	000000	171076		CMP	#0,ANS7	:CHECK THIRD WORD
007720	001401				BEG	+.4	
007722	104010				HLT+8.		:THIRD # 0
007724	022767	000000	171066		CMP	#0,ANS8	:CHECK FORTH WORD
007732	001401				BEG	+.4	
007734	104010				HLT+8.		:FORTH # 0

F04

```

*****
:TEST 42                                TEST OF MODD FPU INSTRUCTION
:      0 0 0 0 * 0 0 0 0 =
:      0 0 0 0 & 0 0 0 0
:      USING AC2                        FPS = 47604      FEC = N/A
*****

```

007736	104400				SCOPE		
007740	170127	047600			LDFPS	#47604&57760	
007744	172667	000030			LDD	M42,2	:LOAD 0 INTO AC2
007750	171667	000034			MODD	M42,2	:MODIFY 0 BY 0
007754	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
007756	022700	047604			CMP	#47604,FPS	:CHECK FLOATING POINT STATUS
007762	001401				BEG	+.4	:BRANCH IF OK
007764	104000				HLT		:FPS NOT EQUAL TO 47604
007766	174267	171010			STD	2,ANS1	:STORE FRACTIONAL PART
007772	174367	171014			STD	2,1,ANS5	:STORE INTERGER PART
007776	000410				BR	042	
010000	000000	000000	000000	M42:	O.O.C.C		
010006	000000						
010010	000000	000000	000000	M42:	O.O.C.C		
010016	000000						
010020	022767	000000	170754	042:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
010026	001401				BEG	+.4	
010030	104010				HLT+8.		:FIRST # 0
010032	022767	000000	170744		CMP	#0,ANS2	:CHECK SECOND WORD
010040	001401				BEG	+.4	
010042	104010				HLT+8.		:SECOND # 0
010044	022767	000000	170734		CMP	#0,ANS3	:CHECK THIRD WORD
010052	001401				BEG	+.4	
010054	104010				HLT+8.		:THIRD # 0
010056	022767	000000	170724		CMP	#0,ANS4	:CHECK FORTH
010064	001401				BEG	+.4	
010066	104010				HLT+8.		:FORTH # 0
010070	022767	000000	170714		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER
010076	001401				BEG	+.4	
010080	104010				HLT+8.		:FIRST # 0
010102	022767	000000	170704		CMP	#0,ANS6	:CHECK SECOND WORD
010110	001401				BEG	+.4	
010112	104010				HLT+8.		:SECOND # 0
010114	022767	000000	170674		CMP	#0,ANS7	:CHECK THIRD WORD
010122	001401				BEG	+.4	
010124	104010				HLT+8.		:THIRD # 0
010126	022767	000000	170664		CMP	#0,ANS8	:CHECK FORTH WORD
010134	001401				BEG	+.4	
010136	104010				HLT+8.		:FORTH # 0

MACY11 27.732) 03-SEP-76 14:57 PAGE 45

```

*****
:TEST 43 TEST OF MODD FPU INSTRUCTION
: 0 0 0 0 * 40200 0 0 0 =
: 0 0 0 0 & 0 0 0 0
: USING AC2 FPS = 47604 FEC = N/A
*****

```

010140	104400				SCOPE		
010142	170127	047600			LDFPS	#47604&57760	
010146	172667	000030			LDD	M43.2	:LOAD 0 INTO AC2
010152	171567	000034			MODD	M43.2	:MODIFY 0 BY 40200
010156	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
010160	022700	047604			CMP	#47604.FPS	:CHECK FLOATING POINT STATUS
010164	001401				BEG	+.4	:BRANCH IF OK
010166	104000				HLT		:FPS NOT EQUAL TO 47604
010170	174267	170606			STD	2,ANS1	:STORE FRACTIONAL PART
010174	174367	170612			STD	2!1,ANS5	:STORE INTERGER PART
010200	000410				BR	043	
010202	000000	000000	000000	N43:	0.0.0.0		
010210	000000						
010212	040200	000000	000000	M43:	40200.0.0.0		
010220	000000						
010222	022767	000000	170552	043:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
010230	001401				BEG	+.4	
010232	104010				HLT+8.		:FIRST # 0
010234	022767	000000	170542		CMP	#0,ANS2	:CHECK SECOND WORD
010242	001401				BEG	+.4	
010244	104010				HLT+8.		:SECOND # 0
010246	022767	000000	170532		CMP	#0,ANS3	:CHECK THIRD WORD
010254	001401				BEG	+.4	
010256	104010				HLT+8.		:THIRD # 0
010260	022767	000000	170522		CMP	#0,ANS4	:CHECK FORTH
010266	001401				BEG	+.4	
010270	104010				HLT+8.		:FORTH # 0
010272	022767	000000	170512		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER
010300	001401				BEG	+.4	
010302	104010				HLT+8.		:FIRST # 0
010304	022767	000000	170502		CMP	#0,ANS6	:CHECK SECOND WORD
010312	001401				BEG	+.4	
010314	104010				HLT+8.		:SECOND # 0
010316	022767	000000	170472		CMP	#0,ANS7	:CHECK THIRD WORD
010324	001401				BEG	+.4	
010326	104010				HLT+8.		:THIRD # 0
010330	022767	000000	170462		CMP	#0,ANS8	:CHECK FORTH WORD
010336	001401				BEG	+.4	
010340	104010				HLT+8.		:FORTH # 0

H04

```

*****
:TEST 44 TEST OF MODC FPU INSTRUCTION
: 40200 0 0 0 * 0 0 0 0 =
: 0 0 0 0 & 0 0 0 0
: USING AC2 FPS = 47604 FEC = N/A
*****

```

010342	104400				SCOPE			
010344	170127	047600			LDFPS	#47604&57760		
010350	172667	000030			LDD	M44,2	:LOAD 40200 INTO AC2	
010354	171667	000034			MODD	M44,2	:MODIFY 40200 BY 0	
010360	170200				STFPS	FPS	:STORE FLOATING POINT STATUS	
010362	022700	047604			CMP	#47604,FPS	:CHECK FLOATING POINT STATUS	
010366	001401				BEG	+.4	:BRANCH IF OK	
010370	104000				HLT		:FPS NOT EQUAL TO 47604	
010372	174267	170404			STD	2,ANS1	:STORE FRACTIONAL PART	
010376	174367	170410			STC	2!1,ANS5	:STORE INTERGER PART	
010402	000410				BR	044		
010404	040200	000000	000000	N44:	40200,0,0,0			
010412	000000							
010414	000000	000000	000000	M44:	0,0,0,0			
010422	000000							
010424	022767	000000	170350	044:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION	
010432	001401				BEG	+.4		
010434	104010				HLT+8.		:FIRST # 0	
010436	022767	000000	170340		CMP	#0,ANS2	:CHECK SECOND WORD	
010444	001401				BEG	+.4		
010446	104010				HLT+8.		:SECOND # 0	
010450	022767	000000	170330		CMP	#0,ANS3	:CHECK THIRD WORD	
010456	001401				BEG	+.4		
010460	104010				HLT+8.		:THIRD # 0	
010462	022767	000000	170320		CMP	#0,ANS4	:CHECK FORTH	
010470	001401				BEG	+.4		
010472	104010				HLT+8.		:FORTH # 0	
010474	022767	000000	170310		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER	
010502	001401				BEG	+.4		
010504	104010				HLT+8.		:FIRST # 0	
010506	022767	000000	170300		CMP	#0,ANS6	:CHECK SECOND WORD	
010514	001401				BEG	+.4		
010516	104010				HLT+8.		:SECOND # 0	
010520	022767	000000	170270		CMP	#0,ANS7	:CHECK THIRD WORD	
010526	001401				BEG	+.4		
010530	104010				HLT+8.		:THIRD # 0	
010532	022767	000000	170260		CMP	#0,ANS8	:CHECK FORTH WORD	
010540	001401				BEG	+.4		
010542	104010				HLT+8.		:FORTH # 0	

TEST 45 TEST OF MODD FPU INSTRUCTION
40200 0 0 0 * 40200 0 0 0 =
40200 0 0 0 & 0 0 0 0
USING AC2 FPS = 47604 FEC = N/A

010544	104400				SCOPE		
010546	170127	047600			LDFPS	#47604&57760	
010552	172667	000030			LDD	M45,2	:LOAD 40200 INTO AC2
010556	171667	000034			MODD	M45,2	:MODIFY 40200 BY 40200
010562	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
010564	022700	047604			CMP	#47604,FPS	:CHECK FLOATING POINT STATUS
010570	001401				BEG	.+4	:BRANCH IF OK
010572	104000				HLT		:FPS NOT EQUAL TO 47604
010574	174267	170202			STD	2,ANS1	:STORE FRACTIONAL PART
010600	174367	170206			STD	2:1,ANS5	:STORE INTERGER PART
010604	000410				BR	045	
010606	040200	000000	000000	M45:	40200,0,0,0		
010614	000000						
010616	040200	000000	000000	M45:	40200,0,0,0		
010624	000000						
010626	022767	000000	170146	045:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
010634	001401				BEG	.+4	
010636	104010				HLT+8.		:FIRST # 0
010640	022767	000000	170136		CMP	#0,ANS2	:CHECK SECOND WORD
010646	001401				BEG	.+4	
010650	104010				HLT+8.		:SECOND # 0
010652	022767	000000	170126		CMP	#0,ANS3	:CHECK THIRD WORD
010660	001401				BEG	.+4	
010662	104010				HLT+8.		:THIRD # 0
010664	022767	000000	170116		CMP	#0,ANS4	:CHECK FORTH
010672	001401				BEG	.+4	
010674	104010				HLT+8.		:FORTH # 0
010676	022767	040200	170106		CMP	#40200,ANS5	:CHECK FIRST WORD OF INTEGER
010704	001401				BEG	.+4	
010706	104010				HLT+8.		:FIRST # 40200
010710	022767	000000	170076		CMP	#0,ANS6	:CHECK SECOND WORD
010716	001401				BEG	.+4	
010720	104010				HLT+8.		:SECOND # 0
010722	022767	000000	170066		CMP	#0,ANS7	:CHECK THIRD WORD
010730	001401				BEG	.+4	
010732	104010				HLT+8.		:THIRD # 0
010734	022767	000000	170056		CMP	#0,ANS8	:CHECK FORTH WORD
010742	001401				BEG	.+4	
010744	104010				HLT+8.		:FORTH # 0


```

*****
TEST 46 TEST OF MODD FPU INSTRUCTION
40214 146314 146314 146315 * 40200 0 0 0 =
40200 0 0 0 & 37314 146314 146314 146320
JSING AC2 FPS = 47600 FEC = N/A
*****

```

010746	104400				SCOPE		
010750	170127	047600			LDFPS	#47600&57760	
010754	172667	000030			LDJ	M46,2	:LOAD 40214 INTO AC2
010760	171667	000034			MODD	M46,2	:MODIFY 40214 BY 40200
010764	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
010766	022700	047600			CMP	#47600.FPS	:CHECK FLOATING POINT STATUS
010772	001401				BEG	+.4	:BRANCH IF OK
010774	104000				HLT		:FPS NOT EQUAL TO 47600
010776	174267	170000			STD	2,ANS1	:STORE FRACTIONAL PART
011002	174367	170004			STD	2!1,ANS5	:STORE INTERGER PART
011006	000410				BR	046	
011010	040214	146314	146314	M46:	40214,146314,146314,146315		
011016	146315						
011020	040200	000000	000000	M46:	40200,0,0,0		
011026	000000						
011030	022767	037314	167744	M46:	CMP #37314,ANS1		:CHECK FIRST WORD OF FRACTION
011036	001401				BEG	+.4	
011040	104010				HLT+8.		:FIRST # 37314
011042	022767	146314	167734		CMP #146314,ANS2		:CHECK SECOND WORD
011050	001401				BEG	+.4	
011052	104010				HLT+8.		:SECOND # 146314
011054	022767	146314	167724		CMP #146314,ANS3		:CHECK THIRD WORD
011062	001401				BEG	+.4	
011064	104010				HLT+8.		:THIRD # 146314
011066	022767	146320	167714		CMP #146320,ANS4		:CHECK FORTH
011074	001401				BEG	+.4	
011076	104010				HLT+8.		:FORTH # 146320
011100	022767	040200	167704		CMP #40200,ANS5		:CHECK FIRST WORD OF INTEGER
011106	001401				BEG	+.4	
011110	104010				HLT+8.		:FIRST # 40200
011112	022767	000000	167674		CMP #0,ANS6		:CHECK SECOND WORD
011120	001401				BEG	+.4	
011122	104010				HLT+8.		:SECOND # 0
011124	022767	000000	167664		CMP #0,ANS7		:CHECK THIRD WORD
011132	001401				BEG	+.4	
011134	104010				HLT+8.		:THIRD # 0
011136	022767	000000	167654		CMP #0,ANS8		:CHECK FORTH WORD
011144	001401				BEG	+.4	
011146	104010				HLT+8.		:FORTH # 0

K04

MAINDEC-11-DCFPL-B
DCFPLB.F11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 49

:TEST 47 TEST OF MODD FPU INSTRUCTION
: 40414 146314 146314 146315 * 40200 0 0 0 =
: 40400 0 0 0 & 37514 146314 146314 146320
: USING AC2 FPS = 47600 FEC = N/A

011150	104400				SCOPE		
011152	170127	047600			LDFPS	#47600&57760	
011156	172667	000030			LDD	M47,2	:LOAD 40414 INTO AC2
011162	171667	000034			MODD	M47,2	:MODIFY 40414 BY 40200
011166	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
011170	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
011174	001401				BEG	.+4	:BRANCH IF OK
011176	104000				HLT		:FPS NOT EQUAL TO 47600
011200	174267	167576			STD	2,ANS1	:STORE FRACTIONAL PART
011204	174367	167602			STD	2!,ANS5	:STORE INTERGER PART
011210	000410				BR	047	
011212	040414	146314	146314	N47:		40414,146314,146314,146315	
011220	146315						
011222	040200	000000	000000	M47:		40200,0,0,0	
011230	000000						
011232	022767	037514	167542	047:	CMP	#37514,ANS1	:CHECK FIRST WORD OF FRACTION
011240	001401				BEG	.+4	
011242	104010				HLT+8.		:FIRST # 37514
011244	022767	146314	167532		CMP	#146314,ANS2	:CHECK SECOND WORD
011252	001401				BEG	.+4	
011254	104010				HLT+8.		:SECOND # 146314
011256	022767	146314	167522		CMP	#146314,ANS3	:CHECK THIRD WORD
011264	001401				BEG	.+4	
011266	104010				HLT+8.		:THIRD # 146314
011270	022767	146320	167512		CMP	#146320,ANS4	:CHECK FORTH
011276	001401				BEG	.+4	
011300	104010				HLT+8.		:FORTH # 146320
011302	022767	040400	167502		CMP	#40400,ANS5	:CHECK FIRST WORD OF INTEGER
011310	001401				BEG	.+4	
011312	104010				HLT+8.		:FIRST # 40400
011314	022767	000000	167472		CMP	#0,ANS6	:CHECK SECOND WORD
011322	001401				BEG	.+4	
011324	104010				HLT+8.		:SECOND # 0
011326	022767	000000	167462		CMP	#0,ANS7	:CHECK THIRD WORD
011334	001401				BEG	.+4	
011336	104010				HLT+8.		:THIRD # 0
011340	022767	000000	167452		CMP	#0,ANS8	:CHECK FORTH WORD
011346	001401				BEG	.+4	
011350	104010				HLT+8.		:FORTH # 0

```

*****
:TEST 50 TEST OF MODD FPU INSTRUCTION
: 40523 31463 31463 31463 * 40200 0 0 0 =
: 40500 0 0 0 & 3/631 114631 114631 114630
: USING AC2 FPS = 47600 FEC = N/A
*****

```

```

011352 104400 SCOPE
011354 170127 047600 LDFPS #47600&57760
011360 172667 000030 LDD NS0,2 ;LOAD 40523 INTO AC2
011364 171667 000034 MODD MS0,2 ;MODIFY 40523 BY 40200
011370 170200 STFPS FPS ;STORE FLOATING POINT STATUS
011372 022700 047600 CMP #47600,FPS ;CHECK FLOATING POINT STATUS
011376 001401 BEQ .+4 ;BRANCH IF OK
011400 104000 HLT ;FPS NOT EQUAL TO 47600

011402 174267 167374 STD 2,ANS1 ;STORE FRACTIONAL PART
011406 174367 167400 STD 2!1,ANS5 ;STORE INTERGER PART
011412 000410 BR 050

011414 040523 031463 031463 NS0: 40523,31463,31463,31463
011422 031463
011424 040200 000000 000000 MS0: 40200,0,0,0
011432 000000
011434 022767 037631 167340 OS0: CMP #37631,ANS1 ;CHECK FIRST WORD OF FRACTION
011442 001401 BEQ .+4
011444 104010 HLT+8. ;FIRST # 37631

011446 022767 114631 167330 CMP #114631,ANS2 ;CHECK SECOND WORD
011454 001401 BEQ .+4
011456 104010 HLT+8. ;SECOND # 114631

011460 022767 114631 167320 CMP #114631,ANS3 ;CHECK THIRD WORD
011466 001401 BEQ .+4
011470 104010 HLT+8. ;THIRD # 114631

011472 022767 114630 167310 CMP #114630,ANS4 ;CHECK FORTH
011500 001401 BEQ .+4
011502 104010 HLT+8. ;FORTH # 114630

011504 022767 040500 167300 CMP #40500,ANS5 ;CHECK FIRST WORD OF INTEGER
011512 001401 BEQ .+4
011514 104010 HLT+8. ;FIRST # 40500

011516 022767 000000 167270 CMP #0,ANS6 ;CHECK SECOND WORD
011524 001401 BEQ .+4
011526 104010 HLT+8. ;SECOND # 0

011530 022767 000000 167260 CMP #0,ANS7 ;CHECK THIRD WORD
011536 001401 BEQ .+4
011540 104010 HLT+8. ;THIRD # 0

011542 022767 000000 167250 CMP #0,ANS8 ;CHECK FORTH WORD
011550 001401 BEQ .+4
011552 104010 HLT+8. ;FORTH # 0

```

```

*****
:TEST S1 TEST OF MODD FPU INSTRUCTION
: 40660 0 0 0 * 40200 0 0 0 =
: 40640 0 0 0 & 40000 0 0 0
: USING AC2 FPS = 47600 FEC = N/A
*****

```

```

011554 104400
011556 170127 047600
011562 172667 000030
011566 171667 000034
011572 170200
011574 022700 047600
011600 001401
011602 104000
SCOPE
LDFPS #47600&57760
LDD NS1,2 ;LOAD 40660 INTO AC2
MODD M51,2 ;MODIFY 40660 BY 40200
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #47600,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 47600

011604 174267 167172
011610 174367 167176
011614 000410
STD 2,ANS1 ;STORE FRACTIONAL PART
STD 2,1,ANS5 ;STORE INTERGER PART
BR 051

011616 040660 000000 000000 NS1: 40660,0,0,0
011624 000000
011626 040200 000000 000000 M51: 40200,0,0,0
011634 000000
011636 022767 040000 167136 051: CMP #40000,ANS1 ;CHECK FIRST WORD OF FRACTION
011644 001401 BEQ .+4
011646 104010 HLT+8. ;FIRST # 40000

011650 022767 000000 167126 CMP #0,ANS2 ;CHECK SECOND WORD
011656 001401 BEQ .+4
011660 104010 HLT+8. ;SECOND # 0

011662 022767 000000 167116 CMP #0,ANS3 ;CHECK THIRD WORD
011670 001401 BEQ .+4
011672 104010 HLT+8. ;THIRD # 0

011674 022767 000000 167106 CMP #0,ANS4 ;CHECK FORTH
011702 001401 BEQ .+4
011704 104010 HLT+8. ;FORTH # 0

011706 022767 040640 167076 CMP #40640,ANS5 ;CHECK FIRST WORD OF INTEGER
011714 001401 BEQ .+4
011716 104010 HLT+8. ;FIRST # 40640

011720 022767 000000 167066 CMP #0,ANS6 ;CHECK SECOND WORD
011726 001401 BEQ .+4
011730 104010 HLT+8. ;SECOND # 0

011732 022767 000000 167056 CMP #0,ANS7 ;CHECK THIRD WORD
011740 001401 BEQ .+4
011742 104010 HLT+8. ;THIRD # 0

011744 022767 000000 167046 CMP #0,ANS8 ;CHECK FORTH WORD
011752 001401 BEQ .+4
011754 104010 HLT+8. ;FORTH # 0

```

```

*****
TEST 52 TEST OF MODD FPU INSTRUCTION
140200 0 0 0 * 40200 0 0 0 =
140200 0 0 0 & 0 0 0 0
USING AC2 FPS = 47604 FEC = N/A
*****

```

```

011756 104400
011760 170127 047600
011764 172667 000030
011770 171667 000034
011774 170200
011776 022700 047604
012002 001401
012004 104000

012006 174267 166770
012012 174367 166774
012016 000410

012020 140200 000000 000000 NS2: 140200,0,0,0
012026 000000
012030 040200 000000 000000 MS2: 40200,0,0,0
012036 000000
012040 022767 000000 166734 052: CMP #0,ANS1 ;CHECK FIRST WORD OF FRACTION
012046 001401 BEQ .+4 ;FIRST # 0
012050 104010 HLT+8.

012052 022767 000000 166724 CMP #0,ANS2 ;CHECK SECOND WORD
012060 001401 BEQ .+4 ;SECOND # 0
012062 104010 HLT+8.

012064 022767 000000 166714 CMP #0,ANS3 ;CHECK THIRD WORD
012072 001401 BEQ .+4 ;THIRD # 0
012074 104010 HLT+8.

012076 022767 000000 166704 CMP #0,ANS4 ;CHECK FORTH
012104 001401 BEQ .+4 ;FORTH # 0
012106 104010 HLT+8.

012110 022767 140200 166674 CMP #140200,ANS5 ;CHECK FIRST WORD OF INTEGER
012116 001401 BEQ .+4 ;FIRST # 140200
012120 104010 HLT+8.

012122 022767 000000 166664 CMP #0,ANS6 ;CHECK SECOND WORD
012130 001401 BEQ .+4 ;SECOND # 0
012132 104010 HLT+8.

012134 022767 000000 166654 CMP #0,ANS7 ;CHECK THIRD WORD
012142 001401 BEQ .+4 ;THIRD # 0
012144 104010 HLT+8.

012146 022767 000000 166644 CMP #0,ANS8 ;CHECK FORTH WORD
012154 001401 BEQ .+4 ;FORTH # 0
012156 104010 HLT+8.

SCOPE
LDFPS #47604&57760
LDD NS2,2 ;LOAD 140200 INTO AC2
MODD MS2,2 ;MODIFY 140200 BY 40200
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #47604,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 47604

STD 2,ANS1 ;STORE FRACTIONAL PART
STD 2!1,ANS5 ;STORE INTERGER PART
BR 052

```

```

*****
TEST 53 TEST OF MODD FPU INSTRUCTION
140214 146314 146314 146315 * 40200 0 0 0 =
140200 0 0 0 8 137314 146314 146314 146320
USING AC2 FPS = 47610 FEC = N/A
*****

```

012160	104400				SCOPE		
012162	170127	047600			LDFPS	#47610&57760	
012164	172667	000030			LDC	M53,2	:LOAD 140214 INTO AC2
012166	171667	000034			MODD	M53,2	:MODIFY 140214 BY 40200
012170	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
012200	022700	047610			CMP	#47610,FPS	:CHECK FLOATING POINT STATUS
012204	001401				BEG	.+4	:BRANCH IF OK
012206	104000				HLT		:FPS NOT EQUAL TO 47610
012210	174267	166566			STC	2,ANS1	:STORE FRACTIONAL PART
012214	174367	166572			STC	2,1,ANS5	:STORE INTEGER PART
012220	000410				BR	053	
012222	140214	146314	146314	M53:	140214,146314,146314,146315		
012230	146315						
012232	040200	000000	000000	M53:	40200,0,0,0		
012240	000000						
012242	022767	137314	166532	053:	CMP #137314,ANS1	:CHECK FIRST WORD OF FRACTION	
012250	001401				BEG .+4		
012252	104010				HLT+B.	:FIRST # 137314	
012254	022767	146314	166522		CMP #146314,ANS2	:CHECK SECOND WORD	
012262	001401				BEG .+4		
012264	104010				HLT+B.	:SECOND # 146314	
012266	022767	146314	166512		CMP #146314,ANS3	:CHECK THIRD WORD	
012274	001401				BEG .+4		
012276	104010				HLT+B.	:THIRD # 146314	
012300	022767	146320	166502		CMP #146320,ANS4	:CHECK FORTH	
012306	001401				BEG .+4		
012310	104010				HLT+B.	:FORTH # 146320	
012312	022767	140200	166472		CMP #140200,ANS5	:CHECK FIRST WORD OF INTEGER	
012320	001401				BEG .+4		
012322	104010				HLT+B.	:FIRST # 140200	
012324	022767	000000	166462		CMP #0,ANS6	:CHECK SECOND WORD	
012332	001401				BEG .+4		
012334	104010				HLT+B.	:SECOND # 0	
012336	022767	000000	166452		CMP #0,ANS7	:CHECK THIRD WORD	
012344	001401				BEG .+4		
012346	104010				HLT+B.	:THIRD # 0	
012350	022767	000000	166442		CMP #0,ANS8	:CHECK FORTH WORD	
012358	001401				BEG .+4		
012360	104010				HLT+B.	:FORTH # 0	

C05

TEST 54 TEST OF MODD FPU INSTRUCTION
140414 146314 146314 146315 * 40200 0 0 0 =
140400 0 0 0 & 137514 146314 146314 146320
USING AC2 FPS = 47610 FEC = N/A

012362 104400
012364 170127 047600
012370 172667 000030
012374 171667 000034
012400 170200
012402 022700 047610
012406 001401
012410 104010

SCOPE
LOFPS #47610&57760
LDD NS4,2 :LOAD 140414 INTO AC2
MODD NS4,2 :MODIFY 140414 BY 40200
STFPS FPS :STORE FLOATING POINT STATUS
CMP #47610.FPS :CHECK FLOATING POINT STATUS
BEG .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 47610

012412 174267 166364
012416 174367 166370
012422 000410

STD 2,ANS1 :STORE FRACTIONAL PART
STC 21,ANS5 :STORE INTEGER PART
BR 054

012424 140414 146314 146314 NS4: 140414,146314,146314,146315

012432 146315
012434 040200 000000 000000 NS4: 40200,0,0,0
012442 000000

012444 022767 137514 166330 054: CMP #137514,ANS1 :CHECK FIRST WORD OF FRACTION
BEG .+4
HLT+B. :FIRST # 137514

012456 022767 146314 166320 CMP #146314,ANS2 :CHECK SECOND WORD
BEG .+4
HLT+B. :SECOND # 146314

012470 022767 146314 166310 CMP #146314,ANS3 :CHECK THIRD WORD
BEG .+4
HLT+B. :THIRD # 146314

012502 022767 146320 166300 CMP #146320,ANS4 :CHECK FORTH
BEG .+4
HLT+B. :FORTH # 146320

012514 022767 140400 166270 CMP #140400,ANS5 :CHECK FIRST WORD OF INTEGER
BEG .+4
HLT+B. :FIRST # 140400

012526 022767 000000 166260 CMP #0,ANS6 :CHECK SECOND WORD
BEG .+4
HLT+B. :SECOND # 0

012540 022767 000000 166250 CMP #0,ANS7 :CHECK THIRD WORD
BEG .+4
HLT+B. :THIRD # 0

012552 022767 000000 166240 CMP #0,ANS8 :CHECK FORTH WORD
BEG .+4
HLT+B. :FORTH # 0

:TEST 55 TEST OF MODD FPU INSTRUCTION
: 140523 31463 31463 31463 * 40200 0 0 0 =
: 140500 0 0 0 & 137631 114631 114631 114630
: USING AC2 FPS = 47610 FEC = N/A

012564	104400				SCOPE		
012566	170127	047600			LOFPS	#47610&57760	
012572	172667	000030			LDD	M55.2	:LOAD 140523 INTO AC2
012576	171667	000034			MODD	M55.2	:MODIFY 140523 BY 40200
012602	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
012604	022700	047610			CMP	#47610.FPS	:CHECK FLOATING POINT STATUS
012610	001401				BEG	.+4	:BRANCH IF OK
012612	104000				HLT		:FPS NOT EQUAL TO 47610
012614	174267	166162			STC	2,ANS1	:STORE FRACTIONAL PART
012620	174367	166166			STC	21,ANS5	:STORE INTEGER PART
012624	000410				BP	055	
012626	140523	031463	031463	M55:	140523,31463,31463,31463		
012634	031463						
012636	040200	000000	000000	M55:	40200,0,0,0		
012644	000000						
012646	022767	137631	166126	C55:	CMP #137631,ANS1		:CHECK FIRST WORD OF FRACTION
012654	001401				BEG .+4		
012656	104010				HLT+B.		:FIRST # 137631
012660	022767	114631	166116		CMP #114631,ANS2		:CHECK SECOND WORD
012666	001401				BEG .+4		
012670	104010				HLT+B.		:SECOND # 114631
012672	022767	114631	166106		CMP #114631,ANS3		:CHECK THIRD WORD
012700	001401				BEG .+4		
012702	104010				HLT+B.		:THIRD # 114631
012704	022767	114630	166076		CMP #114630,ANS4		:CHECK FORTH
012712	001401				BEG .+4		
012714	104010				HLT+B.		:FORTH # 114630
012716	022767	140500	166066		CMP #140500,ANS5		:CHECK FIRST WORD OF INTEGER
012724	001401				BEG .+4		
012726	104010				HLT+B.		:FIRST # 140500
012730	022767	000000	166056		CMP #0,ANS6		:CHECK SECOND WORD
012736	001401				BEG .+4		
012740	104010				HLT+B.		:SECOND # 0
012742	022767	000000	166046		CMP #0,ANS7		:CHECK THIRD WORD
012750	001401				BEG .+4		
012752	104010				HLT+B.		:THIRD # 0
012754	022767	000000	166036		CMP #0,ANS8		:CHECK FORTH WORD
012762	001401				BEG .+4		
012764	104010				HLT+B.		:FORTH # 0


```

*****
TEST 56 TEST OF MODD FPU INSTRUCTION
140660 0 0 0 * 40200 0 0 0 =
140640 0 0 0 & 140000 0 0 0
USING AC2 FPS = 47610 FEC = N/A
*****

```

013000	171667	000034							
013004	170200								
013006	022700	047610							
013012	001401								
013014	104000								
013016	174267	165760							
013022	174367	165764							
013026	000410								
013030	140660	000000	000000	NS6:	140660,0,0,0				
013036	000000								
013040	040200	000000	000000	MS6:	40200,0,0,0				
013046	000000								
013050	022767	140000	165724	OS6:	CMP #140000,ANS1			:CHECK FIRST WORD OF FRACTION	
013056	001401				BEQ .+4				
013060	104010				HLT+8.			:FIRST # 140000	
013062	022767	000000	165714		CMP #0,ANS2			:CHECK SECOND WORD	
013070	001401				BEQ .+4				
013072	104010				HLT+8.			:SECOND # 0	
013074	022767	000000	165704		CMP #0,ANS3			:CHECK THIRD WORD	
013102	001401				BEQ .+4				
013104	104010				HLT+8.			:THIRD # 0	
013106	022767	000000	165674		CMP #0,ANS4			:CHECK FORTH	
013114	001401				BEQ .+4				
013116	104010				HLT+8.			:FORTH # 0	
013120	022767	140640	165664		CMP #140640,ANS5			:CHECK FIRST WORD OF INTEGER	
013126	001401				BEQ .+4				
013130	104010				HLT+8.			:FIRST # 140640	
013132	022767	000000	165654		CMP #0,ANS6			:CHECK SECOND WORD	
013140	001401				BEQ .+4				
013142	104010				HLT+8.			:SECOND # 0	
013144	022767	000000	165644		CMP #0,ANS7			:CHECK THIRD WORD	
013152	001401				BEQ .+4				
013154	104010				HLT+8.			:THIRD # 0	
013156	022767	000000	165634		CMP #0,ANS8			:CHECK FORTH WORD	
013164	001401				BEQ .+4				
013166	104010				HLT+8.			:FORTH # 0	

F05

MAINDEC-11-CCFP-1-2
CCFP_LB.F11 TEST

```

*****
:TEST 57 TEST OF MODD FPU INSTRUCTION
: 52525 52525 52525 52525 * 40200 0 0 0 =
: 52525 52525 52525 40000 & 37652 124000 0 0
: USING AC2 FPS = 47600 FEC = N/A
*****

```

013170	104400				SCOPE		
013172	170127	047600			LDFPS	#47600&57760	
013176	172667	000030			LDD	M57,2	:LOAD 52525 INTO AC2
013202	171667	000034			MODD	M57,2	:MODIFY 52525 BY 40200
013206	170200				S FPS	FPS	:STORE FLOATING POINT STATUS
013210	022700	047600			CMP	#47600,FPS	:CHECK FLOATING POINT STATUS
013214	001401				BEQ	+.4	:BRANCH IF OK
013216	104000				HLT		:FPS NOT EQUAL TO 47600
013220	174267	165556			STD	2,ANS1	:STORE FRACTIONAL PART
013224	174367	165552			STD	21,ANS5	:STORE INTERGER PART
013230	000410				BR	057	
013232	052525	052525	052525	N57:	52525,52525,52525,52525		
013240	052525						
013242	040200	000000	000000	M57:	40200,0,0,0		
013250	000000						
013252	022767	037652	165522	O57:	CMP #37652,ANS1		:CHECK FIRST WORD OF FRACTION
013260	001401				BEQ	+.4	
013262	104010				HLT+8.		:FIRST # 37652
013264	022767	124000	165512		CMP #124000,ANS2		:CHECK SECOND WORD
013272	001401				BEQ	+.4	
013274	104010				HLT+8.		:SECOND # 124000
013276	022767	000000	165502		CMP #0,ANS3		:CHECK THIRD WORD
013304	001401				BEQ	+.4	
013306	104010				HLT+8.		:THIRD # 0
013310	022767	000000	165472		CMP #0,ANS4		:CHECK FORTH
013316	001401				BEQ	+.4	
013320	104010				HLT+8.		:FORTH # 0
013322	022767	052525	165462		CMP #52525,ANS5		:CHECK FIRST WORD OF INTEGER
013330	001401				BEQ	+.4	
013332	104010				HLT+8.		:FIRST # 52525
013334	022767	052525	165452		CMP #52525,ANS6		:CHECK SECOND WORD
013342	001401				BEQ	+.4	
013344	104010				HLT+8.		:SECOND # 52525
013346	022767	052525	165442		CMP #52525,ANS7		:CHECK THIRD WORD
013354	001401				BEQ	+.4	
013356	104010				HLT+8.		:THIRD # 52525
013360	022767	040000	165432		CMP #40000,ANS8		:CHECK FORTH WORD
013366	001401				BEQ	+.4	
013370	104010				HLT+8.		:FORTH # 40000

G05

MAINDEC-11-DCFPL-B
DCFPLB.F11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 58

```
*****
:TEST 60 TEST OF MODD FPU INSTRUCTION
: 125252 125252 125252 125252 * 40200 0 0 0 =
: 0 0 0 0 & 125252 125252 125252 125252
: USING AC2 FPS = 47610 FEC = N/A
*****
013372 104400 SCOPE
013374 170127 047600 LDFPS #47610&57760
013400 172667 000030 LDD M60,2 ;LOAD 125252 INTO AC2
013404 171667 000034 MODD M60,2 ;MODIFY 125252 BY 40200
013410 170200 STFPS FPS ;STORE FLOATING POINT STATUS
013412 022700 047610 CMP #47610,FPS ;CHECK FLOATING POINT STATUS
013416 001401 BEQ .+4 ;BRANCH IF OK
013420 104000 HLT ;FPS NOT EQUAL TO 47610

013422 174267 165354 STD 2,ANS1 ;STORE FRACTIONAL PART
013426 174367 165360 STD 2!1,ANS5 ;STORE INTERGER PART
013432 000410 BR 060

013434 125252 125252 125252 125252 M60: 125252,125252,125252,125252
013442 125252
013444 040200 000000 000000 M60: 40200,0,0,0
013452 000000
013454 022767 125252 165320 060: CMP #125252,ANS1 ;CHECK FIRST WORD OF FRACTION
013462 001401 BEQ .+4
013464 104010 HLT+8. ;FIRST # 125252

013466 022767 125252 165310 CMP #125252,ANS2 ;CHECK SECOND WORD
013474 001401 BEQ .+4
013476 104010 HLT+8. ;SECOND # 125252

013500 022767 125252 165300 CMP #125252,ANS3 ;CHECK THIRD WORD
013506 001401 BEQ .+4
013510 104010 HLT+8. ;THIRD # 125252

013512 022767 125252 165270 CMP #125252,ANS4 ;CHECK FORTH
013520 001401 BEQ .+4
013522 104010 HLT+8. ;FORTH # 125252

013524 022767 000000 165260 CMP #0,ANS5 ;CHECK FIRST WORD OF INTEGER
013532 001401 BEQ .+4
013534 104010 HLT+8. ;FIRST # 0

013536 022767 000000 165250 CMP #0,ANS6 ;CHECK SECOND WORD
013544 001401 BEQ .+4
013546 104010 HLT+8. ;SECOND # 0

013550 022767 000000 165240 CMP #0,ANS7 ;CHECK THIRD WORD
013556 001401 BEQ .+4
013560 104010 HLT+8. ;THIRD # 0

013562 022767 000000 165230 CMP #0,ANS8 ;CHECK FORTH WORD
013570 001401 BEQ .+4
013572 104010 HLT+8. ;FORTH # 0
```

```

*****
:TEST 61 TEST OF MODD FPU INSTRUCTION
: 52525 52525 52525 52525 * 44000 0 0 0 =
: 56325 52525 52525 52525 & 0 0 0 0
: USING ACD FPS = 47604 FEC = N/A
*****

```

013574	104400				SCOPE		
013576	170127	047600			LDFPS	#47604&57760	
013602	172467	000030			LDD	M61.0	:LOAD 52525 INTO ACD
013606	171467	000034			MODD	M61.0	:MODIFY 52525 BY 44000
013612	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
013614	022700	047604			CMP	#47604,FPS	:CHECK FLOATING POINT STATUS
013620	001401				BEQ	+.4	:BRANCH IF OK
013622	104000				HLT		:FPS NOT EQUAL TO 47604
013624	174067	165152			STD	0,ANS1	:STORE FRACTIONAL PART
013630	174167	165156			S*D	0!1,ANS5	:STORE INTERGER PART
013634	000410				BR	061	
013636	052525	052525	052525	N61:	52525,52525,52525,52525		
013644	052525						
013646	044000	000000	000000	M61:	44000,0,0,0		
013654	000000						
013656	022767	000000	165116	061:	CMP	#0,ANS1	:CHECK FIRST WORD OF FRACTION
013664	001401				BEQ	+.4	
013666	104010				HLT+8.		:FIRST # 0
013670	022767	000000	165106		CMP	#0,ANS2	:CHECK SECOND WORD
013676	001401				BEQ	+.4	
013700	104010				HLT+8.		:SECOND # J
013702	022767	000000	165076		CMP	#0,ANS3	:CHECK THIRD WORD
013710	001401				BEQ	+.4	
013712	104010				HLT+8.		:THIRD # 0
013714	022767	000000	165066		CMP	#0,ANS4	:CHECK FORTH
013722	001401				BEQ	+.4	
013724	104010				HLT+8.		:FORTH # 0
013726	022767	056325	165056		CMP	#56325,ANS5	:CHECK FIRST WORD OF INTEGER
013734	001401				BEQ	+.4	
013736	104010				HLT+8.		:FIRST # 56325
013740	022767	052525	165046		CMP	#52525,ANS6	:CHECK SECOND WORD
013746	001401				BEQ	+.4	
013750	104010				HLT+8.		:SECOND # 52525
013752	022767	052525	165036		CMP	#52525,ANS7	:CHECK THIRD WORD
013760	001401				BEQ	+.4	
013762	104010				HLT+8.		:THIRD # 52525
013764	022767	052525	165026		CMP	#52525,ANS8	:CHECK FORTH WORD
013772	001401				BEQ	+.4	
013774	104010				HLT+8.		:FORTH # 52525

```

*****
:TEST 62 TEST OF MODD FPU INSTRUCTION
: 37600 0 0 0 * 40200 0 0 0 =
: 0 0 0 0 & 37600 0 0 0
: USING AC2 FPS = 47600 FEC = N/A
*****

```

```

013776 104400
014000 170127 047600
014004 172667 000030
014010 171667 000034
014014 170200
014016 022700 047600
014022 001401
014024 104000
SCOPE
LDFPS #47600&57760
LDD M62.2 :LOAD 37600 INTO AC2
MODD M62.2 :MODIFY 37600 BY 40200
STFPS FPS :STORE FLOATING POINT STATUS
CMP #47600.FPS :CHECK FLOATING POINT STATUS
BEQ .+4 :BRANCH IF OK
HLT :FPS NOT EQUAL TO 47600

014026 174267 164750
014032 174367 164754
014036 000410
STD 2,ANS1 :STORE FRACTIONAL PART
STC 2+1,ANS5 :STORE INTERGER PART
BR 062

014040 037600 000000 000000 M62: 37600.0,0,0
014046 000000
014050 040200 000000 000000 M62: 40200.0,0,0
014056 000000
014060 022767 037600 164714 062: CMP #37600,ANS1 :CHECK FIRST WORD OF FRACTION
014066 001401 BEQ .+4 :FIRST # 37600
014070 104010 HLT+8.

014072 022767 000000 164704 CMP #0,ANS2 :CHECK SECOND WORD
014100 001401 BEQ .+4 :SECOND # 0
014102 104010 HLT+8.

014104 022767 000000 164674 CMP #0,ANS3 :CHECK THIRD WORD
014112 001401 BEQ .+4 :THIRD # 0
014114 104010 HLT+8.

014116 022767 000000 164664 CMP #0,ANS4 :CHECK FORTH
014124 001401 BEQ .+4 :FORTH # 0
014126 104010 HLT+8.

014130 022767 000000 164654 CMP #0,ANS5 :CHECK FIRST WORD OF INTEGER
014136 001401 BEQ .+4 :FIRST # 0
014140 104010 HLT+8.

014142 022767 000000 164644 CMP #0,ANS6 :CHECK SECOND WORD
014150 001401 BEQ .+4 :SECOND # 0
014152 104010 HLT+8.

014154 022767 000000 164634 CMP #0,ANS7 :CHECK THIRD WORD
014162 001401 BEQ .+4 :THIRD # 0
014164 104010 HLT+8.

014166 022767 000000 164624 CMP #0,ANS8 :CHECK FORTH WORD
014174 001401 BEQ .+4 :FORTH # 0
014176 104010 HLT+8.

```

```

*****
:TEST 63 TEST OF MODD FPU INSTRUCTION
: 40200 0 0 0 * 10000 0 0 0 =
: 0 0 0 0 10000 0 0 0
: USING AC2 FPS = 47600 FEC = N/A
*****

```

014200	104400				SCOPE		
014202	170127	047600			LDFPS	#47600&57760	
014206	172667	000030			LDD	M63.2	:LOAD 40200 INTO AC2
014212	171667	000034			MODD	M63.2	:MODIFY 40200 BY 10000
014216	170200				STFPS	FPS	:STORE FLOATING POINT STATUS
014220	022700	047600			CMP	#47600.FPS	:CHECK FLOATING POINT STATUS
014224	001401				BEG	+.4	:BRANCH IF OK
014226	104000				HLT		:FPS NOT EQUAL TO 47600
014230	174267	164546			STD	2,ANS1	:STORE FRACTIONAL PART
014234	174367	164552			STD	211,ANS5	:STORE INTERGER PART
014240	000410				BR	063	
014242	040200	000000	000000	M63:	40200.0,0,0		
014250	000000						
014252	010000	000000	000000	M63:	10000.0,0,0		
014260	000000						
014262	022767	010000	164512	063:	CMP	#10000,ANS1	:CHECK FIRST WORD OF FRACTION
014270	001401				BEG	+.4	
014272	104010				HLT+B.		:FIRST # 10000
014274	022767	000000	164502		CMP	#0,ANS2	:CHECK SECOND WORD
014302	001401				BEG	+.4	
014304	104010				HLT+B.		:SECOND # 0
014306	022767	000000	164472		CMP	#0,ANS3	:CHECK THIRD WORD
014314	001401				BEG	+.4	
014316	104010				HLT+B.		:THIRD # 0
014320	022767	000000	164462		CMP	#0,ANS4	:CHECK FORTH
014326	001401				BEG	+.4	
014330	104010				HLT+B.		:FORTH # 0
014332	022767	000000	164452		CMP	#0,ANS5	:CHECK FIRST WORD OF INTEGER
014340	001401				BEG	+.4	
014342	104010				HLT+B.		:FIRST # 0
014344	022767	000000	164442		CMP	#0,ANS6	:CHECK SECOND WORD
014352	001401				BEG	+.4	
014354	104010				HLT+B.		:SECOND # 0
014356	022767	000000	164432		CMP	#0,ANS7	:CHECK THIRD WORD
014364	001401				BEG	+.4	
014366	104010				HLT+B.		:THIRD # 0
014370	022767	000000	164422		CMP	#0,ANS8	:CHECK FORTH WORD
014376	001401				BEG	+.4	
014400	104010				HLT+B.		:FORTH # 0

K05

MAINDEC-11-DCFPL-B
DCFPLB.P11 TEST

TEST OF MODF AND MODD MACY11 27(732) 03-SEP-76 14:57 PAGE 62

```

*****
:TEST 64 TEST OF MODD FPU INSTRUCTION
: 40600 0 0 0 * 37400 0 0 0 =
: 0 0 0 0 & 40000 0 0 0
: USING ACO FPS = 47600 FEC = N/A
*****

```

```

014402 104400
014404 170127 047600
014410 172467 000030
014414 171467 000034
014420 170200
014422 022700 047600
014426 001401
014430 104000

```

```

SCOPE
LDFPS #47600&57760
LDD M64,0 ;LOAD 40600 INTO ACO
MODD M64,0 ;MODIFY 40600 BY 37400
STFPS FPS ;STORE FLOATING POINT STATUS
CMP #47600,FPS ;CHECK FLOATING POINT STATUS
BEQ .+4 ;BRANCH IF OK
HLT ;FPS NOT EQUAL TO 47600

```

```

014432 174067 164344
014436 174167 164350
014442 000410

```

```

STD 0,ANS1 ;STORE FRACTIONAL PART
STD 0!1,ANS5 ;STORE INTERGER PART
BR 064

```

```

014444 040600 000000 000000 M64: 40600.0,0,0
014452 000000
014454 037400 000000 000000 M64: 37400.0,0,0
014462 000000
014464 022767 040000 164310 064:
014472 001401
014474 104010

```

```

CMP #40000,ANS1 ;CHECK FIRST WORD OF FRACTION
BEQ .+4
HLT+8. ;FIRST # 40000

```

```

014476 022767 000000 164300
014504 001401
014506 104010

```

```

CMP #0,ANS2 ;CHECK SECOND WORD
BEQ .+4
HLT+8. ;SECOND # 0

```

```

014510 022767 000000 164270
014516 001401
014520 104010

```

```

CMP #0,ANS3 ;CHECK THIRD WORD
BEQ .+4
HLT+8. ;THIRD # 0

```

```

014522 022767 000000 164260
014530 001401
014532 104010

```

```

CMP #0,ANS4 ;CHECK FORTH
BEQ .+4
HLT+8. ;FORTH # 0

```

```

014534 022767 000000 164250
014542 001401
014544 104010

```

```

CMP #0,ANS5 ;CHECK FIRST WORD OF INTEGER
BEQ .+4
HLT+8. ;FIRST # 0

```

```

014546 022767 000000 164240
014554 001401
014556 104010

```

```

CMP #0,ANS6 ;CHECK SECOND WORD
BEQ .+4
HLT+8. ;SECOND # 0

```

```

014560 022767 000000 164230
014566 001401
014570 104010

```

```

CMP #0,ANS7 ;CHECK THIRD WORD
BEQ .+4
HLT+8. ;THIRD # 0

```

```

014572 022767 000000 164220
014600 001401
014602 104010

```

```

CMP #0,ANS8 ;CHECK FORTH WORD
BEQ .+4
HLT+8. ;FORTH # 0

```

014604	104400			DONE:	SCOPE		
014606	032737	002000	177570		BIT	#SW10, @#SWR	:RING THE BELL?
014614	001005				BNE	1\$:NO!
014616	012767	000007	001242		MOV	#BELL, .TYPE	:TYPE A BELL
014624	000004	016066			TYPE	-.TYPE	
014630	005046			1\$:	CLR	-(6)	:CLEAR TRACE TRAP
014632	032737	010000	177570		BIT	#SW12, @#SWR	:RUN WITH TRT?
014640	001010				BNE	2\$	
014642	005167	001222			COM	TRPB	
014646	100005				BPL	2\$	
014650	052716	000020			BIS	#20, (6)	:SET TRACE TRAP
014654	012746	001062			MOV	#BEGIN, -(6)	:JUMP TO START OF TEST
014660	000412				BR	YESRT	
014662	012746	001062		2\$:	MOV	#BEGIN, -(6)	:JUMP TO START OF TEST
014666	013700	000042			MOV	@#42, R0	:GET MONITOR ADDRESS
014672	001404				BEQ	3\$:IF NONE
014674	004710				JSR	7, (0)	:GO TO MONITOR
014676	000240				NOP		
014700	000240				NOP		
014702	000240				NOP		
014704	000002			3\$:	RTI		
014706	000002			YESRT:	RTI		:RETURN TO PROGRAM FROM TRAP
014710	032737	000400	177570	.EMT:	BIT	#SW08, @#SWR	:KILL LDUB OR LOOP ON SPEC. TEST
014716	001404				BEQ	1\$	
014720	123767	177570	164052		CMPB	@#SWR, ICNT	:ON RIGHT TEST? *SW7-0*
014726	001437				BEQ	OVER	
014730	113703	177570		1\$:	MOVB	@#SWR, R3	:GET UB BITS
014734	170003				LDUB		
014736	032737	040000	177570		BIT	#SW14, @#SWR	:LOOP ON TEST
014744	001026				BNE	KIT	
014746	032737	004000	177570		BIT	#SW11, @#SWR	:KILL ITERATIONS
014754	001012				BNE	SAVLAD	
014756	105767	164017			TSTB	ICNT+1	
014762	001404				BEQ	2\$:BRANCH IF FIRST
014764	126767	001106	164007		CMPB	TIMES, ICNT+1	:DONE?
014772	001013				BNE	KIT	:BRANCH IF NOT
014774	112767	000001	163777	2\$:	MOVB	#1, ICNT+1	:FIRST ITERATION
015002	105267	163772		SAVLAD:	INCB	ICNT	:COUNT TEST NUMBERS
015006	011667	001060			MOV	(6), LAD	:SAVE LOOP ADDRESS
015012	016737	163762	177570		MOV	ICNT, @#DISPLAY	:DISPLAY TEST NO. AND ITERATION COUNT
015020	000002				RTI		:RETURN
015022	105267	163753		KIT:	INCB	ICNT+1	
015026	016737	163746	177570	OVER:	MOV	ICNT, @#DISPLAY	:SET UP DISPLAY
015034	005767	001032			LAD		:FIRST ONE?
015040	001760				BEQ	SAVLAD	
015042	016716	001024			MOV	LAD, (6)	:FUDGE RETURN ADDRESS
015046	000002				RTI		:FIXES PS

M05

MAINDEC-11-DCFPL-B
DCFPLB.P11

TEST OF MODF AND MODD
HLT ROUTINE (ERROR TYPEOUT)

MACY11 27(732) 03-SEP-76 14:57 PAGE 64

015050	032737	002000	177570	.TRP:	BIT	#SW10,2#SWR	;BELL ON ERROR?
015056	001405				BEQ	1\$;NO - SKIP
015060	012767	000007	001000		MOV	#BELL,TYPE	;TYPE A BELL
015066	000004	016066			TYPE	TYPE	
015072	004767	000406		1\$:	JSR	PC,ERROR	;COUNT THE NUMBER OF ERRORS
015076	010446				MOV	R4,-(6)	
015100	032737	020000	177570		BIT	#SW13,2#SWR	;SKIP TYPEOUT IF SET
015106	001072				BNE	4\$	
015110	000004	016034			TYPE	RETURN	
015114	016646	000002			MOV	2(6),-(6)	;PUT ADDRESS OF INSTRUCTION ON STACK
015120	162716	000002			SUB	#2,(6)	
015124	011605				MOV	(6) TTY	;TYPE (6) IN OCTAL
015126	004767	000212			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
015132	000004	016042			TYPE	SPACE+3	
015136	010005				MOV	R0,TTY	;TYPE R0 IN OCTAL
015140	004767	000200			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
015144	000004	016043			TYPE	SPACE+4	
015150	012703	0010C2			MOV	#ANS1,R3	;ADDRESS OF DATA
015154	113604				MOVB	2(6)+,R4	;AMOUNT OF DATA IN TABLE
015156	001426				BEQ	3\$	
015160	100016				BPL	2\$;TYPE STACK?
015162	016667	000006	163612		MOV	6(6),ANS1	
015170	016667	000010	163606		MOV	10(6),ANS2	
015176	016667	000012	163602		MOV	12(6),ANS3	
015204	016667	000014	163576		MOV	14(6),ANS4	
015212	042704	177600			BIC	#177600,R4	;CLEAR SIGN
015216	000004	016043		2\$:	TYPE	SPACE+4	
015222	012305				MOV	(3)+,TTY	;TYPE (3)+ IN OCTAL
015224	004767	000114			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
015230	005304				DEC	R4	
015232	001371				BNE	2\$	
015234	005700			3\$:	TST	FPS	
015236	100016				BPL	4\$	
015240	000004	016037			TYPE	SPACE	
015244	170367	163552			STST	FEC	
015250	016705	163546			MOV	FEC,TTY	;TYPE FEC IN OCTAL
015254	004767	000064			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
015260	000004	016042			TYPE	SPACE+3	
015264	016705	163534			MOV	FEA,TTY	;TYPE FEA IN OCTAL
015270	004767	000050			JSR	%7,PRINTR	;TYPE LEADING ZERO'S
015274	012604			4\$:	MOV	(6)+,R4	
015276	005737	177570			TST	2#SWR	;HALT ON ERROR
015302	100001				BPL	+.4	;SKIP IF CONTINUE
015304	000000				HALT		;HALT ON ERROR!
015306	032737	001000	177570		BIT	#SW09,2#SWR	;CHECK FOR INHIBIT LOOP ON ERROR
015314	001001				BNE	+.4	;SKIP IF LOOP ON ERROR
015316	000002				RTI		
015320	105067	163455			CLRB	ICNT+1	
015324	032737	000400	177570		BIT	#SW08,2#SWR	;CHECK FOR LOAD MICROBREAK
015332	001233				BNE	KIT	;BRANCH IF NOT
015334	113703	177570			MOVB	2#SWR,R3	;PUT MICROBREAK ADDRESS IN R3
015340	170003				LDUB		;LOAD MICROBREAK
015342	000627				BR	KIT	;LOOP ON TEST UNTIL NO ERRORS

```

015344 112767 000001 000130 PRINTR: MOVB #1,A4$ ;SET ZERO FILL SWITCH
015352 000402 BR .+6
015354 005067 000122 PRINTS: CLR A4$ ;SUPRESS LEADING ZERO'S
015360 112767 177772 000115 MOVB #-6,A4$+1 ;SET COUNT
015366 010446 MOV R4 -(6) ;SAVE R4
015370 012704 015472 MOV #3$,R4 ;SET POINTER TO FIRST ASCII CHAR.
015374 105014 CLRB (4) ;CLEAR FIRST BYTE
015376 000405 BR 2$ ;ROTATE FIRST BIT
015400 105014 1$: CLRB (4) ;CLEAR BYTE OF CHARACTER
015402 006105 ROL TTY ;ROTATE BIT INTO C
015404 106114 ROLB (4) ;PACK IT
015406 006105 ROL TTY ;ROTATE BIT INTO C
015410 106114 ROLB (4) ;PACK IT
015412 006105 2$: ROL TTY ;ROTATE BIT INTO C
015414 106114 ROLB (4) ;PACK IT
015416 105714 TSTB (4)
015420 001402 BEQ .+6
015422 105267 000054 INCB A4$
015426 105767 000050 TSTB A4$ ;CHECK FILL SWITCH
015432 001402 BEQ .+6
015434 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCII CHAR
015440 105267 000037 INCB A4$+1
015444 001355 BNE 1$ ;REPEAT
015446 022704 015472 CMP #3$,R4
015452 001002 BNE .+6
015454 112724 000060 MOVB #'0,(4)+
015460 105014 CLRB (4)
015462 000004 015472 TYPE ,3$ ;TYPE IT
015466 012604 MOV (6)+,R4 ;RESTORE R4
015470 000207 RTS PC

015472 000004 3$: .BLKW 4
015502 000000 A4$: 0

015504 005267 000364 ERROR: INC ERRORS ;COUNT ERRORS
015510 132737 000001 000041 BITB #1,2#4i ;AUTO MODE?
015516 001412 BEQ 1$ ;NO!
015520 022767 000010 000346 CMP #10,ERRORS ;TOO MANY?
015526 001006 BNE 1$ ;NOT YET
015530 013700 000042 MOV 2#42,R0 ;GET ADDRESS
015534 001403 BEQ 1$ ;FORGET IT IF ZERO
015536 005037 000042 CLR 2#42 ;ZAP 42
015542 004710 JSR PC,(0) ;CALL THE MONITOR
015544 000207 1$: RTS PC ;RETURN
    
```

TEST OF MODF AND MODD POWER DOWN AND UP ROUTINES

```

015634 000000 000240 000306 POWDOWN: MOV      #11UP, @UPVEC      :SET FOR FAST JP
015636 000000 000240 000302 MOV      #340, @UPVEC+2  :PRIO:7
015638 000000 000240 000302 STFPS    -(6)           :GET THE FPS
015640 000000 000240 000302 SETD     AC0, -(6)      :SAVE AC'S
015642 000000 000240 000302 STD     AC1, -(6)
015644 000000 000240 000302 STD     AC2, -(6)
015646 000000 000240 000302 STD     AC3, -(6)
015648 000000 000240 000302 LDD     AC4, AC0
015650 000000 000240 000302 STD     AC0, -(6)
015652 000000 000240 000302 LDD     AC5, AC0
015654 000000 000240 000302 STD     AC0, -(6)
015656 000000 000240 000302 MOV     R0, -(6)       :SAVE REGISTERS
015658 000000 000240 000302 MOV     R1, -(6)
015660 000000 000240 000302 MOV     R2, -(6)
015662 000000 000240 000302 MOV     R3, -(6)
015664 000000 000240 000302 MOV     R4, -(6)
015666 000000 000240 000302 MOV     R5, -(6)
015668 000000 000240 000302 MOV     SP, SAVE6      :SAVE SP
015670 000000 000240 000302 MOV     @POWUP, @UPVEC  :SET UP VECTOR
015672 000000 000240 000302 HALT

015636 016706 000204 POWUP:  MOV     SAVE6, SP      :GET SP
015638 005001 000204 000302 CLR     R1              :WAIT LOOP FOR THE TRY
015640 005201 000204 000302 IS:    INC     R1
015642 001376 000204 000302 BNE     IS
015644 012605 000204 000302 MOV     (6)+, R5       :GET THE REGISTERS
015646 012604 000204 000302 MOV     (6)+, R4
015648 012603 000204 000302 MOV     (6)+, R3
015650 012602 000204 000302 MOV     (6)+, R2
015652 012601 000204 000302 MOV     (6)+, R1
015654 012600 000204 000302 MOV     (6)+, R0
015656 170011 000204 000302 SETD     AC0
015658 172426 000204 000302 LDD     (6)+, AC0      :RESTORE THE AC'S
015660 174005 000204 000302 STD     AC0, AC5
015662 172426 000204 000302 LDD     (6)+, AC0
015664 174004 000204 000302 STD     AC0, AC4
015666 172726 000204 000302 LDD     (6)+, AC3
015668 172626 000204 000302 LDD     (6)+, AC2
015670 172526 000204 000302 LDD     (6)+, AC1
015672 172426 000204 000302 LDD     (6)+, AC0
015674 170126 000204 000302 LDFPS    (6)+         :RESTORE FPS
015676 012777 015546 000140 MOV     @POWDOWN, @DOWNVEC :SET UP THE POWER DOWN VECTOR
015678 012777 000340 000134 MOV     #340, @DOWNVEC+2
015680 000004 015730 000134 TYPE    ..+2           :.ASCIZ (15)(12)"POWER"
015682 000002 000000 000134 RTI

015742 000000 000776 ILLUP: HALT
015744 000776 000776 BR      .-2           : THE POWER UP SEQUENCE WAS STARTED
: BEFORE THE POWER DOWN WAS COMPLETE

```

TYPE AC...

TES OF MODF AND MODD

010546			.IOT:	MOV	TTY -(6)	:SAVE TTY
017605	000002			MOV	22(6), TTY	:GET ADDRESS TO BE TYPED
105715			18:	TSTB	(TTY)	:TERMINATOR?
001406				BFC	28	
112537	177566			MOV8	(TTY)+28177566	:LOAD AND TYPE THE CHARACTER
105737	177564			TSTB	28177564	:IS THE PRINTER READY?
100375				BPL	4	
000770				BPL	4	
017646	000002		25:	MOV	22(6), -6	:GET THE NEXT CHARACTER
062766	000002	000004		ADD	2(6)+6	:GET ADDRESS TO BE TYPED
022666	000002			BFC	2(6)+26	:ADD 2 TO THE ADDRESS
001008				BFC	2(6)+26	:IS IT "+2?"
062705	000002			ADD	2(6)+26	:NO
042705	000001			BFC	2(6)+26	:ADD 2 TO THE ADDRESS
010566	007002			MOV	2(6)+26	:BACK UP TO AN EVEN BYTE
012605			38:	MOV	6(6)+26	:RESTORE ADDRESS
000002				MOV	6(6)+26	:RESTORE TTY
				MOV	6(6)+26	:RETURN
005015	000		RETJRN:	AS017	1111	:RETURN AND LINEFEED
020012	020040		SPACE:	AS017	1111	:RETURN AND 3 SPACES
016046	000000			SAVE6:	0	
172160				FPTADR:	244	:FLOATING POINT ADDRESS ON THE : 20
000244	000246			FVVEC:	244	:FLOATING POINT VECTOR ADDRESS
000024	000026			DNVEC:	244	:POWER DOWN VECTOR ADDRESS
000024	000026			UPVEC:	244	:POWER UP VECTOR ADDRESS
000000				TYPE:	0	
000000				ABB:	0	
000000				AC:	0	:JOB ADDRESS
000000				PRCS:	0	:PROR COL
000000				TES:	0	:PERATION COL
000000				.E.C	0	

CROSS REFERENCE TABLE -- USER SYMBOLS

TEST OF MODF AND MODC	TEST OF MODF AND MODC	TEST OF MODF AND MODC
001000	467	478
001001	473	744
001002	479	782
001003	485	820
001004	491	858
001005	497	896
001006	503	934
001007	509	972
001008	515	1010
001009	521	516
001010	527	1048
001011	533	1086
001012	539	1124
001013	545	1162
001014	551	1200
001015	557	1238
001016	563	1276
001017	569	1314
001018	575	1352
001019	581	1390
001020	587	554
001021	593	1432
001022	599	1474
001023	605	1516
001024	611	1558
001025	617	1600
001026	623	1642
001027	629	1684
001028	635	1726
001029	641	1768
001030	647	1810
001031	653	1852
001032	659	1894
001033	665	1936
001034	671	1978
001035	677	2020
001036	683	2062
001037	689	2104
001038	695	2146
001039	701	2188
001040	707	2230
001041	713	2272
001042	719	2314
001043	725	2356
001044	731	2398
001045	737	2440
001046	743	2482
001047	749	2524
001048	755	2566
001049	761	2608
001050	767	2650
001051	773	2692
001052	779	2734
001053	785	2776
001054	791	2818
001055	797	2860
001056	803	2902
001057	809	2944
001058	815	2986
001059	821	3028
001060	827	3070
001061	833	3112
001062	839	3154
001063	845	3196
001064	851	3238
001065	857	3280
001066	863	3322
001067	869	3364
001068	875	3406
001069	881	3448
001070	887	3490
001071	893	3532
001072	899	3574
001073	905	3616
001074	911	3658
001075	917	3700
001076	923	3742
001077	929	3784
001078	935	3826
001079	941	3868
001080	947	3910
001081	953	3952
001082	959	3994
001083	965	4036
001084	971	4078
001085	977	4120
001086	983	4162
001087	989	4204
001088	995	4246
001089	1001	4288
001090	1007	4330
001091	1013	4372
001092	1019	4414
001093	1025	4456
001094	1031	4498
001095	1037	4540
001096	1043	4582
001097	1049	4624
001098	1055	4666
001099	1061	4708
001100	1067	4750

0022610	818	823
0022700	856	861
0022800	894	899
0022900	932	937
0023000	970	975
0023100	1008	1013
0023200	1046	1051
0023300	1084	1089
0023400	1122	1127
0023500	1160	1166
0023600	1198	1204
0023700	1236	1242
0023800	1274	1280
0023900	1312	1318
0024000	1350	1356
0024100	1388	1394
0024200	1426	1432
0024300	1464	1470
0024400	1502	1508
0024500	1540	1546
0024600	1578	1584
0024700	1616	1622
0024800	1654	1660
0024900	1692	1698
0025000	1730	1736
0025100	1768	1774
0025200	1806	1812
0025300	1844	1850
0025400	1882	1888
0025500	1920	1926
0025600	1958	1964
0025700	1996	2002
0025800	2034	2040
0025900	2072	2078
0026000	2110	2116
0026100	2148	2154
0026200	2186	2192
0026300	2224	2230
0026400	2262	2268
0026500	2300	2306
0026600	2338	2344
0026700	2376	2382
0026800	2414	2420
0026900	2452	2458
0027000	2490	2496
0027100	2528	2534
0027200	2566	2572
0027300	2604	2610
0027400	2642	2648
0027500	2680	2686
0027600	2718	2724
0027700	2756	2762
0027800	2794	2800
0027900	2832	2838
0028000	2870	2876
0028100	2908	2914
0028200	2946	2952
0028300	2984	2990
0028400	3022	3028
0028500	3060	3066
0028600	3098	3104
0028700	3136	3142
0028800	3174	3180
0028900	3212	3218
0029000	3250	3256
0029100	3288	3294
0029200	3326	3332
0029300	3364	3370
0029400	3402	3408
0029500	3440	3446
0029600	3478	3484
0029700	3516	3522
0029800	3554	3560
0029900	3592	3598
0030000	3630	3636

31:0+ 31:2+ 31:2+
31:6+
3056 3064 3067 3081

27.732)

MACY11
SYMBOLS
MODS
AND
MODF
TABLE
REFERENCE
OF
CROSS
REFERENCES

381*	450*	2993*	3042	3120*	3137	3156*								
392*	3138	3148*	3149*	3155*										
393*	3139	3154*												
384*	3006*	3045*	3078*	3140	3153*									
385*	3033	3046*	3053*	3057*	3068*	3085	3086*	3104	3109*	3141	3152*			
386*	3142	3151*												
3143*	3147	3196*												
3011	3017*	3025												
376*	465	503	541	579	617	655	693	731	769	807	845	883	921	959
921	959	997	1035	1073	1111	1149	1203	1257	1311	1365	1419	1473	1527	1581
1527	1581	1635	1689	1743	1797	1851	1905	1959	2013	2067	2121	2175	2229	2283
2229	2283	2337	2391	2445	2499	2553	2607	2661	2715	2769	2823	2877	2931	2985
2931	2973													
388*	435*	445*	3143	3147*										
3041	3044	3054	3061	3055	3193*									
405*														
406*														
374*	375	2980	2985	3002	3004	3006	3008	3010	3028	3034	3063	3072		
3076	3078													
403*	3002	3076												
402*	3072													
401*	2980	3028												
400*	3010													
399*	2985													
398*	3034													
397*	3008													
396*														
3014	3205*													
2987*	3202*													
387*	3039*	3042*	3055*	3063*	3066*	3090*	3092*	3094*	3174	3175*	3176	3178		
3186*	3187*	3188	3189*											
378*	2983	3031	3036	3041	3044	3054	3061	3065	3108	3169				
3125*	3126*	3144*	3200*											
438*	446	2991	3000*											
410*	411	412*	416*	421*	471	482	486	490	494	509	520	531	542	553
528	532	547	552	562	566	570	585	596	600	604	608	612	616	620
624	638	642	646	661	672	676	690	694	699	703	707	711	715	719
722	737	746	752	756	760	775	786	790	794	798	802	806	810	814
828	832	836	851	862	866	870	874	889	900	904	908	912	916	920
927	938	942	946	950	965	976	980	984	988	992	996	1000	1004	1008
1022	1026	1041	1052	1056	1060	1064	1079	1090	1094	1098	1102	1106	1110	1114
1128	1132	1136	1140	1155	1167	1171	1175	1179	1183	1187	1191	1195	1199	1203
1209	1221	1225	1229	1233	1237	1241	1245	1249	1263	1275	1279	1283	1287	1291
1287	1291	1295	1299	1303	1317	1329	1333	1337	1341	1345	1349	1353	1357	1361
1357	1371	1383	1387	1391	1395	1399	1403	1407	1411	1415	1419	1423	1427	1431
1445	1449	1453	1457	1461	1465	1479	1491	1495	1499	1503	1507	1511	1515	1519
1515	1519	1533	1545	1549	1553	1557	1561	1565	1569	1573	1577	1581	1585	1589
1603	1607	1611	1615	1619	1623	1627	1641	1653	1657	1661	1665	1669	1673	1677
1673	1677	1681	1695	1707	1711	1715	1719	1723	1727	1731	1735	1739	1743	1747
1761	1765	1769	1773	1777	1781	1785	1789	1803	1815	1819	1823	1827	1831	1835
1831	1835	1839	1843	1857	1869	1873	1877	1881	1885	1889	1893	1897	1901	1905
1911	1923	1927	1931	1935	1939	1943	1947	1951	1965	1969	1973	1977	1981	1985
1989	1993	1997	2001	2005	2009	2013	2017	2021	2025	2029	2033	2037	2041	2045
2053	2057	2061	2065	2069	2073	2077	2081	2085	2089	2093	2097	2101	2105	2109
2113	2117	2121	2125	2129	2133	2137	2141	2145	2149	2153	2157	2161	2165	2169
2173	2177	2181	2185	2189	2193	2197	2201	2205	2209	2213	2217	2221	2225	2229

2305	2309	2313	2317	2321	2325	2329	2343	2355	2359	2363	2367	2371
2375	2379	2383	2397	2409	2413	2417	2421	2425	2429	2433	2437	2441
2463	2467	2471	2475	2479	2483	2487	2491	2495	2499	2503	2507	2511
2533	2537	2541	2545	2559	2571	2575	2579	2583	2587	2591	2595	2599
2613	2625	2629	2633	2637	2641	2645	2649	2653	2657	2661	2665	2669
2691	2695	2699	2703	2707	2721	2733	2737	2741	2745	2749	2753	2757
2761	2775	2787	2791	2795	2799	2803	2807	2811	2815	2819	2823	2827
2849	2853	2857	2861	2865	2869	2873	2877	2881	2885	2889	2893	2897
2919	2923	2937	2949	2953	2957	2961	2965	2969	2973	2977	2981	2985
3082	3097	3100	3105	3112*	3169	3173	3180	3195*	3201	3205	3209	3213
453	3002*											
449	3174*											
451	3028*											
2982*	2983	3030*	3031	3201*								

3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100

ADD BEJ	3183 471 585 699 813 927 1041 1155 1241 1337 1425 1511 1607 1695 1781 1877 1965 2051 2147 2235 2321 2417 2505 2591 2687 2775 2861 2957 3117 3053 2989 3101 2980 3116 2981 2988 439 1008 1754 2564 457 3075 470 584 698 812 926 1040 1154 1240 1336 1424 1510 1606 1694 1780	3196 482 596 710 824 938 1052 1167 1245 1341 1437 1515 1611 1707 1785 1881 1977 2055 2151 2247 2325 2421 2517 2595 2691 2787 2865 2961 3121 3187	486 600 714 828 942 1056 1171 1249 1345 1441 1519 1615 1711 1789 1885 1981 2059 2155 2251 2329 2425 2521 2599 2695 2791 2869 2965 3177	490 604 718 832 946 1060 1175 1263 1349 1445 1545 1619 1715 1803 1899 1985 2073 2159 2255 2343 2429 2525 2613 2699 2795 2883 2969	494 608 722 836 950 1064 1179 1275 1353 1453 1549 1623 1723 1815 1893 1989 2085 2163 2259 2355 2433 2533 2629 2703 2799 2895 2973	509 623 737 851 965 1079 1183 1279 1357 1453 1549 1627 1723 1819 1897 1993 2089 2167 2263 2359 2437 2533 2629 2707 2803 2899 2977	520 634 748 862 976 1090 1187 1283 1371 1457 1553 1641 1727 1823 1911 1997 2093 2181 2267 2363 2451 2537 2633 2721 2807 2903 2994	524 638 752 866 980 1094 1191 1287 1383 1461 1557 1653 1731 1827 1911 1997 2097 2193 2271 2367 2451 2541 2637 2733 2811 2907 3003	528 642 756 870 984 1098 1195 1291 1387 1465 1561 1657 1735 1831 1927 2005 2097 2197 2275 2371 2467 2545 2641 2737 2815 2911 3005	532 646 760 874 988 1102 1209 1295 1391 1479 1565 1661 1749 1835 1931 2019 2097 2201 2289 2375 2471 2559 2645 2741 2829 2915 3013	547 661 775 889 1003 1117 1225 1299 1395 1491 1569 1665 1761 1839 1935 2031 2109 2205 2301 2379 2475 2571 2649 2745 2841 2919 3025	558 672 786 900 1014 1128 1225 1303 1399 1495 1573 1669 1765 1843 1939 2035 2113 2209 2305 2383 2479 2575 2653 2749 2845 2923 3029	562 676 790 904 1018 1132 1229 1317 1403 1499 1587 1673 1769 1857 1943 2039 2127 2213 2309 2397 2483 2579 2667 2753 2849 2937 3047	566 680 794 908 1022 1136 1233 1329 1407 1503 1599 1677 1773 1869 1947 2043 2139 2217 2313 2409 2487 2583 2679 2757 2853 2949 3097	570 684 798 912 1026 1140 1237 1333 1411 1507 1599 1681 1777 1873 1951 2047 2143 2221 2317 2413 2491 2587 2683 2761 2857 2953 3103
BIC BIS BISB BIT BITB BNE BP BR	3117 3053 2989 3101 2980 3116 2981 2988 439 1008 1754 2564 457 3075 470 584 698 812 926 1040 1154 1240 1336 1424 1510 1606 1694 1780	2985 2986 3048 476 1046 1808 2618 458 3087 481 595 709 823 937 1051 1166 1244 1340 1436 1514 1610 1706 1784	3002 3009 3060 514 1084 1862 2672 2984 3089 485 599 713 827 941 1055 1170 1248 1344 1440 1518 1614 1710 1788	3008 3011 3070 552 1122 1916 2726 3083 3107 489 603 717 831 945 1059 1174 1262 1348 1444 1544 1618 1714 1802	3010 3015 3180 590 1160 1970 2780 3122 493 607 721 835 949 1063 1178 1274 1352 1448 1548 1622 1718 1814	3028 3035 628 666 704 742 780 818 856 894 932 970	3034 3058 666 704 742 780 818 856 894 932 970	3072 3073 3077 780 818 856 894 932 970	3076 3077 3103 3105 3119 3150 3185	3103 3105 3119 3150 3185	3105 3119 3150 3185	3150 3185	3185	970 1700 2510	
CL CLRB CMP	470 584 698 812 926 1040 1154 1240 1336 1424 1510 1606 1694 1780	481 595 709 823 937 1051 1166 1244 1340 1436 1514 1610 1706 1784	485 599 713 827 941 1055 1170 1248 1344 1440 1518 1614 1710 1788	489 603 721 835 949 1063 1178 1274 1352 1448 1548 1622 1718 1814	508 622 736 850 964 1078 1182 1278 1356 1452 1548 1626 1722 1818	519 633 747 861 975 1089 1186 1282 1370 1456 1552 1640 1736 1822	523 637 751 865 979 1093 1190 1286 1382 1468 1564 1660 1748 1834	527 641 755 869 983 1097 1194 1290 1386 1472 1568 1664 1750 1836	531 645 759 873 987 1101 1208 1294 1390 1478 1564 1660 1748 1834	546 660 774 888 1002 1116 1220 1298 1394 1480 1568 1664 1750 1838	557 671 785 899 1013 1127 1224 1298 1394 1480 1568 1664 1750 1838	561 675 789 903 1017 1131 1228 1316 1402 1498 1586 1672 1768 1856	565 679 793 907 1021 1135 1232 1328 1406 1502 1598 1676 1772 1868	569 683 797 911 1025 1139 1236 1332 1410 1506 1602 1680 1776 1872	

STF	474	475	512	513	550	551	588	589	626	627	664	665	702	703	740
	741	778	779	816	817	854	855	892	893	930	931	968	969	1006	1007
STFPS	1044	1045	1082	1083	1120	1121									
	417	469	507	545	583	621	659	697	735	773	811	849	887	925	963
	1001	1039	1077	1115	1153	1207	1261	1315	1369	1423	1477	1531	1585	1639	1693
	1747	1801	1855	1909	1963	2017	2071	2125	2179	2233	2287	2341	2395	2449	2503
	2557	2611	2665	2719	2773	2827	2881	2935	3127						
STST	418	3062													
SJB	3038														
TRAP	37E														
TST	437	3024	3059	3069											
TSTB	3012	3096	3099	3176	3179										
.ASCIZ	3170	3192	3193												
.BLKW	3112														
.ENABL	371														
.END	3206														
.ENDC	474	512	550	588	626	664	702	740	778	816	854	892	930	968	1006
	1044	1082	1120	1158	1212	1266	1320	1374	1428	1482	1536	1590	1644	1698	1752
	1806	1860	1914	1968	2022	2076	2130	2184	2238	2292	2346	2400	2454	2508	2562
	2616	2670	2724	2778	2832	2886	2940								
.EVEN	3170	3195													
.IFNZ	474	512	550	588	626	664	702	740	778	816	854	892	930	968	1006
	1044	1082	1120	1158	1212	1266	1320	1374	1428	1482	1536	1590	1644	1698	1752
	1806	1860	1914	1968	2022	2076	2130	2184	2238	2292	2346	2400	2454	2508	2562
	2616	2670	2724	2778	2832	2886	2940								
.IIF	470	508	546	584	622	660	698	736	774	812	850	888	926	964	1002
	1040	1078	1116	1154	1208	1262	1316	1370	1424	1478	1532	1586	1640	1694	1748
	1802	1856	1910	1964	2018	2072	2126	2180	2234	2288	2342	2396	2450	2504	2558
	2612	2666	2720	2774	2828	2882	2936								
.LIST	327	371	411	421	459	497	535	573	611	649	687	725	763	801	839
	877	915	953	991	1029	1067	1105	1143	1197	1251	1305	1359	1413	1467	1521
	1575	1629	1683	1737	1791	1845	1899	1953	2007	2061	2115	2169	2223	2277	2331
	2385	2439	2493	2547	2601	2655	2709	2763	2817	2871	2925	2979	3028	3081	3125
	3170	3174													
.MACR	371	459													
.MACRO	371														
.NLIST	327	371	411	421	459	497	535	573	611	649	687	725	763	801	839
	877	915	953	991	1029	1067	1105	1143	1197	1251	1305	1359	1413	1467	1521
	1575	1629	1683	1737	1791	1845	1899	1953	2007	2061	2115	2169	2223	2277	2331
	2385	2439	2493	2547	2601	2655	2709	2763	2817	2871	2925	2979	3028	3081	3125
	3170	3174													
.PAGE	497	535	573	611	649	687	725	763	801	839	877	915	953	991	1029
	1067	1105	1143	1197	1251	1305	1359	1413	1467	1521	1575	1629	1683	1737	1791
	1845	1899	1953	2007	2061	2115	2169	2223	2277	2331	2385	2439	2493	2547	2601
	2655	2709	2763	2817	2871	2925	2979								
.REM	330														
.REPT	2	411	1143												
.SBTTL	327	371	421	459	2979	3028	3081	3125	3174						
.TITLE	327														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

SE:ERFZ EN:ABS=DSAM:DCFPLB.F11
S

