FP11

MULF, MULD

MD-11-DCFPF-B

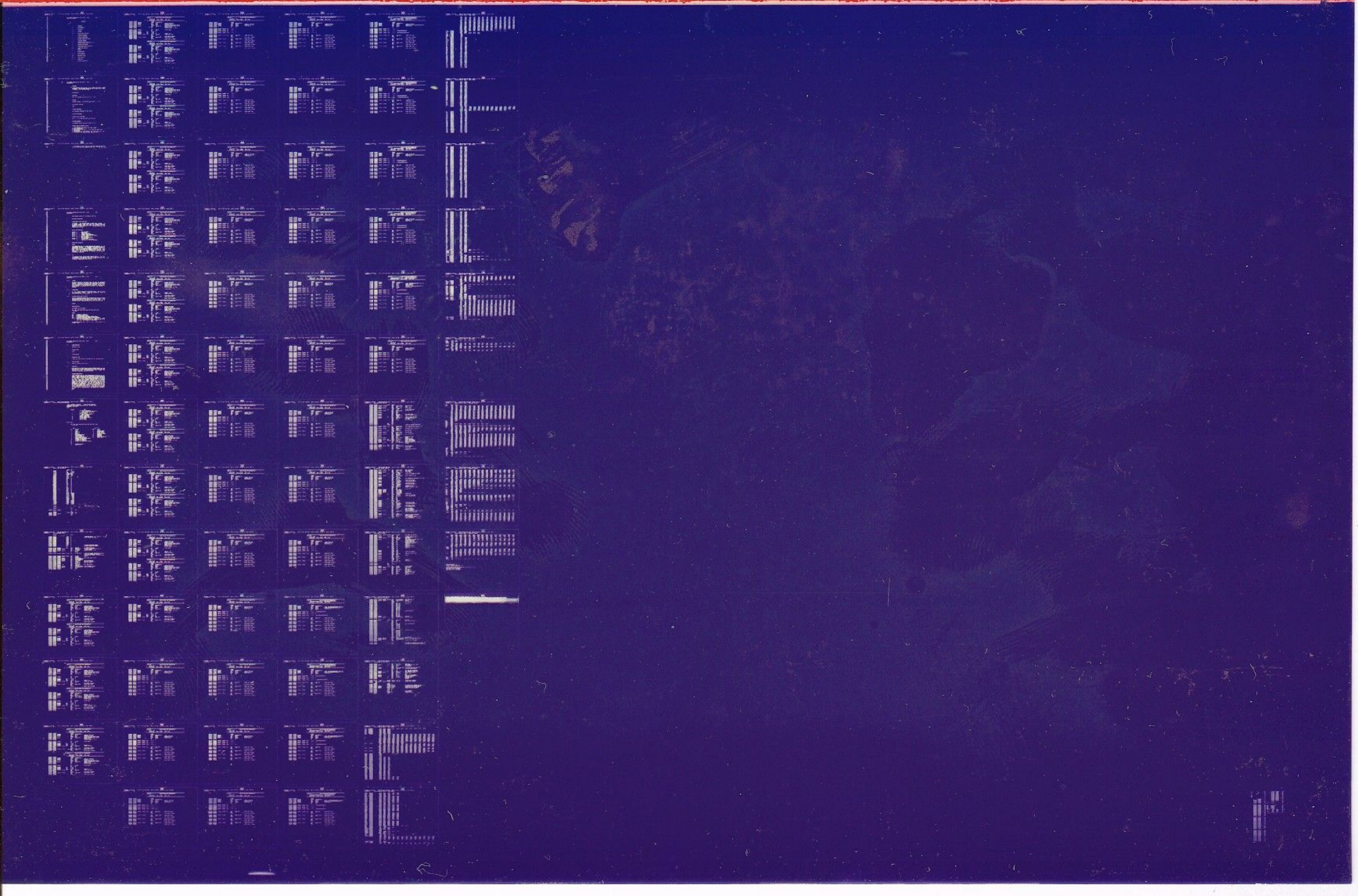
EP-DCFPF-B-DL-A

OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.



FP11 BASIC INSTRUCTION TEST DCFPA - DCFP_ PAGE 2
TABLE OF CONTENTS

CONTENTS

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

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 3
DESCRIPTION

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 0 - 17776

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
2) LOAD ADDRESS 200.
3) SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE
4) PRESS START.
5) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
6) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

[illegible]

7) 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

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 JB REGISTER
```

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1) IF A HLT IS EXECUTED. THE SUBTEST WILL BE LOOPED UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW 14 IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW 13 ON A 1.

5.2.3 TRTRAP

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

5.2.4 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERUPTS 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 R0.

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.

FP11 BASIC INSTRUCTION TEST DCFPA - DCFPL PAGE 6
DESCRIPTION

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 INITALLY 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 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 LOOPED UPON UNTIL SW(8) IS PUT ON A 0 OR THE RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE PROGRAM WILL BE RUN AS USUAL.

.ERROR *

```
.TITLE  MAINDEC-11-DCFPF-B      TEST OF MULF AND MULD
;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
;PROGRAM BY BOB BRAIN
.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
```

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

```
                                .ENABL  ABS
000001                          N=      1
177776                          PS=     177776
177570                          SWR=    177570
177570                          DISPLAY=SWR
104400                          SCOPE=  TRAP
104000                          HLT=    EMT
000004                          TYPE=   IOT
000007                          BELL=   7
000000                          FPS=    %0
000000                          R0=     %0
000001                          R1=     %1
000002                          R2=     %2
000003                          R3=     %3
000004                          R4=     %4
000005                          R5=     %5
000005                          SP=     %6
000006                          PC=     %7
000007
000000                          AC0=    %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                          LCDB=   170003
170005                          STAD=   170005
170007                          STQD=   170007
170006                          MRS=    170006
170004                          LOSC=   170004

000000                          .=      0                       ;TRAP CATCHER FROM 0 - 776

000200                          .=      200

000200  000167  000622                  JMP     BEG

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

```
                001000                  .=      1000

001000  000000                 ICNT:  0                        ;ITERATION COUNT - LH   TEST NO. - RH
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  011042        MOV     #6,YESRT         ;FUDGE IN RTT IF 11/45
001052  000403                       BR      BEGIN

001054  016737  012176  000010 M1120: MOV     FPTADR,@#10      ;LOAD THE ILLEGAL INSTRUCTION VECTOR
                                                               ; WITH THE ADDRESS OF THE FPU.
                                                               ; THE FPU WILL HANDLE THE BAD OPCODES
001062  012737  000006  000004 BEGIN: MOV     #6,@#4           ;RESET 4
001070  012706  000600               MOV     #600,SP
001074  012737  012114  000014        MOV     #YESRT,@#14      ;SET TRACE TRAP VECTOR
001102  012777  012754  012154        MOV     #POWDWN,@DWNVEC
001110  012777  000340  012150        MOV     #340,@DWNVEC+2
001116  012737  013154  000020        MOV     #.IOT,@#20       ;SET UP VECTOR 20
001124  012700  000030               MOV     #30,R0           ;SET R0 TO VECTOR 30
001130  012720  012256               MOV     #.TRP,(0)+       ;SET EMT VECTOR
001134  012720  000340               MOV     #340,(0)+
001140  012720  012116               MOV     #.EMT,(0)+       ;SET TRAP VECTOR
001144  012710  000340               MOV     #340,(0)
001150  012777  000760  012102        MOV     #FLTERR,@FPVECT  ;LOAD INTERUPT VECTOR
001156  012777  000340  012076        MOV     #340,@FPVECT+2   ;LOCK UP PROCESSOR
001164  005067  177610               CLR     ICNT
001170  005067  012134               CLR     LAC
```

```
                                ;*****************************************************************
                                ;TEST 1         TEST OF MULF FPU INSTRUCTION
                                ;    0 * 0 = 0
                                ;    USING AC0    FPS = 47404    FEC = N/A
                                ;*****************************************************************
001174  104400                          SCOPE
001176  170127  047400                  LDFPS   #47404857760
001202  172467  000024                  LDF     N1,0            ;LOAD 0 INTO AC1
001206  171067  000024                  MULF    M1,0            ;MULTIPLY 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 RESULT
001230  000406                          BR      01
001232  000000  000000          N1:     .FLT2   0
001236  000000  000000          M1:     .FLT2   0
001242  000000  000000          AN1:    .FLT2   0               ;RESULT = 0
001246  026767  177770  177526  01:     CMP     AN1,ANS1        ;CHECK LEFT HALF
001254  001401                          BEQ     .+4
001256  104002                          HLT+2                   ;LEFT HALF IS WRONG
001260  026767  177760  177516          CMP     AN1+2,ANS2      ;CHECK RIGHT HALF
001266  001401                          BEQ     .+4
001270  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 2         TEST OF MULF FPU INSTRUCTION
                                ;    0 * 1. = 0
                                ;    USING AC0    FPS = 47404    FEC = N/A
                                ;*****************************************************************
001272  104400                          SCOPE
001274  170127  047400                  LDFPS   #47404857760
001300  172467  000024                  LDF     N2,0            ;LOAD 0 INTO AC2
001304  171067  000024                  MULF    M2,0            ;MULTIPLY 0 BY 1.
001310  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001312  022700  047404                  CMP     #47404,FPS      ;CHECK FLOATING POINT STATUS
001316  001401                          BEQ     .+4             ;BRANCH IF OK
001320  104000                          HLT                     ;FPS NOT EQUAL TO 47404

001322  174067  177454                  STF     0,ANS1          ;STORE RESULT
001326  000406                          BR      02
001330  000000  000000          N2:     .FLT2   0
001334  040200  000000          M2:     .FLT2   1.
001340  000000  000000          AN2:    .FLT2   0               ;RESULT = 0
001344  026767  177770  177430  02:     CMP     AN2,ANS1        ;CHECK LEFT HALF
001352  001401                          BEQ     .+4
001354  104002                          HLT+2                   ;LEFT HALF IS WRONG
001356  026767  177760  177420          CMP     AN2+2,ANS2      ;CHECK RIGHT HALF
001364  001401                          BEQ     .+4
001366  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                        ;*****************************************************************
                                        ;TEST 3         TEST OF MULF FPU INSTRUCTION
                                        ;    1. * 1. = 1.
                                        ;    USING AC0    FPS = 47400    FEC = N/A
                                        ;*****************************************************************
001370  104400                          SCOPE
001372  170127  047400                  LDFPS   #47400&57760
001376  172467  000024                  LDF     N3,0            ;LOAD 1. INTO AC3
001402  171067  000024                  MULF    M3,0            ;MULTIPLY 1. BY 1.
001406  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001410  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
001414  001401                          BEQ     .+4             ;BRANCH IF OK
001416  104000                          HLT                     ;FPS NOT EQUAL TO 47400

001420  174067  177356                  STF     0,ANS1          ;STORE RESULT
001424  000406                          BR      C3
001426  040200  000000          N3:     .FLT2   1.
001432  040200  000000          M3:     .FLT2   1.
001436  040200  000000          AN3:    .FLT2   1.              ;RESULT = 1.
001442  026767  177770  177332  C3:     CMP     AN3,ANS1        ;CHECK LEFT HALF
001450  001401                          BEQ     .+4
001452  104002                          HLT+2                   ;LEFT HALF IS WRONG
001454  026767  177766  177322          CMP     AN3+2,ANS2      ;CHECK RIGHT HALF
001462  001401                          BEQ     .+4
001464  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                        ;*****************************************************************
                                        ;TEST 4         TEST OF MULF FPU INSTRUCTION
                                        ;    -1. * 1. = -1.
                                        ;    USING AC0    FPS = 47410    FEC = N/A
                                        ;*****************************************************************
001466  104400                          SCOPE
001470  170127  047400                  LDFPS   #47410&57760
001474  172467  000024                  LDF     N4,0            ;LOAD -1. INTO AC4
001500  171067  000024                  MULF    M4,0            ;MULTIPLY -1. BY 1.
001504  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001506  022700  047410                  CMP     #47410,FPS      ;CHECK FLOATING POINT STATUS
001512  001401                          BEQ     .+4             ;BRANCH IF OK
001514  104000                          HLT                     ;FPS NOT EQUAL TO 47410

001516  174067  177260                  STF     0,ANS1          ;STORE RESULT
001522  000406                          BR      C4
001524  140200  000000          N4:     .FLT2   -1.
001530  040200  000000          M4:     .FLT2   1.
001534  140200  000000          AN4:    .FLT2   -1.             ;RESULT = -1.
001540  026767  177770  177234  C4:     CMP     AN4,ANS1        ;CHECK LEFT HALF
001546  001401                          BEQ     .+4
001550  104002                          HLT+2                   ;LEFT HALF IS WRONG
001552  026767  177760  177224          CMP     AN4+2,ANS2      ;CHECK RIGHT HALF
001560  001401                          BEQ     .+4
001562  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 5         TEST OF MULF FPU INSTRUCTION
                                ;    1. * -1. = -1.
                                ;    USING AC0    FPS = 47410    FEC = N/A
                                ;*****************************************************************
001564  104400                          SCOPE
001566  170127  047400                  LDFPS   #47410&57760
001572  172467  000024                  LDF     N5,0                ;LOAD 1. INTO AC5
001576  171067  000024                  MULF    M5,0                ;MULTIPLY 1. BY -1.
001602  170200                          STFPS   FPS                 ;STORE FLOATING POINT STATUS
001604  022700  047410                  CMP     #47410,FPS          ;CHECK FLOATING POINT STATUS
001610  001401                          BEQ     .+4                 ;BRANCH IF OK
001612  104000                          HLT                         ;FPS NOT EQUAL TO 47410

001614  174067  177162                  STF     0,ANS1              ;STORE RESULT
001620  000406                          BR      05
001622  040200  000000          N5:     .FLT2   1.
001626  140200  000000          M5:     .FLT2   -1.
001632  140200  000000          ANS:    .FLT2   -1.                 ;RESULT = -1.
001636  026767  177770  177136  05:     CMP     ANS,ANS1            ;CHECK LEFT HALF
001644  001401                          BEQ     .+4
001646  104002                          HLT+2                       ;LEFT HALF IS WRONG
001650  026767  177760  177126          CMP     ANS+2,ANS2          ;CHECK RIGHT HALF
001656  001401                          BEQ     .+4
001660  104002                          HLT+2                       ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 6         TEST OF MULF FPU INSTRUCTION
                                ;    -1. * -1. = 1.
                                ;    USING AC0    FPS = 47400    FEC = N/A
                                ;*****************************************************************
001662  104400                          SCOPE
001664  170127  047400                  LDFPS   #47400&57760
001670  172467  000024                  LDF     N6,0                ;LOAD -1. INTO AC6
001674  171067  000024                  MULF    M6,0                ;MULTIPLY -1. BY -1.
001700  170200                          STFPS   FPS                 ;STORE FLOATING POINT STATUS
001702  022700  047400                  CMP     #47400,FPS          ;CHECK FLOATING POINT STATUS
001706  001401                          BEQ     .+4                 ;BRANCH IF OK
001710  104000                          HLT                         ;FPS NOT EQUAL TO 47400

001712  174067  177064                  STF     0,ANS1              ;STORE RESULT
001716  000406                          BR      06
001720  140200  000000          N6:     .FLT2   -1.
001724  140200  000000          M6:     .FLT2   -1.
001730  040200  000000          AN6:    .FLT2   1.                  ;RESULT = 1.
001734  026767  177770  177040  06:     CMP     AN6,ANS1            ;CHECK LEFT HALF
001742  001401                          BEQ     .+4
001744  104002                          HLT+2                       ;LEFT HALF IS WRONG
001746  026767  177760  177030          CMP     AN6+2,ANS2          ;CHECK RIGHT HALF
001754  001401                          BEQ     .+4
001756  104002                          HLT+2                       ;RIGHT HALF IS WRONG
```

,

```
                                        ;*****************************************************************
                                        ;TEST 7                 TEST OF MULF FPU INSTRUCTION
                                        ;        2 * 1 = 2
                                        ;        USING AC0      FPS = 47400     FEC = N/A
                                        ;*****************************************************************
001760  104400                                  SCOPE
001762  170127  047400                          LDFPS   #47400&57760
001766  172467  000024                          LDF     N7,0                    ;LOAD 2 INTO AC7
001772  171067  000024                          MULF    M7,0                    ;MULTIPLY 2 BY 1
001776  170200                                  STFPS   FPS                     ;STORE FLOATING POINT STATUS
002000  022700  047400                          CMP     #47400,FPS              ;CHECK FLOATING POINT STATUS
002004  001401                                  BEQ     .+4                     ;BRANCH IF OK
002006  104000                                  HLT                             ;FPS NOT EQUAL TO 47400

002010  174067  176766                          STF     0,ANS1                  ;STORE RESULT
002014  000406                                  BR      C7
002016  040400  000000          N7:             .FLT2   2
002022  040200  000000          M7:             .FLT2   1
002026  040400  000000          AN7:            .FLT2   2                       ;RESULT = 2
002032  026767  177770  176742  C7:             CMP     AN7,ANS1                ;CHECK LEFT HALF
002040  001401                                  BEQ     .+4
002042  104002                                  HLT+2                           ;LEFT HALF IS WRONG
002044  026767  177760  176732                  CMP     AN7+2,ANS2              ;CHECK RIGHT HALF
002052  001401                                  BEQ     .+4
002054  104002                                  HLT+2                           ;RIGHT HALF IS WRONG

                                        ;*****************************************************************
                                        ;TEST 10                        TEST OF MULF FPU INSTRUCTION
                                        ;        3 * 1 = 3
                                        ;        USING AC0      FPS = 47400     FEC = N/A
                                        ;*****************************************************************
002056  104400                                  SCOPE
002060  170127  047400                          LDFPS   #47400&57760
002064  172467  000024                          LDF     N10,0                   ;LOAD 3 INTO AC10
002070  171067  000024                          MULF    M10,0                   ;MULTIPLY 3 BY 1
002074  170200                                  STFPS   FPS                     ;STORE FLOATING POINT STATUS
002076  022700  047400                          CMP     #47400,FPS              ;CHECK FLOATING POINT STATUS
002102  001401                                  BEQ     .+4                     ;BRANCH IF OK
002104  104000                                  HLT                             ;FPS NOT EQUAL TO 47400

002106  174067  176670                          STF     0,ANS1                  ;STORE RESULT
002112  000406                                  BR      C10
002114  040500  000000          N10:            .FLT2   3
002120  040200  000000          M10:            .FLT2   1
002124  040500  000000          AN10:           .FLT2   3                       ;RESULT = 3
002130  026767  177770  176644  C10:            CMP     AN10,ANS1               ;CHECK LEFT HALF
002136  001401                                  BEQ     .+4
002140  104002                                  HLT+2                           ;LEFT HALF IS WRONG
002142  026767  177760  176634                  CMP     AN10+2,ANS2             ;CHECK RIGHT HALF
002150  001401                                  BEQ     .+4
002152  104002                                  HLT+2                           ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 11                TEST OF MULF FPU INSTRUCTION
                        ;    4 * 1 = 4
                        ;    USING AC0    FPS = 47400    FEC = N/A
                        ;*****************************************************************
002250  104401                  SCOPE
002252  170127  047400          LDFPS   #47400&57760
002256  172467  000024          LDF     N11,0           ;LOAD 4 INTO AC11
002262  171067  000024          MULF    N11,0           ;MULTIPLY 4 BY 1
002266  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002270  022700  047400          CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002274  001401                  BEQ     .+4             ;BRANCH IF OK
002276  104000                  HLT                     ;FPS NOT EQUAL TO 47400

002300  174067  176572          STF     0,ANS1          ;STORE RESULT
002304  000406                  BR      C11
002306  040600  000000  N11:    .FLT2   4
002312  040200  000000  N11+2:  .FLT2   1
002316  040600  000000  ANS11:  .FLT2   4               ;RESULT = 4
002322  026767  177770  175546 C11:    CMP     AN11,ANS1       ;CHECK LEFT HALF
002330  001401                  BEQ     .+4
002332  104002                  HLT+2                   ;LEFT HALF IS WRONG
002334  026767  177760  175536        CMP     AN11+2,ANS2     ;CHECK RIGHT HALF
002342  001401                  BEQ     .+4
002344  104002                  HLT+2                   ;RIGHT HALF IS WRONG

                        ;*****************************************************************
                        ;TEST 12                TEST OF MULF FPU INSTRUCTION
                        ;    5 * 1 = 5
                        ;    USING AC0    FPS = 47400    FEC = N/A
                        ;*****************************************************************
002346  104400                  SCOPE
002350  170127  047400          LDFPS   #47400&57760
002354  172467  000024          LDF     N12,0           ;LOAD 5 INTO AC12
002360  171067  000024          MULF    N12,0           ;MULTIPLY 5 BY 1
002364  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002370  022700  047400          CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002374  001401                  BEQ     .+4             ;BRANCH IF OK
002376  104000                  HLT                     ;FPS NOT EQUAL TO 47400

002400  174067  176474          STF     0,ANS1          ;STORE RESULT
002404  000406                  BR      C12
002406  040640  000000  N12:    .FLT2   5
002412  040200  000000  N12+2:  .FLT2   1
002416  040640  000000  ANS12:  .FLT2   5               ;RESULT = 5
002422  026767  177770  175450 C12:    CMP     AN12,ANS1       ;CHECK LEFT HALF
002430  001401                  BEQ     .+4
002432  104002                  HLT+2                   ;LEFT HALF IS WRONG
002434  026767  177760  175440        CMP     AN12+2,ANS2     ;CHECK RIGHT HALF
002442  001401                  BEQ     .+4
002444  104002                  HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 13                TEST OF MULF FPU INSTRUCTION
                                ;    6 * 1 = 6
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
002350  104400                          SCOPE
002352  170127  047400                  LDFPS   #47400857760
002356  172567  000024                  LDF     N13,1           ;LOAD 6 INTO AC13
002362  171167  000024                  MULF    M13,1           ;MULTIPLY 6 BY 1
002366  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002370  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002374  001401                          BEQ     .+4             ;BRANCH IF OK
002376  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002400  174167  176376                  STF     1,ANS1          ;STORE RESULT
002404  000406                          BR      D13
002406  040700  000000          N13:    .FLT2   6
002412  040200  000000          M13:    .FLT2   1
002416  040700  000000          AN13:   .FLT2   6               ;RESULT = 6
002422  026767  177770  176352  D13:    CMP     AN13,ANS1       ;CHECK LEFT HALF
002430  001401                          BEQ     .+4
002432  104002                          HLT+2                   ;LEFT HALF IS WRONG
002434  026767  177760  176342          CMP     AN13+2,ANS2     ;CHECK RIGHT HALF
002442  001401                          BEQ     .+4
002444  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 14                TEST OF MULF FPU INSTRUCTION
                                ;    7 * 1 = 7
                                ;    USING AC3    FPS = 47400    FEC = N/A
                                ;*****************************************************************
002446  104400                          SCOPE
002450  170127  047400                  LDFPS   #47400857760
002454  172767  000024                  LDF     N14,3           ;LOAD 7 INTO AC14
002460  171367  000024                  MULF    M14,3           ;MULTIPLY 7 BY 1
002464  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002466  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002472  001401                          BEQ     .+4             ;BRANCH IF OK
002474  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002476  174367  176300                  STF     3,ANS1          ;STORE RESULT
002502  000406                          BR      D14
002504  040740  000000          N14:    .FLT2   7
002510  040200  000000          M14:    .FLT2   1
002514  040740  000000          AN14:   .FLT2   7               ;RESULT = 7
002520  026767  177770  176254  D14:    CMP     AN14,ANS1       ;CHECK LEFT HALF
002526  001401                          BEQ     .+4
002530  104002                          HLT+2                   ;LEFT HALF IS WRONG
002532  026767  177760  176244          CMP     AN14+2,ANS2     ;CHECK RIGHT HALF
002540  001401                          BEQ     .+4
002542  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 15                TEST OF MULF FPU INSTRUCTION
                                ;    7 * 2 = 14.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
002544  104400                          SCOPE
002546  170127  047400                  LDFPS   #47400&57760
002552  172567  000024                  LDF     N15.1           ;LOAD 7 INTO AC15
002556  171167  000024                  MULF    M15.1           ;MULTIPLY 7 BY 2
002562  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002564  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002570  001401                          BEQ     .+4             ;BRANCH IF OK
002572  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002574  174167  176202                  STF     1,ANS1          ;STORE RESULT
002600  000406                          BR      .15
002602  040740  000000          N15:    .FLT2   7
002606  040400  000000          M15:    .FLT2   2
002612  041140  000000          AN15:   .FLT2   14.             ;RESULT = 14.
002616  026767  177770  176156  .15:    CMP     AN15,ANS1       ;CHECK LEFT HALF
002624  001401                          BEQ     .+4
002626  104002                          HLT+2                   ;LEFT HALF IS WRONG
002630  026767  177760  176146          CMP     AN15+2,ANS2     ;CHECK RIGHT HALF
002636  001401                          BEQ     .+4
002640  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 16                TEST OF MULF FPU INSTRUCTION
                                ;    7 * 3 = 21.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
002642  104400                          SCOPE
002644  170127  047400                  LDFPS   #47400&57760
002650  172567  000024                  LDF     N16.1           ;LOAD 7 INTO AC16
002654  171167  000024                  MULF    M16.1           ;MULTIPLY 7 BY 3
002660  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002662  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002666  001401                          BEQ     .+4             ;BRANCH IF OK
002670  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002672  174167  176134                  STF     1,ANS1          ;STORE RESULT
002676  000406                          BR      .16
002700  040740  000000          N16:    .FLT2   7
002704  040600  000000          M16:    .FLT2   3
002710  041250  000000          AN16:   .FLT2   21.             ;RESULT = 21.
002714  026767  177770  176060  .16:    CMP     AN16,ANS1       ;CHECK LEFT HALF
002722  001401                          BEQ     .+4
002724  104002                          HLT+2                   ;LEFT HALF IS WRONG
002726  026767  177760  176050          CMP     AN16+2,ANS2     ;CHECK RIGHT HALF
002734  001401                          BEQ     .+4
002736  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 17                TEST OF MULF FPU INSTRUCTION
                                ;    7 * 4 = 28.
                                ;    USING AC2    FPS = 47400    FEC = N/A
                                ;*****************************************************************
002740  104400                          SCOPE
002742  170127  047400                  LDFPS   #47400857760
002746  172667  000024                  LDF     N17.2           ;LOAD 7 INTO AC17
002752  171267  000024                  MULF    N17.2           ;MULTIPLY 7 BY 4
002756  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002760  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002764  001401                          BEQ     .+4             ;BRANCH IF OK
002766  104000                          HALT                    ;FPS NOT EQUAL TO 47400

002770  174267  176006                  STF     2,ANS:          ;STORE RESULT
002774  000406                          BR      .+14
002776  040740  000000          N17:    .FLT2   4
003002  040600  000000                  .FLT2   3
003006  041340  000000          ANS2:   .FLT2   28              ;RESULT = 28.
003012  026767  177770  175762  ANS:    CMP     ANS2,ANS:       ;CHECK LEFT HALF
003020  001401                          BEQ     .+4
003022  104002                          HALT+2                  ;LEFT HALF IS WRONG
003024  026767  177760  175752          CMP     ANS2+2,ANS2     ;CHECK RIGHT HALF
003032  001401                          BEQ     .+4
003034  104002                          HALT+2                  ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 20                TEST OF MULF FPU INSTRUCTION
                                ;    7 * 5 = 35.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003036  104400                          SCOPE
003040  170127  047400                  LDFPS   #47400857760
003044  172567  000024                  LDF     N20.1           ;LOAD 7 INTO AC20
003050  171167  000024                  MULF    N20.1           ;MULTIPLY 7 BY 5
003054  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003056  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003062  001401                          BEQ     .+4             ;BRANCH IF OK
003064  104000                          HALT                    ;FPS NOT EQUAL TO 47400

003066  174167  175710                  STF     1,ANS:          ;STORE RESULT
003072  000406                          BR      .+14
003074  040740  000000          N20:    .FLT2   4
003100  040640  000000                  .FLT2   5
003104  041414  000000          ANS20:  .FLT2   35              ;RESULT = 35.
003110  026767  177770  175664  ANS:    CMP     ANS20,ANS:      ;CHECK LEFT HALF
003116  001401                          BEQ     .+4
003120  104002                          HALT+2                  ;LEFT HALF IS WRONG
003122  026767  177760  175654          CMP     ANS20+2,ANS2    ;CHECK RIGHT HALF
003130  001401                          BEQ     .+4
003132  104002                          HALT+2                  ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 21                TEST OF MULF FPU INSTRUCTION
                        ;    7 * 6 = 42.
                        ;    USING AC3    FPS = 47400    FEC = N/A
                        ;*****************************************************************
003134  104400                    SCOPE
003136  170127  047400            LDFPS   #47400&57760
003142  172567  000024            MOVF    N21,3           ;LOAD 7 INTO AC3
003146  171167  000024            MULF    N21,3           ;MULTIPLY 7 BY 6
003152  170200                    STSTS   FPS             ;STORE FLOATING POINT STATUS
003154  022700  047400            CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003160  001401                    BEQ     .+4             ;BRANCH IF OK
003162  104000                    HLT                     ;FPS NOT EQUAL TO 47400

003164  174367  175612            MOVF    3,ANS1          ;STORE RESULT
003170  000406                    BR      .+10
003172  040740  000000    N21:    .BLKW   2
003176  040700  000000    ANS1:   .BLKW   2
003202  041450  000000    ANS2:   .BLKW   2
003206  026767  177770  175566    1$:   CMP     AN21,ANS1       ;CHECK LEFT HALF
003214  001401                    BEQ     .+4
003216  104002                    HLT+2                   ;LEFT HALF IS WRONG
003220  026767  177760  175556          CMP     AN21+2,ANS2     ;CHECK RIGHT HALF
003226  001401                    BEQ     .+4
003230  104002                    HLT+2                   ;RIGHT HALF IS WRONG

                        ;*****************************************************************
                        ;TEST 22                TEST OF MULF FPU INSTRUCTION
                        ;    7 * 7 = 49.
                        ;    USING AC1    FPS = 47400    FEC = N/A
                        ;*****************************************************************
003232  104400                    SCOPE
003234  170127  047400            LDFPS   #47400&57760
003240  172567  000024            MOVF    N22,1           ;LOAD 7 INTO AC1
003244  171167  000024            MULF    N22,1           ;MULTIPLY 7 BY 7
003250  170200                    STSTS   FPS             ;STORE FLOATING POINT STATUS
003252  022700  047400            CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003256  001401                    BEQ     .+4             ;BRANCH IF OK
003260  104000                    HLT                     ;FPS NOT EQUAL TO 47400

003262  174167  175514            MOVF    1,ANS1          ;STORE RESULT
003266  000406                    BR      .+10
003270  040740  000000    N22:    .BLKW   2
003274  040740  000000    ANS1:   .BLKW   2
003300  041534  000000    ANS2:   .BLKW   2
003304  026767  177770  175470    2$:   CMP     AN22,ANS1       ;CHECK LEFT HALF
003312  001401                    BEQ     .+4
003314  104002                    HLT+2                   ;LEFT HALF IS WRONG
003316  026767  177760  175460          CMP     AN22+2,ANS2     ;CHECK RIGHT HALF
003324  001401                    BEQ     .+4
003326  104002                    HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 23                TEST OF MULF FPU INSTRUCTION
                                ;    5 * 6 = 30.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003330  104400                          SCOPE
003332  170127  047400                  LDFPS   #47400&57760
003336  172567  000024                  LDF     N23.1           ;LOAD 5 INTO AC23
003342  171167  000024                  MULF    M23.1           ;MULTIPLY 5 BY 6
003346  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003350  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003354  001401                          BEQ     .+4             ;BRANCH IF OK
003356  104000                          HLT                     ;FPS NOT EQUAL TO 47400

003360  174167  175416                  STF     1,ANS1          ;STORE RESULT
003364  000406                          BR      C23
003366  040640  000000          N23:    .FLT2   5
003372  040700  000000          M23:    .FLT2   6
003376  041360  000000          AN23:   .FLT2   30.             ;RESULT = 30.
003402  026767  177770  175372  C23:    CMP     AN23,ANS1       ;CHECK LEFT HALF
003410  001401                          BEQ     .+4
003412  104002                          HLT+2                   ;LEFT HALF IS WRONG
003414  026767  177760  175362          CMP     AN23+2,ANS2     ;CHECK RIGHT HALF
003422  001401                          BEQ     .+4
003424  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 24                TEST OF MULF FPU INSTRUCTION
                                ;    4 * 6 = 24.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003426  104400                          SCOPE
003430  170127  047400                  LDFPS   #47400&57760
003434  172567  000024                  LDF     N24.1           ;LOAD 4 INTO AC24
003440  171167  000024                  MULF    M24.1           ;MULTIPLY 4 BY 6
003444  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003446  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003452  001401                          BEQ     .+4             ;BRANCH IF OK
003454  104000                          HLT                     ;FPS NOT EQUAL TO 47400

003456  174167  175320                  STF     1,ANS1          ;STORE RESULT
003462  000406                          BR      C24
003464  040600  000000          N24:    .FLT2   4
003470  040700  000000          M24:    .FLT2   6
003474  041300  000000          AN24:   .FLT2   24.             ;RESULT = 24.
003500  026767  177770  175274  C24:    CMP     AN24,ANS1       ;CHECK LEFT HALF
003506  001401                          BEQ     .+4
003510  104002                          HLT+2                   ;LEFT HALF IS WRONG
003512  026767  177760  175264          CMP     AN24+2,ANS2     ;CHECK RIGHT HALF
003520  001401                          BEQ     .+4
003522  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 25                TEST OF MULF FPU INSTRUCTION
                                ;    3 * 6 = 18.
                                ;    USING AC2    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003524  104400                          SCOPE
003526  170127  047400                  LDFPS   #47400&57760
003532  172667  000024                  LDF     N25,2           ;LOAD 3 INTO AC25
003536  171267  000024                  MULF    M25,2           ;MULTIPLY 3 BY 6
003542  170300                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003544  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003550  001401                          BEQ     .+4             ;BRANCH IF OK
003552  104000                          HLT                     ;FPS NOT EQUAL TO 47400

003554  174267  175222                  STF     3,ANS1          ;STORE RESULT
003560  000406                          BR      .25
003562  040500  000000          N25:    .FLT2   3
003566  040700  000000          M25:    .FLT2   6
003572  041220  000000          AN25:   .FLT2   18.             ;RESULT = 18.
003576  026767  177770  175176  025:    CMP     AN25,ANS1       ;CHECK LEFT HALF
003604  001401                          BEQ     .+4
003606  104002                          HLT+2                   ;LEFT HALF IS WRONG
003610  026767  177760  175166          CMP     AN25+2,ANS2     ;CHECK RIGHT HALF
003616  001401                          BEQ     .+4
003620  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 26                TEST OF MULF FPU INSTRUCTION
                                ;    6 * 2 = 12.
                                ;    USING AC3    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003622  104400                          SCOPE
003624  170127  047400                  LDFPS   #47400&57760
003630  172767  000024                  LDF     N26,3           ;LOAD 6 INTO AC26
003634  171367  000024                  MULF    M26,3           ;MULTIPLY 6 BY 2
003640  170300                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003642  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003646  001401                          BEQ     .+4             ;BRANCH IF OK
003650  104000                          HLT                     ;FPS NOT EQUAL TO 47400

003652  174367  175124                  STF     3,ANS1          ;STORE RESULT
003656  000406                          BR      .26
003660  040700  000000          N26:    .FLT2   0
003664  040400  000000          M26:    .FLT2   2
003670  041100  000000          AN26:   .FLT2   12.             ;RESULT = 12.
003674  026767  177770  175100  026:    CMP     AN26,ANS1       ;CHECK LEFT HALF
003702  001401                          BEQ     .+4
003704  104002                          HLT+2                   ;LEFT HALF IS WRONG
003706  026767  177760  175070          CMP     AN26+2,ANS2     ;CHECK RIGHT HALF
003714  001401                          BEQ     .+4
003716  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 27                TEST OF MULF FPU INSTRUCTION
                                ;    5 * 2 = 10.
                                ;    USING AC3    FPS = 47400    FEC = N/A
                                ;*****************************************************************
003736  104400                          SCOPE
003740  170127  047400                  LDFPS   #47400&57760
003744  172667  000024                  LDF     N27,3           ;LOAD 5 INTO AC27
003750  171367  000024                  MULF    M27,3           ;MULTIPLY 5 BY 2
003754  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
003756  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
003762  001401                          BEQ     .+4             ;BRANCH IF OK
003764  104000                          HLT                     ;FPS NOT EQUAL TO 47400

003766  174367  175026                  STF     3,ANS1          ;STORE RESULT
003772  000406                          BR      .+2
003774  040640  000000          N27:    .FLT2   5
004000  040400  000000          M27:    .FLT2   2
004004  041040  000000          ANS27:  .FLT2   10.             ;RESULT = 10.
004010  026767  177770  175002  027:    CMP     ANS27,ANS.      ;CHECK LEFT HALF
004016  001401                          BEQ     .+4
004020  104002                          HLT+2                   ;LEFT HALF IS WRONG
004022  026767  177750  175012          CMP     ANS27+2,ANS2    ;CHECK RIGHT HALF
004030  001401                          BEQ     .+4
004032  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 30                TEST OF MULF FPU INSTRUCTION
                                ;    5 * 3 = 15.
                                ;    USING AC2    FPS = 47400    FEC = N/A
                                ;*****************************************************************
004034  104400                          SCOPE
004036  170127  047400                  LDFPS   #47400&57760
004042  172667  000024                  LDF     N30,2           ;LOAD 5 INTO AC30
004046  171267  000024                  MULF    M30,2           ;MULTIPLY 5 BY 3
004052  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004054  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
004060  001401                          BEQ     .+4             ;BRANCH IF OK
004062  104000                          HLT                     ;FPS NOT EQUAL TO 47400

004064  174267  174730                  STF     2,ANS1          ;STORE RESULT
004070  000406                          BR      .+2
004072  040640  000000          N30:    .FLT2   5
004076  040500  000000          M30:    .FLT2   3
004102  041140  000000          ANS30:  .FLT2   15.             ;RESULT = 15.
004106  026767  177750  174730  030:    CMP     ANS30,ANS1      ;CHECK LEFT HALF
004114  001401                          BEQ     .+4
004116  104002                          HLT+2                   ;LEFT HALF IS WRONG
004120  026767  177730  174742          CMP     ANS30+2,ANS2    ;CHECK RIGHT HALF
004126  001401                          BEQ     .+4
004130  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 31                 TEST OF MULF FPU INSTRUCTION
                                ;    5 * 4 = 20.
                                ;    USING AC1    FPS = 47400    FEC = N/A
                                ;*****************************************************************
                                        SCOPE
                                        LDFPS   @47400(57760
                                        LDD     N31.1           ;LOAD 5 INTO AC3:
                                        MULF    N31.1           ;MULTIPLY 5 BY 4
                                        STFPS   FPS             ;STORE FLOATING POINT STATUS
                                        CMP     @47400,FPS      ;CHECK FLOATING POINT STATUS
                                        BEQ     .+4             ;BRANCH IF OK
                                        HLT                     ;FPS NOT EQUAL TO 47400

                                        STF     1,ANS1          ;STORE RESULT
                                        SUB     0,0
                                        MOV     0,1
                                        MOV     0,2
                                        MOV     0,3
                                        MOV     0,FEC
                                        CMP     1,ANS1+2        ;RESULT = 20.
                                        BEQ     .+4
                                        HLT                     ;EXT HALF IS WRONG
                                        CMP     2,ANS1+3        ;CHECK FRACTION HALF
                                        BEQ     .+4
                                        HLT                     ;FRACT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 32                TEST OF MULD FPU INSTRUCTION
                        ;    0 * 0 = 0
                        ;    USING AC0    FPS = 47604    FEC = N/A
                        ;*****************************************************************

004212  104400                          SCOPE
004214  170127  047600                  LDFPS   #47604&57760
004220  172467  000012                  LDD     N32,0           ;LOAD 0 INTO AC0
004224  171067  000016                  MULD    M32,0           ;MULTIPLY 0 BY 0
004230  174067  174546                  STD     0,ANS1          ;STORE RESULT
004234  000414                          BR      C32

004236  000000  000000  000000  N32:    .FLT4   0
004244  000000
004246  000000  000000  000000  M32:    .FLT4   0
004254  000000
004256  000000  000000  000000  AN32:   .FLT4   0
004264  000000

004266  026767  177764  174506  C32:    CMP     AN32,ANS1       ;CHECK LEFT HALF
004274  001401                          BEQ     .+4
004276  104004                          HLT+4                   ;LEFT HALF IS WRONG

004300  026767  177754  174476          CMP     AN32+2,ANS2     ;CHECK LEFT HALF
004306  001401                          BEQ     .+4
004310  104004                          HLT+4                   ;LEFT HALF IS WRONG

004312  026767  177744  174466          CMP     AN32+4,ANS3     ;CHECK RIGHT HALF
004320  001401                          BEQ     .+4
004322  104004                          HLT+4                   ;RIGHT HALF IS WRONG

004324  026767  177734  174456          CMP     AN32+6,ANS4     ;CHECK RIGHT HALF
004332  001401                          BEQ     .+4
004334  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 33                TEST OF MULD FPU INSTRUCTION
                        ;    0 * 1. = 0
                        ;    USING AC0    FPS = 47604    FEC = N/A
                        ;*****************************************************************

004336  104400                          SCOPE
004340  170127  047600                  LDFPS   #47604&57760
004344  172467  000012                  LDD     N33,0           ;LOAD 0 INTO AC0
004350  171067  000016                  MULD    M33,0           ;MULTIPLY 0 BY 1.
004354  174067  174422                  STD     0,ANS1          ;STORE RESULT
004360  000414                          BR      033

004362  000000  000000  000000  N33:    .FLT4   0
004370  000000
004372  040200  000000  000000  M33:    .FLT4   1.
004400  000000
004402  000000  000000  000000  AN33:   .FLT4   0
004410  000000

004412  026767  177764  174362  033:    CMP     AN33,ANS1       ;CHECK LEFT HALF
004420  001401                          BEQ     .+4
004422  104004                          HLT+4                   ;LEFT HALF IS WRONG

004424  026767  177754  174352          CMP     AN33+2,ANS2     ;CHECK LEFT HALF
004432  001401                          BEQ     .+4
004434  104004                          HLT+4                   ;LEFT HALF IS WRONG

004436  026767  177744  174342          CMP     AN33+4,ANS3     ;CHECK RIGHT HALF
004444  001401                          BEQ     .+4
004446  104004                          HLT+4                   ;RIGHT HALF IS WRONG

004450  026767  177734  174332          CMP     AN33+6,ANS4     ;CHECK RIGHT HALF
004456  001401                          BEQ     .+4
004460  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 34                TEST OF MULD FPU INSTRUCTION
                        ;    1. * 1. = 1.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

004462  104400                          SCOPE
004464  170127  047600                  LDFPS   #47600&57760
004470  172467  000012                  LDD     N34,0           ;LOAD 1. INTO ACC
004474  171067  000016                  MULD    M34,0           ;MULTIPLY 1. BY 1.
004500  174067  174276                  STD     0,ANS1          ;STORE RESULT
004504  000414                          BR      .34

004506  040200  000000  000000  N34:    .FLT4   1.
004514  000000
004516  040200  000000  000000  M34:    .FLT4   1.
004524  000000
004526  040200  000000  000000  AN34:   .FLT4   1.
004534  000000

004536  026767  177764  174236  .34:    CMP     AN34,ANS1       ;CHECK LEFT HALF
004544  001401                          BEQ     .+4
004546  104004                          HLT+4                   ;LEFT HALF IS WRONG

004550  026767  177754  174226          CMP     AN34+2,ANS2     ;CHECK LEFT HALF
004556  001401                          BEQ     .+4
004560  104004                          HLT+4                   ;LEFT HALF IS WRONG

004562  026767  177744  174216          CMP     AN34+4,ANS3     ;CHECK RIGHT HALF
004570  001401                          BEQ     .+4
004572  104004                          HLT+4                   ;RIGHT HALF IS WRONG

004574  026767  177734  174206          CMP     AN34+6,ANS4     ;CHECK RIGHT HALF
004602  001401                          BEQ     .+4
004604  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
;*****************************************************************
;TEST 35                 TEST OF MULD FPU INSTRUCTION
;    -1. * 1. = -1.
;    USING AC0    FPS = 47610    FEC = N/A
;*****************************************************************

004626  104420            SCOPE
004630  000414            BR      C35

004632  140200  000000  000000  N35:  .FLT4  -1.
004636  000000
004640  040200  000000  000000  M35:  .FLT4  1.
004644  000000
004650  140200  000000  000000  AN35: .FLT4  -1.
004654  000000

004660  126767  177764  174112  C35:  CMP     AN35,ANS1       ;CHECK LEFT HALF
004664  001401                      BEQ     .+4
004666  104034                      ERROR                   ;LEFT HALF IS WRONG

004670  126767  177754  174102        CMP     AN35+2,ANS2     ;CHECK LEFT HALF
004674  001401                      BEQ     .+4
004676  104034                      ERROR                   ;LEFT HALF IS WRONG

004700  126767  177744  174072        CMP     AN35+4,ANS3     ;CHECK RIGHT HALF
004704  001401                      BEQ     .+4
004706  104034                      ERROR                   ;RIGHT HALF .5 WRONG

004710  126767  177734  174062        CMP     AN35+6,ANS4     ;CHECK RIGHT HALF
004714  001401                      BEQ     .+4
004716  104034                      ERROR                   ;RIGHT HALF IS WRONG
```

```
;*****************************************************************
;TEST 36                 TEST OF MULD FPU INSTRUCTION
;    1. * -1. = -1.
;    USING ACO     FPS = 47610     FEC = N/A
;*****************************************************************

[illegible]  [illegible]          SCOPE
[illegible]  [illegible]  047600  LDFPS   #47610357760
[illegible]  [illegible]  000012  LDD     N36,0              ;LOAD 1. INTO ACO
[illegible]  [illegible]  000C16  MULD    M36,0              ;MULTIPLY BY -1.
[illegible]  [illegible]  174026  STD     0,ANS1             ;STORE RESULT
[illegible]  000414               BR      C36

034756  040200  000000  000000  N36:    .FLT4   1.
034764  000000
034766  140200  000000  000000  M36:    .FLT4   -1.
034774  000000
034776  140200  000000  000000  AN36:   .FLT4   -1.
035004  000000

035006  026767  177764  173766  C36:    CMP     AN36,ANS1          ;CHECK LEFT HALF
035014  001401                          BEQ     .+4
035016  104004                          HLT+4                      ;LEFT HALF IS WRONG

035020  026767  177754  173756          CMP     AN36+2,ANS2        ;CHECK LEFT HALF
035026  001401                          BEQ     .+4
035030  104004                          HLT+4                      ;LEFT HALF IS WRONG

035032  026767  177744  173746          CMP     AN36+4,ANS3        ;CHECK RIGHT HALF
035040  001401                          BEQ     .+4
035042  104004                          HLT+4                      ;RIGHT HALF IS WRONG

035044  026767  177734  173736          CMP     AN36+6,ANS4        ;CHECK RIGHT HALF
035052  001401                          BEQ     .+4
035054  104004                          HLT+4                      ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 37                TEST OF MULD FPU INSTRUCTION
                        ;    -1. * -1. = 1.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

005056  104430                    SCOPE
005060  170127  047600            LDFPS   #47600857760
005064  172467  000012            LDD     N37,0           ;LOAD -1. INTO AC0
005070  171067  000016            MULD    M37,0           ;MULTIPLY -1. BY -1.
005074  174067  173702            STD     0,ANS1          ;STORE RESULT
005100  000414                    BR      .+10

005102  140200  000000  000000  N37:    .FLT4   -1.
005110  000000
005112  140200  000000  000000  M37:    .FLT4   -1.
005120  000000
005122  040200  000000  000000  AN37:   .FLT4   1.
005130  000000

005132  026767  177764  173642  .+10:   CMP     AN37,ANS1       ;CHECK LEFT HALF
005140  001401                  BEQ     .+4
005142  104004                  HLT+4                   ;LEFT HALF IS WRONG

005144  026767  177754  173632          CMP     AN37+2,ANS2     ;CHECK LEFT HALF
005152  001401                  BEQ     .+4
005154  104004                  HLT+4                   ;LEFT HALF IS WRONG

005156  026767  177744  173622          CMP     AN37+4,ANS3     ;CHECK RIGHT HALF
005164  001401                  BEQ     .+4
005166  104004                  HLT+4                   ;RIGHT HALF IS WRONG

005170  026767  177734  173612          CMP     AN37+6,ANS4     ;CHECK RIGHT HALF
005176  001401                  BEQ     .+4
005200  104004                  HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 40                TEST OF MULD FPU INSTRUCTION
                        ;    2 * 1 = 2
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

005202  104400                  SCOPE
005204  170127  047600          LDFPS   #47600&57760
005210  172467  000012          LDD     N40,0           ;LOAD 2 INTO AC0
005214  171067  000016          MULD    M40,0           ;MULTIPLY 2 BY 1
005220  174067  173556          STD     0,ANS1          ;STORE RESULT
005224  000414                  BR      040

005226  040400  000000  000000  N40:    .FLT4   2
005234  000000
005236  040200  000000  000000  M40:    .FLT4   1
005244  000000
005246  040400  000000  000000  AN40:   .FLT4   2
005254  000000

005256  026767  177764  173516  040:    CMP     AN40,ANS1       ;CHECK LEFT HALF
005264  001401                   BEQ     .+4
005266  104004                   MLT+4                   ;LEFT HALF IS WRONG

005270  026767  177754  173506          CMP     AN40+2,ANS2     ;CHECK LEFT HALF
005276  001401                          BEQ     .+4
005300  104004                          MLT+4                   ;LEFT HALF IS WRONG

005302  026767  177744  173476          CMP     AN40+4,ANS3     ;CHECK RIGHT HALF
005310  001401                          BEQ     .+4
005312  104004                          MLT+4                   ;RIGHT HALF IS WRONG

005314  026767  177734  173466          CMP     AN40+6,ANS4     ;CHECK RIGHT HALF
005322  001401                          BEQ     .+4
005324  104004                          MLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 41                TEST OF MULD FPU INSTRUCTION
                        ;    3 * 1 = 3
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

[illegible]  124400                          SCOPE
[illegible]  170120  047600                  LDFPS   #47600&57760
[illegible]  172467  000012                  LDD     N41,0           ;LOAD 3 INTO AC0
[illegible]  171067  000010                  MULD    M41,0           ;MULTIPLY 3 BY 1
[illegible]  174067  173432                  STD     0,ANS1          ;STORE RESULT
[illegible]  000414                          BR      .+41

[illegible]  040530  000000  000000  N41:    .FLT4   3
[illegible]  000000
[illegible]  040200  000000  000000  M41:    .FLT4   1
[illegible]  000000
[illegible]  040500  000000  000000  ANS1:   .FLT4   3
[illegible]  000000

[illegible]  026767  177764  173372  041:    CMP     ANS1,ANS1       ;CHECK LEFT HALF
[illegible]  001400                          BEQ     .+4
[illegible]  104004                          HLT+4                   ;LEFT HALF IS WRONG

[illegible]  026767  177754  173362          CMP     ANS1+2,ANS2     ;CHECK LEFT HALF
[illegible]  001400                          BEQ     .+4
[illegible]  104004                          HLT+4                   ;LEFT HALF IS WRONG

[illegible]  026767  177744  173352          CMP     ANS1+4,ANS3     ;CHECK RIGHT HALF
[illegible]  001400                          BEQ     .+4
[illegible]  104004                          HLT+4                   ;RIGHT HALF IS WRONG

[illegible]  026767  177734  173342          CMP     ANS1+6,ANS4     ;CHECK RIGHT HALF
[illegible]  001400                          BEQ     .+4
[illegible]  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
;*****************************************************************
;TEST 42                  TEST OF MULD FPU INSTRUCTION
;    4 * 1 = 4
;    USING AC0    FPS = 47600    FEC = N/A
;*****************************************************************

[illegible]  047600                      SCOPE
[illegible]  000001                      LDFPS   #47600857760
[illegible]  000041                      LDD     #N42,0          ;LOAD 4 INTO AC0
[illegible]  173306                      MULD    #N42,0          ;MULTIPLY 4 BY 1
[illegible]  000414                      BR      .+ANS:          ;SCOPE BESIDE
                                         BR      .+42

[illegible]  040600  000000  000000  N42:   .FLT4    4
[illegible]  000000
[illegible]  040200  000000  000000  N42:   .FLT4    1
[illegible]  000000
[illegible]  040600  000000  000000  ANS:   .FLT4    4
[illegible]  000000

[illegible]  026767  177764  173246  042:   CMP     ANS,ANS:        ;CHECK LEFT HALF
[illegible]  001402                      BEQ     .+4
[illegible]  104004                      HLT     .+4             ;LEFT HALF IS WRONG

[illegible]  026767  177754  173236          CMP     ANS+2,ANS2      ;CHECK LEFT HALF
[illegible]  001402                      BEQ     .+4
[illegible]  104004                      HLT     .+4             ;LEFT HALF IS WRONG

[illegible]  026767  177744  173226          CMP     ANS+4,ANS3      ;CHECK RIGHT HALF
[illegible]  001402                      BEQ     .+4
[illegible]  104004                      HLT     .+4             ;RIGHT HALF IS WRONG

[illegible]  026767  177734  173216          CMP     ANS+6,ANS+      ;CHECK RIGHT HALF
[illegible]  001402                      BEQ     .+4
[illegible]  104004                      HLT     .+4             ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 43                TEST OF MULD FPU INSTRUCTION
                        ;    5 * 1 = 5
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

005576  104400                        SCOPE
005600  170127  047600                LDFPS   #47600&57760
005604  172467  000012                LDD     N43,0           ;LOAD 5 INTO AC0
005610  171067  000016                MULD    M43,0           ;MULTIPLY 5 BY 1
005614  174067  173162                STD     0,ANS1          ;STORE RESULT
005620  000414                        BR      .43

005622  040640  000000  000000  N43:  .FLT4   5
005630  000000
005632  040200  000000  000200  M43:  .FLT4   1
005640  000000
005642  040640  000000  000000  AN43: .FLT4   5
005650  000000

005652  026767  177764  173122  .43:  CMP     AN43,ANS1       ;CHECK LEFT HALF
005660  001401                        BEQ     .+4
005662  104004                        HLT+4                   ;LEFT HALF IS WRONG

005664  026767  177754  173112        CMP     AN43-2,ANS2     ;CHECK LEFT HALF
005672  001401                        BEQ     .+4
005674  104004                        HLT+4                   ;LEFT HALF IS WRONG

005676  026767  177744  173102        CMP     AN43+4,ANS3     ;CHECK RIGHT HALF
005704  001401                        BEQ     .+4
005706  104004                        HLT+4                   ;RIGHT HALF IS WRONG

005710  026767  177734  173072        CMP     AN43+6,ANS4     ;CHECK RIGHT HALF
005716  001401                        BEQ     .+4
005720  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 44                TEST OF MULD FPU INSTRUCTION
                        ;    6 * 1 = 6
                        ;    USING AC1    FPS = 47600    FEC = N/A
                        ;*****************************************************************

005732  104400                    SCOPE
005734  170127  047600            LDFPS   #47600
005736  172567  000012            LDD     N44,1           ;LOAD 6 INTO AC1
005742  171167  000016            MULD    M44,1           ;MULTIPLY 6 BY 1
005746  174167  173036            STD     1,ANS1          ;STORE RESULT
005752  000414                    BR      C44

005754  040700  000000  000000  N44:    .FLT4   6
005762  000000
005764  040200  000000  000000  M44:    .FLT4   1
005772  000000
005774  040700  000000  000000  AN44:   .FLT4   6
006002  000000

006004  026767  177764  172776  C44:    CMP     AN44,ANS1       ;CHECK LEFT HALF
006012  001401                    BEQ     .+4
006014  104004                    HLT+4                   ;LEFT HALF IS WRONG

006016  026767  177754  172766          CMP     AN44+2,ANS2     ;CHECK LEFT HALF
006024  001401                    BEQ     .+4
006026  104004                    HLT+4                   ;LEFT HALF IS WRONG

006030  026767  177744  172756          CMP     AN44+4,ANS3     ;CHECK RIGHT HALF
006036  001401                    BEQ     .+4
006040  104004                    HLT+4                   ;RIGHT HALF IS WRONG

006042  026767  177734  172746          CMP     AN44+6,ANS4     ;CHECK RIGHT HALF
006050  001401                    BEQ     .+4
006052  104004                    HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 45                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 1 = 7
                        ;    USING AC3    FPS = 47600    FEC = N/A
                        ;*****************************************************************

006046  104400                          SCOPE
006050  170127  047600                  LDFPS   #47600&57760
006054  172467  000012                  LDD     N45,3           ;LOAD 7 INTO AC3
006060  171367  000016                  MULD    M45,3           ;MULTIPLY 7 BY 1
006064  174367  172712                  STD     3,ANS1          ;STORE RESULT
006070  000414                          BR      C45

006072  040740  000000  000000  N45:    .FLT4   7
006100  000000
006102  040200  000000  000000  M45:    .FLT4   1
006110  000000
006112  040740  000000  000000  AN45:   .FLT4   7
006120  000000

006122  026767  177764  172652  C45:    CMP     AN45,ANS1       ;CHECK LEFT HALF
006130  001401                          BEQ     .+4
006132  104004                          HLT+4                   ;LEFT HALF IS WRONG

006134  026767  177754  172642          CMP     AN45+2,ANS2     ;CHECK LEFT HALF
006142  001401                          BEQ     .+4
006144  104004                          HLT+4                   ;LEFT HALF IS WRONG

006146  026767  177744  172632          CMP     AN45+4,ANS3     ;CHECK RIGHT HALF
006154  001401                          BEQ     .+4
006156  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006160  026767  177734  172622          CMP     AN45+6,ANS4     ;CHECK RIGHT HALF
006166  001401                          BEQ     .+4
006170  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 46                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 2 = 14.
                        ;    USING AC2    FPS = 47600    FEC = N/A
                        ;*****************************************************************

006172  104400                    SCOPE
006174  170267  047600            LDFPS   #47600&57760
006200  172667  000012            LDD     N46,2                   ;LOAD 7 INTO AC2
006204  171267  000016            MULD    N46,2                   ;MULTIPLY 7 BY 2
006210  174267  172566            STD     2,ANS1                  ;STORE RESULT
006214  000414                    BR      .+46

006216  040740  000000  000000  N46:    .FLT4   7
006224  000000
006226  040400  000000  000000  M46:    .FLT4   2
006234  000000
006236  041160  000000  000000  AN46:   .FLT4   14.
006244  000000

006246  026767  177764  172526  .+46:   CMP     AN46,ANS1               ;CHECK LEFT HALF
006254  001401                          BEQ     .+4
006256  104004                          HLT+4                           ;LEFT HALF IS WRONG

006260  026767  177754  172516          CMP     AN46+2,ANS2             ;CHECK LEFT HALF
006266  001401                          BEQ     .+4
006270  104004                          HLT+4                           ;LEFT HALF IS WRONG

006272  026767  177744  172506          CMP     AN46+4,ANS3             ;CHECK RIGHT HALF
006300  001401                          BEQ     .+4
006302  104004                          HLT+4                           ;RIGHT HALF IS WRONG

006304  026767  177734  172476          CMP     AN46+6,ANS4             ;CHECK RIGHT HALF
006312  001401                          BEQ     .+4
006314  104004                          HLT+4                           ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 47                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 3 = 21.
                        ;    USING AC2    FPS = 47600    FEC = N/A
                        ;*****************************************************************

006316  104400                        SCOPE
006320  170127  047600                LDFPS   #47600&57760
006324  172667  000012                LDD     N47,F2          ;LOAD 7 INTO AC2
006330  171267  000016                MULD    M47,F2          ;MULTIPLY 7 BY 3
006334  174267  172442                STD     F2,ANS1         ;STORE RESULT
006340  000414                        BR      047

006342  040740  000000  000000  N47:  .FLT4   7
006350  000000
006352  040500  000000  000000  M47:  .FLT4   3
006360  000000
006362  041250  000000  000000  AN47: .FLT4   21.
006370  000000

006372  026767  177764  172402  047:  CMP     AN47,ANS1       ;CHECK LEFT HALF
006400  001401                        BEQ     .+4
006402  104004                        HLT+4                   ;LEFT HALF IS WRONG

006404  026767  177754  172372        CMP     AN47+2,ANS2     ;CHECK LEFT HALF
006412  001401                        BEQ     .+4
006414  104004                        HLT+4                   ;LEFT HALF IS WRONG

006416  026767  177744  172362        CMP     AN47+4,ANS3     ;CHECK RIGHT HALF
006424  001401                        BEQ     .+4
006426  104004                        HLT+4                   ;RIGHT HALF IS WRONG

006430  026767  177734  172352        CMP     AN47+6,ANS4     ;CHECK RIGHT HALF
006436  001401                        BEQ     .+4
006440  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 50                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 4 = 28.
                        ;    USING AC3    FPS = 47600    FEC = N/A
                        ;*****************************************************************

006442  104400                          SCOPE
006444  170127  047600                  LDFPS   #47600&57760
006450  172767  000012                  LDD     N50,3           ;LOAD 7 INTO AC3
006454  171367  000016                  MULD    M50,3           ;MULTIPLY 7 BY 4
006460  174367  172316                  STD     3,ANS1          ;STORE RESULT
006464  000414                          BR      050

006466  040740  000000  000000  N50:    .FLT4   7
006474  000000
006476  040600  000000  000000  M50:    .FLT4   4
006504  000000
006506  041340  000000  000000  AN50:   .FLT4   28.
006514  000000

006516  026767  177764  172256  050:    CMP     AN50,ANS1       ;CHECK LEFT HALF
006524  001401                          BEQ     .+4
006526  104004                          HLT+4                   ;LEFT HALF IS WRONG

006530  026767  177754  172246          CMP     AN50+2,ANS2     ;CHECK LEFT HALF
006536  001401                          BEQ     .+4
006540  104004                          HLT+4                   ;LEFT HALF IS WRONG

006542  026767  177744  172236          CMP     AN50+4,ANS3     ;CHECK RIGHT HALF
006550  001401                          BEQ     .+4
006552  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006554  026767  177734  172226          CMP     AN50+6,ANS4     ;CHECK RIGHT HALF
006562  001401                          BEQ     .+4
006564  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 51                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 5 = 35.
                        ;    USING AC1    FPS = 47600    FEC = N/A
                        ;*****************************************************************

006566  104400                        SCOPE
006570  170127  047600                LDFPS   #47600&57760
006574  172567  000012                LDD     N51,1           ;LOAD 7 INTO AC1
006600  171167  000016                MULD    M51,1           ;MULTIPLY 7 BY 5
006604  174167  172172                STD     1,ANS1          ;STORE RESULT
006610  000414                        BR      C51

006612  040740  000000  000000  N51:  .FLT4   7
006620  000000
006622  040640  000000  000000  M51:  .FLT4   5
006630  000000
006632  041414  000000  000000  ANS1: .FLT4   35.
006640  000000

006642  026767  177764  172132  C51:  CMP     ANS1,ANS1       ;CHECK LEFT HALF
006650  001401                        BEQ     .+4
006652  104004                        HLT+4                   ;LEFT HALF IS WRONG

006654  026767  177754  172122        CMP     ANS1+2,ANS2     ;CHECK LEFT HALF
006662  001401                        BEQ     .+4
006664  104004                        HLT+4                   ;LEFT HALF IS WRONG

006666  026767  177744  172112        CMP     ANS1+4,ANS3     ;CHECK RIGHT HALF
006674  001401                        BEQ     .+4
006676  104004                        HLT+4                   ;RIGHT HALF IS WRONG

006700  026767  177734  172102        CMP     ANS1+6,ANS4     ;CHECK RIGHT HALF
006706  001401                        BEQ     .+4
006710  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 52                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 6 = 42.
                        ;    USING AC1    FPS = 47600    FEC = N/A
                        ;*****************************************************************

[illegible]  104400                          SCOPE
[illegible]  104255                          LDFPS   #47600857760
[illegible]  047600  000012                  MOV     NS2,1           ;LOAD 7 INTO AC1
[illegible]  172046  000016                  MULD    MS2,1           ;MULTIPLY 7 BY 6
[illegible]  074167  172046                  STD     1,DNS1          ;STORE RESULT
000734  000414                              BR      .+52

000736  040740  000000  000000  NS2:    .FLT4   7
000744  000000
000746  040700  000000  000000  MS2:    .FLT4   6
000754  000000
000756  041450  000000  000000  ANS2:   .FLT4   42.
000764  000000

000766  026767  177764  172006  052:    CMP     ANS2,ANS1       ;CHECK LEFT HALF
000774  001401                           BEQ     .+4
000776  104004                           HLT+4                   ;LEFT HALF IS WRONG

001000  026767  177754  171776           CMP     ANS2+2,ANS2     ;CHECK LEFT HALF
001006  001401                           BEQ     .+4
001010  104004                           HLT+4                   ;LEFT HALF IS WRONG

001012  026767  177744  171766           CMP     ANS2+4,ANS3     ;CHECK RIGHT HALF
001020  001401                           BEQ     .+4
001022  104004                           HLT+4                   ;RIGHT HALF IS WRONG

001024  026767  177734  171756           CMP     ANS2+6,ANS4     ;CHECK RIGHT HALF
[illegible]  001401                          BEQ     .+4
[illegible]  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 53                TEST OF MULD FPU INSTRUCTION
                        ;    7 * 7 = 49.
                        ;    USING AC3    FPS = 47600    FEC = N/A
                        ;*****************************************************************

                                                SCOPE
                                                LOFPS   #47600&177760
                                                LDD     #53.3           ;LOAD 7 INTO AC3
                                                MULD    #53.3           ;MULTIPLY 7 BY
                                                STD     3,ANS1          ;STORE RESULT
                                                BR      .+53

000706  040740  000000  000000  MS3:    .FLT4   7
000710  000000
000712  040740  000000  000000  M53:    .FLT4   7
000720  000000
000722  041504  000000  000000  ANS3:   .FLT4   49.
000730  000000

000732  026767  177764  171662  053:    CMP     ANS3,ANS1       ;CHECK LEFT HALF
000740  001401                          BEQ     .+4
000742  104004                          HLT     4               ;LEFT HALF IS WRONG

000744  026767  177754  171652          CMP     ANS3+2,ANS2     ;CHECK LEFT HALF
000752  001401                          BEQ     .+4
000754  104004                          HLT     4               ;LEFT HALF IS WRONG

000756  026767  177744  171642          CMP     ANS3+4,ANS3     ;CHECK RIGHT HALF
000764  001401                          BEQ     .+4
000766  104004                          HLT     4               ;RIGHT HALF IS WRONG

000770  026767  177734  171632          CMP     ANS3+6,ANS4     ;CHECK RIGHT HALF
000776  001401                          BEQ     .+4
001000  104004                          HLT     4               ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 54                TEST OF MULD FPU INSTRUCTION
                        ;    5 * 6 = 30.
                        ;    USING AC1    FPS = 47600    FEC = N/A
                        ;*****************************************************************

007162  104400                    SCOPE
007164  170127  047600            LDFPS   #47600&57760
007170  172567  000012            LDD     N54,I           ;LOAD 5 INTO AC1
007174  171167  000016            MULD    M54,I           ;MULTIPLY 5 BY 6
007200  174167  171576            STD     I,ANS1          ;STORE RESULT
007204  000414                    BR      C54

007206  040640  000000  000000  N54:    .FLT4   5
007214  000000
007216  040700  000000  000000  M54:    .FLT4   6
007224  000000
007226  041360  000000  000000  AN54:   .FLT4   30.
007234  000000

007236  026767  177764  171536  C54:    CMP     AN54,ANS1       ;CHECK LEFT HALF
007244  001401                  BEQ     .+4
007246  104004                  HLT+4                   ;LEFT HALF IS WRONG

007250  026767  177754  171526          CMP     AN54+2,ANS2     ;CHECK LEFT HALF
007256  001401                  BEQ     .+4
007260  104004                  HLT+4                   ;LEFT HALF IS WRONG

007262  026767  177744  171516          CMP     AN54+4,ANS3     ;CHECK RIGHT HALF
007270  001401                  BEQ     .+4
007272  104004                  HLT+4                   ;RIGHT HALF IS WRONG

007274  026767  177734  171506          CMP     AN54+6,ANS4     ;CHECK RIGHT HALF
007302  001401                  BEQ     .+4
007304  104004                  HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 55                TEST OF MULD FPU INSTRUCTION
                        ;    4 * 6 = 24.
                        ;    USING AC3    FPS = 47600    FEC = N/A
                        ;*****************************************************************

007306  104400                          SCOPE
007310  170127  047600                  LDFPS   #47600&57760
007314  172767  000012                  LDD     N55,3           ;LOAD 4 INTO AC3
007320  171367  000016                  MULD    M55,3           ;MULTIPLY 4 BY 5
007324  174367  171452                  STD     3,ANS1          ;STORE RESULT
007330  000414                          BR      .55

007332  040600  000000  000000  N55:    .FLT4   4
007340  000000
007342  040700  000000  000000  M55:    .FLT4   6
007350  000000
007352  041300  000000  000000  ANS5:   .FLT4   24.
007360  000000

007362  026767  177764  171412  .55:    CMP     ANS5,ANS1       ;CHECK LEFT HALF
007370  001401                          BEQ     .+4
007372  104004                          HLT+4                   ;LEFT HALF IS WRONG

007374  026767  177754  171402          CMP     ANS5+2,ANS2     ;CHECK LEFT HALF
007402  001401                          BEQ     .+4
007404  104004                          HLT+4                   ;LEFT HALF IS WRONG

007406  026767  177744  171372          CMP     ANS5+4,ANS3     ;CHECK RIGHT HALF
007414  001401                          BEQ     .+4
007416  104004                          HLT+4                   ;RIGHT HALF IS WRONG

007420  026767  177734  171362          CMP     ANS5+6,ANS4     ;CHECK RIGHT HALF
007426  001401                          BEQ     .+4
007430  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
;*****************************************************************
;TEST 56                    TEST OF MULD FPU INSTRUCTION
;    3 * 6 = 18.
;    USING AC1     FPS = 47600     FEC = N/A
;*****************************************************************

[illegible]  104400                    SCOPE
[illegible]  170127  047600            LDFPS   #47600&57760
[illegible]  172567  000012            LDD     N56.1           ;LOAD 3 INTO AC1
[illegible]  172167  000016            MULD    M56.1           ;MULTIPLY 3 BY 6
[illegible]  174167  171326            STD     1,ANS1          ;STORE RESUL
[illegible]  000414                    BR      .+56

[illegible]  040500  000000  000000  N56:    .FLT4   3
[illegible]  000000
[illegible]  040700  000000  000000  M56:    .FLT4   6
[illegible]  000000
[illegible]  041220  000000  000000  AN56:   .FLT4   18.
[illegible]  000000

[illegible]  026767  177764  171266  056:    CMP     AN56,ANS1       ;CHECK LEFT HALF
[illegible]  001401                    BEQ     .+4
[illegible]  104004                    MLT+4                   ;LEFT HALF IS WRONG

[illegible]  026767  177754  171256          CMP     AN56+2,ANS2     ;CHECK LEFT HALF
[illegible]  001401                    BEQ     .+4
[illegible]  104004                    MLT+4                   ;LEFT HALF IS WRONG

[illegible]  026767  177744  171246          CMP     AN56+4,ANS3     ;CHECK RIGHT HALF
[illegible]  001401                    BEQ     .+4
[illegible]  104004                    MLT+4                   ;RIGHT HALF IS WRONG

[illegible]  026767  177734  171236          CMP     AN56+6,ANS4     ;CHECK RIGHT HALF
[illegible]  001401                    BEQ     .+4
[illegible]  104004                    MLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 57                TEST OF MULD FPU INSTRUCTION
                        ;    6 * 2 = 12.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

                                SCOPE
                                LDFPS   #47600&57760
                                LDD     N57,0           ;LOAD 6 INTO AC0
                                MULD    N57,0           ;MULTIPLY 6 BY 2
                                STD     0,ANS           ;STORE RESULT
                                BR      .+0

                        N57:    .FLT4   6
                                .FLT4   2
                        ANS7:   .FLT4   12.

                        C57:    CMP     ANS7,ANS        ;CHECK LEFT HALF
                                BEQ     .+4
                                HLT     4               ;LEFT HALF IS WRONG

                                CMP     ANS7+2,ANS2     ;CHECK LEFT HALF
                                BEQ     .+4
                                HLT     4               ;LEFT HALF IS WRONG

                                CMP     ANS7+4,ANS3     ;CHECK RIGHT HALF
                                BEQ     .+4
                                HLT     4               ;RIGHT HALF IS WRONG

                                CMP     ANS7+6,ANS4     ;CHECK RIGHT HALF
                                BEQ     .+4
                                HLT     4               ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 60                TEST OF MULD FPU INSTRUCTION
                        ;    5 * 2 = 10.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

[illegible]  104400                         SCOPE
[illegible]  177130  047600                 LDFPS   #47600&57760
[illegible]  177146  000012                 LDD     N60,0               ;LOAD 5 INTO AC0
[illegible]  177166  000016                 MULD    M60,0               ;MULTIPLY 5 BY 2
[illegible]  177206  171056                 STD     0,ANS1              ;STORE RESULT
[illegible]  000414                         BR      .+60

000726  040640  000000  000000  N60:    .FLT4   5
000734  000000
000736  040400  000000  000000  M60:    .FLT4   2
000744  000000
000746  041040  000000  000000  AN60:   .FLT4   10.
000754  000000

000756  026767  177764  171016  .+60:   CMP     AN60,ANS1           ;CHECK LEFT HALF
000764  001401                          BEQ     .+4
000766  104004                          HLT     .+4                 ;LEFT HALF IS WRONG

000770  026767  177754  171006          CMP     AN60+2,ANS2         ;CHECK LEFT HALF
000776  001401                          BEQ     .+4
001000  104004                          HLT     .+4                 ;LEFT HALF IS WRONG

001002  026767  177744  170776          CMP     AN60+4,ANS3         ;CHECK RIGHT HALF
001010  001401                          BEQ     .+4
001012  104004                          HLT     .+4                 ;RIGHT HALF IS WRONG

001014  026767  177734  170766          CMP     AN60+6,ANS4         ;CHECK RIGHT HALF
001022  001401                          BEQ     .+4
001024  104004                          HLT     .+4                 ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 61                TEST OF MULD FPU INSTRUCTION
                        ;    5 * 3 = 15.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010026  104400                          SCOPE
010030  170127  047600                  LDFPS   #47600857760
010034  172467  000012                  LDD     N61,0           ;LOAD 5 INTO AC0
010040  171067  000016                  MULD    M61,0           ;MULTIPLY 5 BY 3
010044  174067  170732                  STD     0,ANS1          ;STORE RESULT
010050  000414                          BR      C61

010052  040640  000000  000000  N61:    .FLT4   5
010060  000000
010062  040500  000000  000000  M61:    .FLT4   3
010070  000000
010072  041160  000000  000000  AN61:   .FLT4   15.
010100  000000

010102  026767  177764  170672  C61:    CMP     AN61,ANS1       ;CHECK LEFT HALF
010110  001401                          BEQ     .+4
010112  104004                          HLT+4                   ;LEFT HALF IS WRONG

010114  026767  177754  170662          CMP     AN61+2,ANS2     ;CHECK LEFT HALF
010122  001401                          BEQ     .+4
010124  104004                          HLT+4                   ;LEFT HALF IS WRONG

010126  026767  177744  170652          CMP     AN61+4,ANS3     ;CHECK RIGHT HALF
010134  001401                          BEQ     .+4
010136  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010140  026767  177734  170642          CMP     AN61+6,ANS4     ;CHECK RIGHT HALF
010146  001401                          BEQ     .+4
010150  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 62                TEST OF MULD FPU INSTRUCTION
                        ;    5 * 4 = 20.
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010152  104400                        SCOPE
010154  170127  047600                LDFPS   #47600&57760
010160  172467  000012                LDD     N62,0           ;LOAD 5 INTO AC0
010164  171067  000016                MULD    M62,0           ;MULTIPLY 5 BY 4
010170  174067  170606                STD     0,ANS1          ;STORE RESULT
010174  000414                        BR      .+62

010176  040640  000000  000000  N62:   .FLT4   5
010204  000000
010206  040600  000000  000000  M62:   .FLT4   4
010214  000000
010216  041240  000000  000000  AN62:  .FLT4   20.
010224  000000

010226  026767  177764  170546  .+62:  CMP     AN62,ANS1       ;CHECK LEFT HALF
010234  001401                        BEQ     .+4
010236  104004                        HLT+4                   ;LEFT HALF IS WRONG

010240  026767  177754  170536        CMP     AN62+2,ANS2     ;CHECK LEFT HALF
010246  001401                        BEQ     .+4
010250  104004                        HLT+4                   ;LEFT HALF IS WRONG

010252  026767  177744  170526        CMP     AN62+4,ANS3     ;CHECK RIGHT HALF
010260  001401                        BEQ     .+4
010262  104004                        HLT+4                   ;RIGHT HALF IS WRONG

010264  026767  177734  170516        CMP     AN62+6,ANS4     ;CHECK RIGHT HALF
010272  001401                        BEQ     .+4
010274  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 63                TEST OF MULD FPU INSTRUCTION
                        ;    .33333333333333333 * 2 = .666666666666666666666667
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010276  104400                        SCOPE
010300  170127  047600                LDFPS   #47600&57760
010304  172467  000012                LDD     N63,0           ;LOAD .33333333333333333 INTO AC0
010310  171067  000016                MULD    M63,0           ;MULTIPLY .33333333333333333 BY 2
010314  174067  170462                STD     0,ANS1          ;STORE RESULT
010320  000414                        BR      063

010322  037652  125252  125252  N63:  .FLT4   .33333333333333333
010330  125253
010332  040400  000000  000000  M63:  .FLT4   2
010340  000000
010342  040052  125252  125252  AN63: .FLT4   .666666666666666666666667
010350  125253

010352  026767  177764  170422  063:  CMP     AN63,ANS1       ;CHECK LEFT HALF
010360  00140:                        BEQ     .+4
010362  104004                        HLT+4                   ;LEFT HALF IS WRONG

010364  026767  177754  170412        CMP     AN63+2,ANS2     ;CHECK LEFT HALF
010372  00140:                        BEQ     .+4
010374  104004                        HLT+4                   ;LEFT HALF IS WRONG

010376  026767  177744  170402        CMP     AN63+4,ANS3     ;CHECK RIGHT HALF
010404  001401                        BEQ     .+4
010406  104004                        HLT+4                   ;RIGHT HALF IS WRONG

010410  026767  177734  170372        CMP     AN63+6,ANS4     ;CHECK RIGHT HALF
010416  001401                        BEQ     .+4
010420  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 64                TEST OF MULD FPU INSTRUCTION
                        ;    .6666666666666667 * 2 = 1.3333333333333333333
                        ;   USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010422  104400                        SCOPE
010424  170127  047600                LDFPS   #47600&57760
010430  172467  000012                LDD     N64,0           ;LOAD .6666666666666666667 INTO AC0
010434  171067  000016                MULD    M64,0           ;MULTIFLY .6666666666666666667 BY 2
010440  174067  170336                STD     0,ANS1          ;STORE RESULT
010444  000414                        BR      C64

010446  040052  125252  125252  N64:  .FLT4   .6666666666666666667
010454  125253
010456  040400  000000  000000  M64:  .FLT4   2
010464  000000
010466  040252  125252  125252  AN64: .FLT4   1.3333333333333333333
010474  125253

010476  026767  177764  170276  C64:  CMP     AN64,ANS1       ;CHECK LEFT HALF
010504  001401                        BEQ     .+4
010506  104004                        HLT+4                   ;LEFT HALF IS WRONG

010510  026767  177754  170266        CMP     AN64+2,ANS2     ;CHECK LEFT HALF
010516  001401                        BEQ     .+4
010520  104004                        HLT+4                   ;LEFT HALF IS WRONG

010522  026767  177744  170256        CMP     AN64+4,ANS3     ;CHECK RIGHT HALF
010530  001401                        BEQ     .+4
010532  104004                        HLT+4                   ;RIGHT HALF IS WRONG

010534  026767  177734  170246        CMP     AN64+6,ANS4     ;CHECK RIGHT HALF
010542  001401                        BEQ     .+4
010544  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 65                TEST OF MULD FPU INSTRUCTION
                        ;    .6666666666666667 * .5 = .3333333333333333
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010546  104400                        SCOPE
010550  170127  047600                LDFPS   #47600&57760
010554  172467  000012                LDD     N65,0           ;LOAD .6666666666666667 INTO AC0
010560  171067  000016                MULD    M65,0           ;MULTIFLY .6666666666666667 BY .5
010564  174067  170212                STD     0,ANS1          ;STORE RESULT
010570  000414                        BR      065

010572  040052  125252  125252  N65:  .FLT4   .6666666666666667
010600  125253
010602  040000  000000  000000  M65:  .FLT4   .5
010610  000000
010612  037652  125252  125252  AN65: .FLT4   .3333333333333333
010620  125253

010622  026767  177764  170152  C65:  CMP     AN65,ANS1       ;CHECK LEFT HALF
010630  001401                        BEQ     .+4
010632  104004                        HLT+4                   ;LEFT HALF IS WRONG

010634  026767  177754  170142        CMP     AN65+2,ANS2     ;CHECK LEFT HALF
010642  001401                        BEQ     .+4
010644  104004                        HLT+4                   ;LEFT HALF IS WRONG

010646  026767  177744  170132        CMP     AN65+4,ANS3     ;CHECK RIGHT HALF
010654  001401                        BEQ     .+4
010656  104004                        HLT+4                   ;RIGHT HALF IS WRONG

010660  026767  177734  170122        CMP     AN65+6,ANS4     ;CHECK RIGHT HALF
010666  001401                        BEQ     .+4
010670  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 66                TEST OF MULD FPU INSTRUCTION
                        ;    1.3333333333333333 * .5 = .6666666666666667
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

010672  104400                        SCOPE
010674  170127  047600                LDFPS   #47600&57760
010700  172467  000012                LDD     N66,0           ;LOAD 1.3333333332333333 INTO AC0
010704  171067  000016                MULD    M66,0           ;MULTIFLY 1.3333333333333333 BY .5
010710  174067  170066                STD     0,ANS1          ;STORE RESULT
010714  000414                        BR      066

010716  040252  125252  125252  N66:  .FLT4   1.33333333333333333
010724  125253
010726  04000C  000000  000000  M66:  .FLT4   .5
010734  000000
010736  040052  125252  125252  AN66: .FLT4   .6666666666666666667
010744  125253

010746  026767  177764  170026  066:  CMP     AN66,ANS1       ;CHECK LEFT HALF
010754  001401                        BEQ     .+4
010756  104004                        HLT+4                   ;LEFT HALF IS WRONG

010760  026767  177754  170016        CMP     AN66+2,ANS2     ;CHECK LEFT HALF
010766  001401                        BEQ     .+4
010770  104004                        HLT+4                   ;LEFT HALF IS WRONG

010772  026767  177744  170006        CMP     AN65+4,ANS3     ;CHECK RIGHT HALF
011000  001401                        BEQ     .+4
011002  104004                        HLT+4                   ;RIGHT HALF IS WRONG

011004  026767  177734  167776        CMP     AN66+6,ANS4     ;CHECK RIGHT HALF
011012  001401                        BEQ     .+4
011014  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 67                TEST OF MULD FPU INSTRUCTION
                        ;    2 * .3333333333333333 = .6666666666666666666667
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

011016  104400                          SCOPE
011020  170127  047600                  LDFPS   #47600857760
011024  172467  000012                  LDD     N67,0           ;LOAD 2 INTO AC0
011030  171367  000016                  MULD    M67,0           ;MULTIPLY 2 BY .3333333333333333
011034  174067  167742                  STD     0,ANS1          ;STORE RESULT
011040  000414                          BR      067

011042  040400  000000  000000  N67:    .FLT4   2
011050  000000
011052  037652  125252  125252  M67:    .FLT4   .3333333333333333
011060  125253
011062  040052  125252  125252  AN67:   .FLT4   .6666666666666666666666667
011070  125253

011072  026767  177764  167702  067:    CMP     AN67,ANS1       ;CHECK LEFT HALF
011100  001401                          BEQ     .+4
011102  104004                          HLT+4                   ;LEFT HALF IS WRONG

011104  026767  177754  167672          CMP     AN67+2,ANS2     ;CHECK LEFT HALF
011112  001401                          BEQ     .+4
011114  104004                          HLT+4                   ;LEFT HALF IS WRONG

011116  026767  177744  167662          CMP     AN67+4,ANS3     ;CHECK RIGHT HALF
011124  001401                          BEQ     .+4
011126  104004                          HLT+4                   ;RIGHT HALF IS WRONG

011130  026767  177734  167652          CMP     AN67+6,ANS4     ;CHECK RIGHT HALF
011136  001401                          BEQ     .+4
011140  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 70                TEST OF MULD FPU INSTRUCTION
                        ;    2 * .6666666666666667 = 1.3333333333333333333
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

011142  104400                        SCOPE
011144  170127  047600                LDFPS   #47600857760
011150  172467  000012                LDD     N70,0           ;LOAD 2 INTO AC0
011154  171067  000016                MULD    M70,0           ;MULTIPLY 2 BY .6666666666666666667
011160  174067  167616                STD     0,ANS1          ;STORE RESULT
011164  000414                        BR      C70

011166  040400  000000  000000  N70:  .FLT4   2
011174  000000
011176  040052  125252  125252  M70:  .FLT4   .6666666666666666667
011204  125253
011206  040252  125252  125252  ANT0: .FLT4   1.3333333333333333333
011214  125253

011216  026767  177764  167556  C70:  CMP     ANT0,ANS1       ;CHECK LEFT HALF
011224  001401                        BEQ     .+4
011226  104004                        HLT+4                   ;LEFT HALF IS WRONG

011230  026767  177754  167546        CMP     ANT0+2,ANS2     ;CHECK LEFT HALF
011236  001401                        BEQ     .+4
011240  104004                        HLT+4                   ;LEFT HALF IS WRONG

011242  026767  177744  167536        CMP     ANT0+4,ANS3     ;CHECK RIGHT HALF
011250  001401                        BEQ     .+4
011252  104004                        HLT+4                   ;RIGHT HALF IS WRONG

011254  026767  177734  167526        CMP     ANT0+6,ANS4     ;CHECK RIGHT HALF
011262  001401                        BEQ     .+4
011264  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 71                TEST OF MULD FPU INSTRUCTION
                        ;    .5 * .6666666666666666667 = .3333333333333333333
                        ;    USING ACC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

011266  104400                        SCOPE
011270  170127  047600                LDFPS   #47600&57760
011274  172467  000012                LDD     N71,0           ;LOAD .5 INTO ACC0
011300  171067  000016                MULD    M71,0           ;MULTIPLY .5 BY .6666666666666666667
011304  174067  167472                STD     0,ANS1          ;STORE RESULT
011310  000414                        BR      .+32            ;71

011312  040000  000000  000000  N71:  .FLT4   .5
011320  000000
011322  040052  125252  125252  M71:  .FLT4   .6666666666666666667
011330  125253
011332  037652  125252  125252  AN71: .FLT4   .3333333333333333333
011340  125253

011342  026767  177764  167432  071:  CMP     AN71,ANS1       ;CHECK LEFT HALF
011350  001401                        BEQ     .+4
011352  104004                        HLT+4                   ;LEFT HALF IS WRONG

011354  026767  177754  167422        CMP     AN71+2,ANS2     ;CHECK LEFT HALF
011362  001401                        BEQ     .+4
011364  104004                        HLT+4                   ;LEFT HALF IS WRONG

011366  026767  177744  167412        CMP     AN71+4,ANS3     ;CHECK RIGHT HALF
011374  001401                        BEQ     .+4
011376  104004                        HLT+4                   ;RIGHT HALF IS WRONG

011400  026767  177734  167402        CMP     AN71+6,ANS4     ;CHECK RIGHT HALF
011406  001401                        BEQ     .+4
011410  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*************************************************************************
                        ;TEST 72                    TEST OF MULD FPU INSTRUCTION
                        ;     .5 * 1.3333333333333333 = .6666666666666667
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*************************************************************************

011412  104400                        SCOPE
011414  170127  047600                LDFPS   #47600&57760
011420  172467  000012                LDD     N72,0           ;LOAD .5 INTO AC0
011424  173067  000016                MULD    M72,0           ;MULTIPLY .5 BY 1.3333333333333333
011430  174067  167346                STD     0,ANS1          ;STORE RESULT
011434  000414                        BR      072

011436  040000  000000  000000  N72:  .FLT4   .5
011444  000000
011446  040252  125252  125252  M72:  .FLT4   1.33333333333333333
011454  125253
011456  040052  125252  125252  AN72: .FLT4   .66666666666666667
011464  125253

011466  026767  177764  167306  072:  CMP     AN72,ANS1       ;CHECK LEFT HALF
011474  001401                        BEQ     .+4
011476  104004                        HLT+4                   ;LEFT HALF IS WRONG

011500  026767  177754  167276        CMP     AN72+2,ANS2     ;CHECK LEFT HALF
011506  001401                        BEQ     .+4
011510  104004                        HLT+4                   ;LEFT HALF IS WRONG

011512  026767  177744  167266        CMP     AN72+4,ANS3     ;CHECK RIGHT HALF
011520  001401                        BEQ     .+4
011522  104004                        HLT+4                   ;RIGHT HALF IS WRONG

011524  026767  177734  167256        CMP     AN72+6,ANS4     ;CHECK RIGHT HALF
011532  001401                        BEQ     .+4
011534  104004                        HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 73                    TEST OF MULD FPU INSTRUCTION
                        ;    .33333333333333333 * 2 = .666666666666666666666667
                        ;    USING AC0 & AC4    FPS = 47600    FEC = N/A
                        ;*****************************************************************

011536  104400                          SCOPE
011540  170227  047600                  STFPS   #47600&57760
011544  172467  000016                  LDD     N73,0           ;LOAD .33333333333333333 INTO AC0
011550  174004                          STD     0,%4            ;LOAD INTO AC4
011552  172467  000020                  LDD     M73,0           ;LOAD 2 INTO 0
011556  171004                          MULD    %4,0            ;MULTIPLY .33333333333333333 * 2
011560  174067  167216                  STD     0,ANS1          ;STORE RESULT
011564  000414                          BR      .+3

011566  037652  125252  125252  N73:    .FLT4   .33333333333333333
011574  125253
011576  040400  000000  000000  M73:    .FLT4   2
011604  000000
011606  040052  125252  125252  AN73:   .FLT4   .666666666666666666666667
011614  125253

011616  026767  177764  167156  073:    CMP     AN73,ANS1       ;CHECK LEFT HALF
011624  001402                          BEQ     .+4
011626  104004                          HLT+4                   ;LEFT HALF IS WRONG

011630  026767  177754  167146          CMP     AN73+2,ANS2     ;CHECK LEFT HALF
011636  001401                          BEQ     .+4
011640  104004                          HLT+4                   ;LEFT HALF IS WRONG

011642  026767  177744  167136          CMP     AN73+4,ANS3     ;CHECK RIGHT HALF
011650  001401                          BEQ     .+4
011652  104004                          HLT+4                   ;RIGHT HALF IS WRONG

011654  026767  177734  167126          CMP     AN73+6,ANS4     ;CHECK RIGHT HALF
011662  001401                          BEQ     .+4
011664  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 74                TEST OF MULD FPU INSTRUCTION
                        ;    1 * -0 = 1
                        ;    USING AC0    FPS = 47600    FEC = N/A
                        ;*****************************************************************

011666  107400                          SCOPE
011672  170127  047600                  LDFPS   #47600&57760
011674  172467  000012                  LDD     N74,0           ;LOAD 1 INTO AC0
011700  171067  000016                  MULD    M74,0           ;MULTIPLY 1 BY -0
011704  174067  167072                  STD     0,ANS1          ;STORE RESULT
011710  000414                          BR      .+4

011712  040200  000000  000000  N74:    .FLT4   1
011720  000000
011722  100000  000000  000000  M74:    .FLT4   -0
011730  000000
011732  040200  000000  000000  AN74:   .FLT4   1
011740  000000

011742  026767  177764  167032  .+4:    CMP     AN74,ANS1       ;CHECK LEFT HALF
011750  001401                          BEQ     .+4
011752  104004                          HLT+4                   ;LEFT HALF IS WRONG

011754  026767  177754  167022          CMP     AN74+2,ANS2     ;CHECK LEFT HALF
011762  001401                          BEQ     .+4
011764  104004                          HLT+4                   ;LEFT HALF IS WRONG

011766  026767  177744  167012          CMP     AN74+4,ANS3     ;CHECK RIGHT HALF
011774  001401                          BEQ     .+4
011776  104004                          HLT+4                   ;RIGHT HALF IS WRONG

012000  026767  177734  167002          CMP     AN74+6,ANS4     ;CHECK RIGHT HALF
012006  001401                          BEQ     .+4
012010  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
012012  104400                  DONE:   SCOPE
012014  032737  002000  177570          BIT     #SW10,@#SWR     ;RING THE BELL?
012022  001005                          BNE     1$              ;NO!
012024  012767  000007  001242          MOV     #BELL,.TYPE     ;TYPE A BELL
012032  000004  013274                  TYPE    ,.TYPE
012036  005046                  1$:     CLR     -(6)            ;CLEAR TRACE TRAP
012040  032737  010000  177570          BIT     #SW12,@#SWR     ;RUN WITH TRT?
012046  001010                          BNE     2$
012050  005167  001222                  COM     TRPB
012054  100005                          BPL     2$
012056  052716  000020                  BIS     #20,(6)         ;SET TRACE TRAP
012062  012746  001062                  MOV     #BEGIN,-(6)     ;JUMP TO STAPT OF TEST
012066  000412                          BR      YESRT
012070  012746  001062          2$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
012074  013700  000042                  MOV     @#42,R0         ;GET MONITOR ADDRESS
012100  001404                          BEQ     3$              ;IF NONE
012102  004710                          JSR     7,(0)           ;GO TO MONITOR
012104  000240                          NOP
012106  000240                          NOP
012110  000240                          NOP
012112  000002                  3$:     RTI
012114  000002                  YESRT:  RTI                     ;RETURN TO PROGRAM FROM TRAP

012116  032737  000400  177570  .EMT:   BIT     #SW08,@#SWR     ;KILL LOOP OR LOOP ON SPEC. TEST
012124  001404                          BEQ     1$
012126  123767  177570  166644          CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   #SW7-0#
012134  001437                          BEQ     OVER
012136  113703  177570          1$:     MOVB    @#SWR,R3        ;GET JB BITS
012142  170003                          LDUB
012144  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST
012152  001026                          BNE     KIT
012154  032737  004000  177570          BIT     #SW11,@#SWR     ;KILL ITERATIONS
012162  001012                          BNE     SAVLAD
012164  105767  166611                  TSTB    ICNT+1
012170  001404                          BEQ     2$              ;BRANCH IF FIRST
012172  126767  001136  166601          CMPB    TIMES,ICNT+1    ;DONE?
012200  001013                          BNE     KIT             ;BRANCH IF NOT
012202  112767  000001  166571  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
012210  105267  166564          SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
012214  011667  001060                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
012220  016737  166554  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
012226  000002                          RTI                     ;RETURN

012230  105267  166545          KIT:    INCB    ICNT+1
012234  016737  166540  177570  OVER:   MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
012242  005767  001032                  TST     LAD             ;FIRST ONE?
012246  001760                          BEQ     SAVLAD
012250  016716  001024                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
012254  000002                          RTI                     ;GIVES PS
```

```
012256  032737  002000  177570  .TRP:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
012264  001405                          BEQ     1$              ;NO - SKIP
012266  012767  000007  001000          MOV     #BELL,.TYPE     ;TYPE A BELL
012274  000004  013274                  TYPE    .TYPE
012300  004767  000406          1$:     JSR     PC,ERROR        ;COUNT THE NUMBER OF ERRORS
012304  010446                          MOV     R4,-(6)
012306  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
012314  001072                          BNE     4$
012316  000004  013242                  TYPE    ,RETURN
012322  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
012326  162715  000002                  SUB     #2,(6)
012332  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
012334  004767  000212                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
012340  000004  013250                  TYPE    ,SPACE+3
012344  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
012346  004767  000200                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
012352  000004  013251                  TYPE    ,SPACE+4
012356  012703  001002                  MOV     #ANS1,R3        ;ADDRESS OF DATA
012362  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
012364  001426                          BEQ     3$
012366  100016                          BPL     2$              ;TYPE STACK?
012370  016667  000006  166404          MOV     6(6),ANS1
012376  016667  000010  166400          MOV     10(6),ANS2
012404  016667  000012  166374          MOV     12(6),ANS3
012412  016667  000014  166370          MOV     14(6),ANS4
012420  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
012424  000004  013251          2$:     TYPE    ,SPACE+4
012430  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
012432  004767  000114                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
012436  005304                          DEC     R4
012440  001371                          BNE     2$
012442  005700                  3$:     TST     FPS
012444  100016                          BPL     4$
012446  000004  013245                  TYPE    ,SPACE
012452  170367  166344                  STST    FEC
012456  016705  166340                  MOV     FEC,TTY         ;TYPE FEC IN OCTAL
012462  004767  000064                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
012466  000004  013250                  TYPE    ,SPACE+3
012472  016705  166326                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
012476  004767  000050                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
012502  012604                  4$:     MOV     (6)+,R4
012504  005737  177570                  TST     @#SWR           ;HALT ON ERROR
012510  100001                          BPL     .+4             ;SKIP IF CONTINUE
012512  000000                          HALT                    ;HALT ON ERROR!
012514  032737  001000  177570          BIT     #SW09,@#SWR     ;CHECK FOR INHIBIT LOOP ON ERROR
012522  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
012524  000002                          RTI
012526  105067  166247                  CLRB    ICNT+1
012532  032737  000400  177570          BIT     #SW08,@#SWR     ;CHECK FOR LOAD MICROBREAK
012540  001233                          BNE     KIT             ;BRANCH IF NOT
012542  113703  177570                  MOVB    @#SWR,R3        ;PUT MICROBREAK ADDRESS IN R3
012546  170003                          LOUB                    ;LOAD MICROBREAK
012550  100227                  KIT:    BR      KIT             ;LOOP ON TEST UNTIL NO ERRORS
```

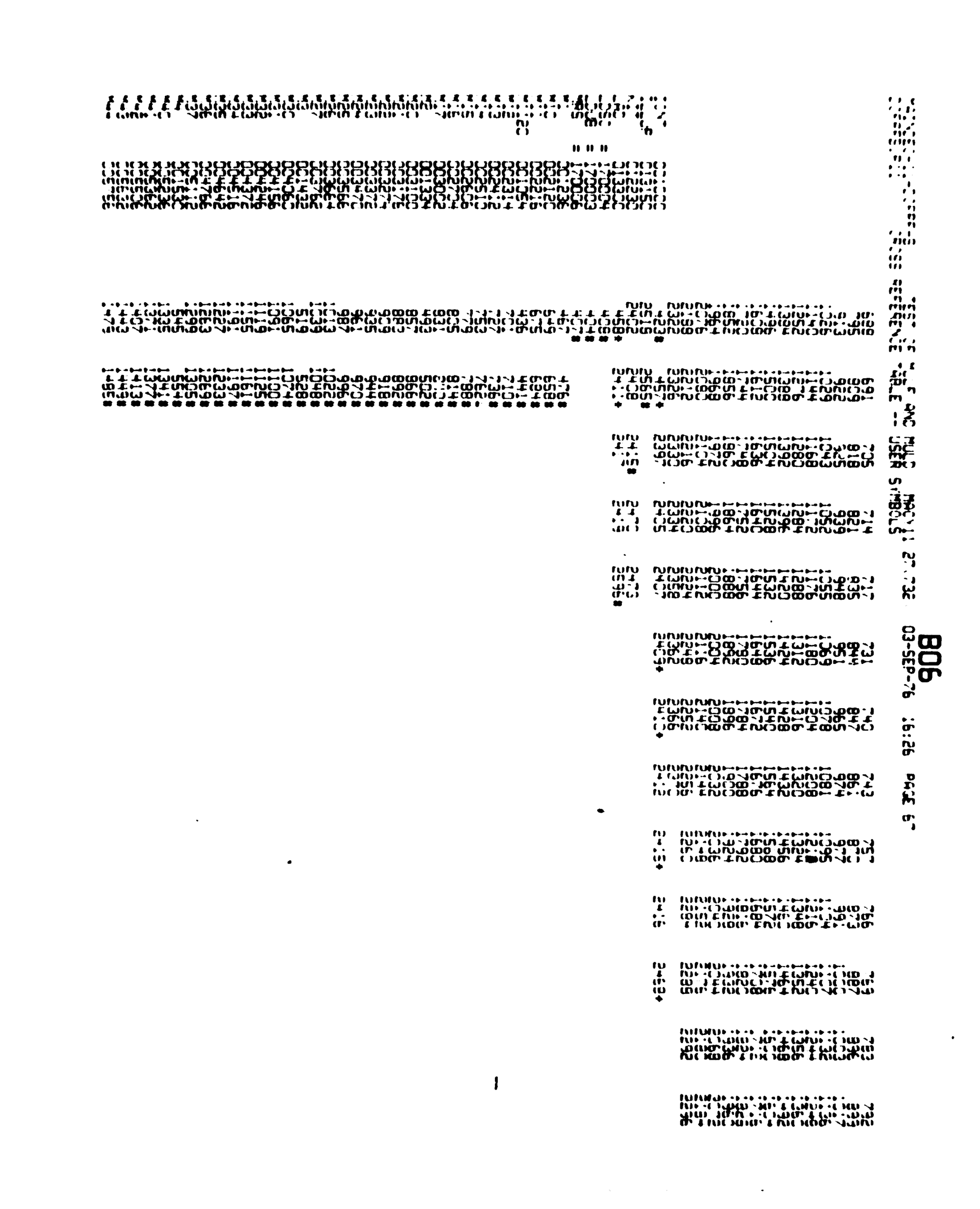
```
012552  112767  000001  000130  PRINTR: MOVB    #1,A4$          ;SET ZERO FILL SWITCH
012560  000402                  BR      .+6
012562  005067  000122          PRINTS: CLR     A4$             ;SUPRESS LEADING ZERO'S
012566  112767  177772  000115          MOVB    #-6,A4$+1       ;SET COUNT
012574  010446                  MOV     R4,-(6)         ;SAVE R4
012576  012704  012700                  MOV     #3$,R4          ;SET POINTER TO FIRST ASCII CHAR.
012602  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
012604  000405                  BR      2$              ;ROTATE FIRST BIT
012606  105014          1$:     CLRB    (4)             ;CLEAR BYTE OF CHARACTER
012610  006105                  ROL     TTY             ;ROTATE BIT INTO C
012612  106114                  ROLB    (4)             ;PACK IT
012614  006105                  ROL     TTY             ;ROTATE BIT INTO C
012616  106114                  ROLB    (4)             ;PACK IT
012620  006105          2$:     ROL     TTY             ;ROTATE BIT INTO C
012622  106114                  ROLB    (4)             ;PACK IT
012624  105714                  TSTB    (4)
012626  001402                  BEQ     .+6
012630  105267  000054          INCB    A4$
012634  105767  000050          TSTB    A4$             ;CHECK FILL SWITCH
012640  001402                  BEQ     .+6
012642  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
012646  105267  000037          INCB    A4$+1
012652  001355                  BNE     1$              ;REPEAT
012654  022704  012700          CMP     #3$,R4
012660  001002                  BNE     .+6
012662  112724  000060                  MOVB    #'0,(4)+
012666  105014                  CLRB    (4)
012670  000004  012700                  TYPE    ,3$             ;TYPE IT
012674  012604                  MOV     (6)+,R4         ;RESTORE R4
012676  000207                  RTS     PC

012700  000004          3$:     .BLKW   4
012710  000000          A4$:    0

012712  005267  000364  ERROR:  INC     ERRORS          ;COUNT ERRORS
012716  132737  000001  000041          BITB    #1,@#41         ;AUTO MODE?
012724  001412                  BEQ     1$              ;NO!
012726  022767  000010  000346          CMP     #10.,ERRORS     ;TOO MANY?
012734  001006                  BNE     1$              ;NOT YET
012736  013700  000042                  MOV     @#42,R0         ;GET ADDRESS
012742  001403                  BEQ     1$              ;FORGET IT IF ZERO
012744  005037  000042                  CLR     @#42            ;ZAP 42
012750  004710                  JSR     PC,(0)          ;CALL THE MONITOR
012752  000207          1$:     RTS     PC              ;RETURN
```

```
012754  012777  013150  000306  POWDWN: MOV     #ILLUP,@UPVEC   ;SET FOR FAST UP
012762  012777  000340  000302          MOV     #340,@UPVEC+2   ;PRIO:7
012770  170246                          STFPS   -(6)            ;GET THE FPS
012772  170011                          SETD
012774  174046                          STD     AC0,-(6)        ;SAVE AC'S
012776  174146                          STD     AC1,-(6)
013000  174246                          STD     AC2,-(6)
013002  174346                          STD     AC3,-(6)
013004  172404                          LDD     AC4,AC0
013006  174046                          STD     AC0,-(6)
013010  172405                          LDD     AC5,AC0
013012  174046                          STD     AC0,-(6)
013014  010046                          MOV     R0,-(6)         ;SAVE REGISTERS
013016  010146                          MOV     R1,-(6)
013020  010246                          MOV     R2,-(6)
013022  010346                          MOV     R3,-(6)
013024  010446                          MOV     R4,-(6)
013026  010546                          MOV     R5,-(6)
013030  010667  000220                  MOV     SP,SAVE6        ;SAVE SP
013034  012777  013044  000226          MOV     #POWUP,@UPVEC   ;SET UP VECTOR
013042  000000                          HALT

013044  016706  000204          POWUP:  MOV     SAVE6,SP        ;GET SP
013050  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
013052  005201                  1$:     INC     R1
013054  001376                          BNE     1$
013056  012605                          MOV     (6)+,R5         ;GET THE REGISTERS
013060  012604                          MOV     (6)+,R4
013062  012603                          MOV     (6)+,R3
013064  012602                          MOV     (6)+,R2
013066  012601                          MOV     (6)+,R1
013070  012600                          MOV     (6)+,R0
013072  170011                          SETD
013074  172426                          LDD     (6)+,AC0        ;RESTORE THE AC'S
013076  174005                          STD     AC0,AC5
013100  172426                          LDD     (6)+,AC0
013102  174004                          STD     AC0,AC4
013104  172726                          LDD     (6)+,AC3
013106  172626                          LDD     (6)+,AC2
013110  172526                          LDD     (6)+,AC1
013112  172426                          LDD     (6)+,AC0
013114  170126                          LDFPS   (6)+            ;RESTORE FPS
013116  012777  012754  000140          MOV     #POWDWN,@DWNVEC ;SET UP THE POWER DOWN VECTOR
013124  012777  000340  000134          MOV     #340,@DWNVEC+2
013132  000004  013136                  TYPE    ..+2            ;.ASCIZ <15><12>"POWER"
013146  000002                          RTI

013150  000000                  ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
013152  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
```

```
013154  010546                  .IOT:   MOV     TTY,-(6)        ;SAVE TTY
013156  017605  000002                  MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
013162  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
013164  001406                          BEQ     2$
013166  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
013172  105737  177564                  TSTB    @#177564        ;IS THE PRINTER READY
013176  100375                          BPL     .-4
013200  000770                          BR      1$              ;GET THE NEXT CHARACTER
013202  017646  000002          2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
013206  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
013214  022666  000002                  CMP     (6)+,2(6)       ;IS IT .+2?
013220  001006                          BNE     3$              ;NO
013222  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
013226  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
013232  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
013236  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
013240  000002                          RTI                     ;RETURN

013242  005015     000          RETURN: .ASCIZ  <15><12>        ;RETURN AND LINEFEED
013245     015  020012  020040  SPACE:  .ASCIZ  <15><12>"   "   ;RETURN AND 3 SPACES
013252     000
        013254                  .EVEN
013254  000000                  SAVE6:  0
013256  172160                  FPTADR: 172160                  ;FLOATING POINT ADDRESS ON THE 11/20
013260  000244  000246          FPVECT: 244,246                 ;FLOATING POINT VECTOR ADDRESS
013264  000024  000026          DWNVEC: 24,26                   ;POWER DOWN VECTOR ADDRESS
013270  000024  000026          UPVEC:  24,26                   ;POWER UP VECTOR ADDRESS
013274  000000                  .TYPE:  0
013276  000000                  TRPB:   0
013300  000000                  LAD:    0                       ;LOOP ADDRESS
013302  000000                  ERRORS: 0                       ;ERROR COUNT
013304  000377                  TIMES:  377                     ;ITERATION COUNT
        000001                  .END
```

```
AC0     =%000000        390*  2522  2526*  2527  2528*  2529  2551*  2552  2553*  2554  2558*
AC1     =%000001        391*  2523  2557*
AC2     =%000002        392*  2524  2556*
AC3     =%000003        393*  2525  2555*
AC4     =%000004        394*  2526  2554*
AC5     =%000005        395*  2528  2552*
ANS1      001002        424*   473*   478   499*   504   525*   530   551*   556   577*   582   603*   608
                        629*   634   655*   660   681*   686   707*   712   733*   738   759*   764   785*
                        790    811*  816    837*  842    863*  868    889*  894    915*  920    941*  946
                        967*   972   993*   998  1019*  1024  1045*  1050  1071*  1076  1097*  1102  1120*
                       1130   1156*  1166  1192*  1202  1228*  1233  1264*  1274  1300*  1310  1336*  1346
                       1372*  1382  1408*  1418  1444*  1454  1480*  1490  1516*  1526  1552*  1562  1588*
                       1598   1624*  1634  1660*  1670  1696*  1706  1732*  1742  1768*  1778  1804*  1814
                       1840*  1850  1876*  1886  1912*  1922  1948*  1958  1984*  1994  2020*  2030  2056*
                       2066   2092*  2102  2128*  2138  2164*  2174  2200*  2210  2236*  2246  2272*  2282
                       2310*  2320  2346*  2356  2438  2442*
ANS2      001004        425*   481   507   533   559   585   611   637   663   689   715   741   767
                        793    819   845   871   897   923   949   975  1001  1027  1053  1079  1105
                       1134   1170  1206  1242  1278  1314  1350  1386  1422  1458  1494  1530  1566
                       1602   1638  1674  1710  1746  1782  1818  1854  1890  1926  1962  1998  2034
                       2070   2106  2142  2178  2214  2250  2286  2324  2360  2443*
ANS3      001006        426*  1138  1174  1210  1246  1282  1318  1354  1390  1426  1462  1498  1534
                       1570   1606  1642  1678  1714  1750  1786  1822  1858  1894  1930  1966  2002
                       2038   2074  2110  2146  2182  2218  2254  2290  2328  2364  2444*
ANS4      001010        427*  1142  1178  1214  1250  1286  1322  1358  1394  1430  1466  1502  1538
                       1574   1610  1646  1682  1718  1754  1790  1926  1862  1898  1934  1970  2006
                       2042   2078  2114  2150  2186  2222  2258  2294  2332  2368  2445*
ANS5      001012        428*
ANS6      001014        429*
ANS7      001016        430*
ANS8      001020        431*
AN1       001242        477*   478   481
AN10      002124        659*   660   663
AN11      002222        685*   686   689
AN12      002320        711*   712   715
AN13      002416        737*   738   741
AN14      002514        763*   764   767
AN15      002612        789*   790   793
AN16      002710        815*   816   819
AN17      003006        841*   842   845
AN2       001340        503*   504   507
AN20      003104        867*   868   871
AN21      003202        893*   894   897
AN22      003300        919*   920   923
AN23      003376        945*   946   949
AN24      003474        971*   972   975
AN25      003572        997*   998  1001
AN26      003670       1023*  1024  1027
AN27      003766       1049*  1050  1053
AN3       001436        529*   530   533
AN30      004064       1075*  1076  1079
AN31      004162       1101*  1102  1105
AN32      004256       1127*  1130  1134  1138  1142
AN33      004402       1163*  1166  1170  1174  1178
AN34      004526       1199*  1202  1206  1210  1214
AN35      004652       1235*  1238  1242  1246  1250
```

```
AN36    004776          1271*   1274    1278    1282    1286
AN37    005122          1307*   1310    1314    1318    1322
AN4     001534           555*    556     559
AN40    005246          1343*   1346    1350    1354    1358
AN41    005372          1379*   1382    1386    1390    1394
AN42    005516          1415*   1418    1422    1426    1430
AN43    005642          1451*   1454    1458    1462    1466
AN44    005766          1487*   1490    1494    1498    1502
AN45    006112          1523*   1526    1530    1534    1538
AN46    006236          1559*   1562    1566    1570    1574
AN47    006362          1595*   1598    1602    1606    1610
AN5     001632           581*    582     585
AN50    006506          1631*   1634    1638    1642    1646
AN51    006632          1667*   1670    1674    1678    1682
AN52    006756          1703*   1706    1710    1714    1718
AN53    007102          1739*   1742    1746    1750    1754
AN54    007226          1775*   1778    1782    1786    1790
AN55    007352          1811*   1814    1818    1822    1826
AN56    007476          1847*   1850    1854    1858    1862
AN57    007622          1883*   1886    1890    1894    1898
AN6     001730           607*    608     611
AN60    007746          1919*   1922    1926    1930    1934
AN61    010072          1955*   1958    1962    1966    1970
AN62    010216          1991*   1994    1998    2002    2006
AN63    010342          2027*   2030    2034    2038    2042
AN64    010466          2063*   2066    2070    2074    2078
AN65    010612          2099*   2102    2106    2110    2114
AN66    010736          2135*   2138    2142    2146    2150
AN67    011062          2171*   2174    2178    2182    2186
AN7     002026           633*    634     637
AN70    011206          2207*   2210    2214    2218    2222
AN71    011332          2243*   2246    2250    2254    2258
AN72    011456          2279*   2282    2286    2290    2294
AN73    011606          2317*   2320    2324    2328    2332
AN74    011732          2353*   2356    2360    2364    2368
A4S     012710          2474*   2476*   2477*   2491*   2492    2495*   2506*
BEG     001026           414     435*
BEGIN   001062           439     444*   2383    2385
BELL  = 000007           379*   2375    2423
DISPLA= 177570           375*   2412*   2416*
DONE    012012          2372*
DWNVEC  013264           447*    448*   2560*   2561*   2592*
ERROR   012712          2425    2508*
ERRORS  013302          2508*   2511    2597*
FEA     001024           433*   2459
FEC     001022           418*    432*   2455*   2456
FLTERR  000760           417*    455
FPS   =%000000           380*    417*    468*    469     494*    495     520*    521     546*    547     572*    573     598*
                         599     624*    625     650*    651     676*    677     702*    703     728*    729     754*    755
                         780*    781     806*    807     832*    833     858*    859     884*    885     910*    911     936*
                         937     962*    963     988*    989     1014*   1015    1040*   1041    1066*   1067    1092*   1093
                        2452
FPTADR  013256           441    2590*
FPVECT  013260           455*    456*   2591*
HLT   = 104000           377*    471     480     483     497     506     509     523     532     535     549     558     561
                         575     584     587     601     610     613     627     536     639     653     662     665     679
```



TEST OF MULE AND MULD MACY11 27(732)
CROSS REFERENCE TABLE -- USER SYMBOLS

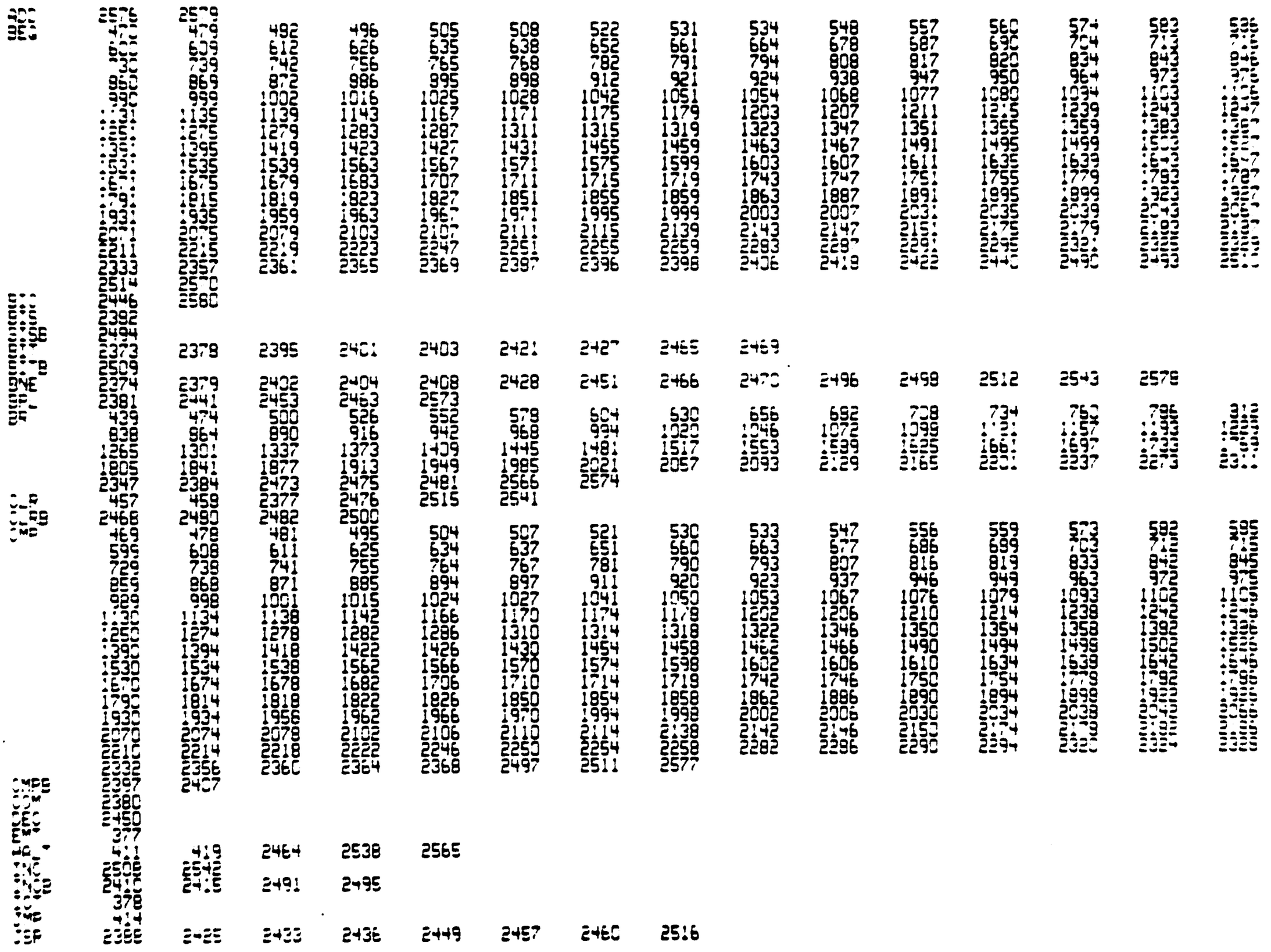
[illegible] 004632 1226 1231#
[illegible] 004756 1262 1267#
[illegible] 005126 1298 1303#
[illegible] 001524 544 553#
[illegible] 005326 1334 1339#
[illegible] 005456 1370 1375#
[illegible] 005626 1406 1411#
[illegible] 005756 1442 1447#
[illegible] 006126 1478 1483#
[illegible] 006256 1514 1519#
[illegible] 006426 1550 1555#
[illegible] 006346 1586 1591#
[illegible] 001622 570 579#
[illegible] 006466 1622 1627#
[illegible] 006616 1658 1663#
[illegible] 006736 1694 1699#
[illegible] 007062 1730 1735#
[illegible] 007226 1766 1771#
[illegible] 007332 1802 1807#
[illegible] 007456 1838 1843#
[illegible] 007602 1874 1879#
[illegible] 001720 596 605#
[illegible] 007726 1910 1915#
[illegible] 010052 1946 1951#
[illegible] 010176 1982 1987#
[illegible] 010322 2018 2023#
[illegible] 010446 2054 2059#
[illegible] 010572 2090 2095#
[illegible] 010716 2126 2131#
[illegible] 011042 2162 2167#
[illegible] 002016 622 631#
[illegible] 011166 2198 2203#
[illegible] 011312 2234 2239#
[illegible] 011436 2270 2275#
[illegible] 011566 2306 2313#
[illegible] 011712 2344 2349#
[illegible] 012234 2398 2416#
[illegible] 001244 474 478#
[illegible] 002130 656 660#
[illegible] 002226 682 686#
[illegible] 002324 708 712#
[illegible] 002422 734 738#
[illegible] 002520 760 764#
[illegible] 002616 786 790#
[illegible] 002714 812 816#
[illegible] 003012 938 842#
[illegible] 001344 500 504#
[illegible] 003110 864 868#
[illegible] 003206 890 894#
[illegible] 003304 916 920#
[illegible] 003402 942 946#
[illegible] 003500 968 972#
[illegible] 003576 994 998#
[illegible] 003674 1020 1024#
[illegible] [illegible] 1046 1050#
[illegible] [illegible] 526 530#

```
030       004070             1072    1076#
031       004166             1098    1102#
032       004266             1121    1130#
033       004412             1157    1166#
034       004536             1193    1202#
035       004662             1229    1238#
036       005006             1265    1274#
037       005132             1301    1310#
04        001540              552     556#
040       005256             1337    1346#
041       005402             1373    1382#
042       005526             1409    1418#
043       005652             1445    1454#
044       005776             1481    1490#
045       006122             1517    1526#
046       006246             1553    1562#
047       006372             1589    1598#
05        001636              578     582#
050       006516             1625    1634#
051       006642             1661    1670#
052       006766             1697    1706#
053       007112             1733    1742#
054       007236             1769    1778#
055       007362             1805    1814#
056       007506             1841    1850#
057       007632             1877    1886#
06        001734              604     608#
060       007756             1913    1922#
061       010102             1949    1958#
062       010226             1985    1994#
063       010352             2021    2030#
064       010476             2057    2066#
065       010622             2093    2102#
066       010746             2129    2138#
067       011072             2165    2174#
07        002032              630     634#
070       011216             2201    2210#
071       011342             2237    2246#
072       011466             2273    2282#
073       011616             2311    2320#
074       011742             2347    2356#
PC      =%000007              389#   2425*   2503*   2516*   2517*
POWDWN    012754              447    2518#   2560
POWUP     013044             2537    2540#
PRINTR    012552             2433    2436    2449    2457    2460    2474#
PRINTS    012562             2476#
PS      = 177776              373#
RETURN    013242             2429    2585#
R0      =%000000              381#    450*   2386*   2435    2513*   2530    2549*
R1      =%000001              382#   2531    2541*   2542*   2548*
R2      =%000002              383#   2532    2547*
R3      =%000003              384#   2399*   2438*   2471*   2533    2546*
R4      =%000004              385#   2426    2439*   2446*   2450*   2461*   2478    2479*   2497    2502*   2534    2545*
R5      =%000005              386#   2535    2544*
SAVEE     013254             2536*   2540    2589#
SAVLOC    012220             2404    2410#   2418
```

F06


```
SCOPE  = 104400     376#   464    490    516    542    568    594    620    646    672    698    724    750
                    776    902    828    854    880    906    932    958    984    1010   [illegible] 1062 [illegible]
                    1116   1152   1188   1224   1260   1296   1332   1368   1404   1440   1476   [illegible] [illegible]
                    1584   1620   1656   1692   1728   1764   1800   1836   1872   1908   1944   1980   2016
                    2052   2088   2124   2160   2196   2232   2268   2304   2342   2372
SP     =%000006     388#   435*   445*   2536   2540*
SPACE    013245     2434   2437   2447   2454   2458   2586#
[illegible] = [illegible]  405#
[illegible] = [illegible]  406#
SWR    = 177570     374#   375    2373   2378   2395   2397   2399   2401   2403   2421   2427   2462   2465
                    2469   2471
SW00   = 000400     403#   2395   2469
[illegible] = 001000  402#   2465
[illegible] = 002000  401#   2373   2421
[illegible] = 004000  400#   2403
SW12   = 010000     399#   2378
SW13   = 020000     398#   2427
SW14   = 040000     397#   2401
SW15   = 100000     396#
TIMES    013304     2407   2598#
TRPD     013276     2380*  2595#
[illegible] =%000005  387#   2432*  2435*  2448*  2456*  2459*  2483*  2485*  2487*  2567   2568*  2569   2571
                    2579*  2580*  2581   2582*
TYPE   = 000004     379#   2376   2424   2429   2434   2437   2447   2454   2458   2501   2562
[illegible] 013270  2518*  2519*  2537*  2593#
[illegible] [illegible]  438*   446    2384   2393#
.      = 013306     410#   411    412#   416#   421#   470    479    482    496    505    508    [illegible] [illegible]
                    534    548    557    560    574    583    586    600    609    612    626    [illegible] [illegible]
                    652    661    664    679    687    690    704    713    [illegible] 730 [illegible] [illegible] [illegible]
                    765    768    782    791    794    808    817    820    834    843    846    [illegible] [illegible]
                    872    886    895    898    912    921    924    939    947    950    964    [illegible] [illegible]
                    990    999    1002   1016   1025   1028   1042   1051   1054   1068   [illegible] [illegible] [illegible]
                    1103   1106   1131   1135   1139   1143   1167   1171   1175   1179   [illegible] [illegible] [illegible]
                    1215   1239   1243   1247   1251   1275   1279   1283   1287   1311   [illegible] [illegible] [illegible]
                    1347   1351   1355   1359   1383   1387   1391   1395   1419   1423   1427   [illegible] [illegible]
                    1459   1463   1467   1491   1495   1499   1503   1527   1531   1535   1539   [illegible] [illegible]
                    1571   1575   1599   1603   1607   1611   1635   1639   1643   1647   [illegible] [illegible] [illegible]
                    1683   1707   1711   1715   1719   1743   1747   1751   1755   1779   [illegible] [illegible] [illegible]
                    1815   1819   1823   1827   1851   1855   1859   1863   1887   1891   1895   [illegible] [illegible]
                    1927   1931   1935   1959   1963   1967   1971   1995   1999   2003   [illegible] [illegible] [illegible]
                    2039   2043   2067   2071   2075   2079   2103   2107   2111   [illegible] [illegible] [illegible] [illegible]
                    2151   2175   2179   2183   2187   2211   2215   2219   [illegible] [illegible] [illegible] [illegible] [illegible]
                    2263   2287   2291   2295   2321   2325   2329   2333   [illegible] [illegible] [illegible] [illegible] [illegible]
                    2466   2475   2490   2493   2498   2505#  2562   2566   2570   [illegible]#
[illegible] [illegible]  453    2395#
[illegible] [illegible]  449    2567#
[illegible] [illegible]  451    2421#
[illegible] [illegible]  2375*  2376   2423*  2424   2594#
```

JUMP	3218#	2433	2435	2448	2456	2459									
[illegible]	[illegible]#	2562													
[illegible]	[illegible]#														
[illegible]	[illegible]#	468	494	520	546	572	598	624	650	676	702	728	754	780	806
	0832	858	884	910	936	962	988	1014	1040	1066	1092				
[illegible]	[illegible]#	2375	2423												
[illegible]	[illegible]#	1110	1146	1182	1218	1254	1290	1326	1362	1398	1434	1470	1506	1542	1578
	[illegible]#	1650	1686	1722	1758	1794	1830	1866	1902	1938	1974	2010	2046	2082	2118
	[illegible]#	2190	2225	2262	2336										
[illegible]	[illegible]#	2298													
[illegible]	[illegible]#	485	511	537	563	589	615	641	667	693	719	745	771	797	823
	[illegible]#	845	871	897	923	949	975	1001	1027	1053					



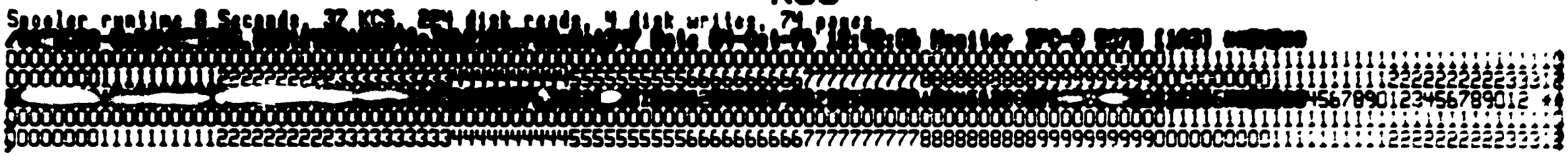
.CCC	1118	1154	1190	1226	1262	1298	1334	1370	1406	1442	1478	1514	1550	1586	1622
	1656	1692	1730	1766	1802	1838	1874	1910	1946	1982	2018	2054	2090	2126	2162
	2198	2234	2270	2306	2308	2344	2526	2528	2551	2553	2555	2556	2557	2558	
.CDF	466	492	518	544	570	596	622	648	674	700	726	752	778	804	830
	856	882	908	934	960	986	1012	1038	1064	1090					
.CDFPS	465	491	517	543	569	595	621	647	673	699	725	751	777	803	829
	855	881	907	933	959	985	1011	1037	1063	1089	1117	1153	1189	1225	1261
	1297	1333	1369	1405	1441	1477	1513	1549	1585	1621	1657	1693	1729	1765	1801
.CDS	1837	1873	1909	1945	1981	2017	2053	2089	2125	2161	2197	2233	2269	2305	2341
.CDSB	2377	2413	2449												
.CDSC	400	436	438	441	444	445	446	447	448	449	450	451	452	453	454
	455	456	2375	2383	2385	2386	2411	2412	2416	2419	2423	2426	2430	2433	2436
	2439	2442	2443	2444	2445	2448	2456	2459	2462	2478	2479	2502	2531	2538	2539
	2530	2531	2532	2533	2534	2535	2536	2537	2540	2544	2545	2546	2547	2548	2549
	2560	2561	2567	2568	2575	2581	2582								
.CDSD	2399	2409	2439	2477	2474	2477	2499	2571							
.CDSM	1118	1155	1191	1227	1263	1299	1335	1371	1407	1443	1479	1515	1551	1587	1623
	1659	1695	1731	1767	1803	1839	1875	1911	1947	1983	2019	2055	2091	2127	2163
	2199	2235	2271	2309	2345										
.CDSP	467	493	519	545	571	597	623	649	675	701	727	753	779	805	831
	857	883	909	935	961	987	1013	1039	1065	1091					
.NCP	2389	2390	2391												
.ROL	2483	2485	2487												
.ROLS	2484	2486	2488												
.RTT	420	2392	2393	2413	2420	2467	2563	2583							
.RTS	2503	2517													
.SETD	2521	2550													
.STD	1120	1156	1192	1228	1264	1300	1336	1372	1408	1444	1480	1516	1552	1588	1624
	1660	1696	1732	1768	1804	1840	1876	1912	1948	1984	2020	2056	2092	2128	2164
	2200	2236	2272	2307	2310	2346	2522	2523	2524	2525	2527	2529	2552	2554	2555
.STF	473	499	525	551	577	603	629	655	681	707	733	759	785	811	837
	863	889	915	941	967	993	1019	1045	1071	1097					
.STFPS	417	468	494	520	546	572	598	624	650	676	702	728	754	780	806
	832	858	884	910	936	962	988	1014	1040	1066	1092	2305	2520		
.STSK	418	2455													
.SWAB	2431														
.TBP	376														
.TST	437	2417	2452	2462											
.TSTB	2405	2489	2492	2569	2572										
.ASCIZ	2563	2585	2586												
.BLKW	2505														
.ENABL	371														
.END	2599														
.ENDC	473	499	525	551	577	603	629	655	681	707	733	759	785	811	837
	863	889	915	941	967	993	1019	1045	1071	1097					
.ERR	2588	2588													
.IF	476	476	477	501	502	503	527	528	529	553	554	555	579	580	581
	606	606	607	631	632	633	657	658	659	683	684	685	709	710	711
	735	736	737	761	762	763	787	788	789	813	814	815	839	840	841
	865	866	867	891	892	893	917	918	919	943	944	945	969	970	971
	995	996	997	1021	1022	1023	1047	1048	1049	1073	1074	1075	1099	1100	1101
.IFF	1125	1125	1127	1159	1161	1163	1195	1197	1199	1231	1233	1235	1267	1269	1271
.IFT	1305	1305	1307	1339	1341	1343	1375	1377	1379	1411	1413	1415	1447	1449	1451
.IFTF	1485	1485	1487	1519	1521	1523	1555	1557	1559	1591	1593	1595	1627	1629	1631
.IIF	1665	1665	1667	1699	1701	1703	1735	1737	1739	1771	1773	1775	1807	1809	1811
.IRP	1845	1845	1847	1879	1881	1883	1915	1917	1919	1951	1953	1955	1987	1989	1991

```
                2023  2025  2027  2059  2061  2063  2095  2097  2099  2131  2133  2135  2167  2169  2171
                2203  2205  2207  2239  2241  2243  2275  2277  2279  2313  2315  2317  2349  2351  2353
.IFNC            473   499   525   551   577   603   629   655   681   707   733   759   785   811   837
                 863   889   915   941   967   993  1019  1045  1071  1097
.IIF             459   469   485   495   511   521   537   547   563   573   589   599   615   625   641
                 651   667   677   693   703   719   729   745   755   771   781   797   807   823   833
                 949   859   875   885   901   911   927   937   953   963   979   989  1005  1015  1031
                1041  1057  1067  1083  1093
.LIST            321   371   411   421   459   485   511   537   563   589   615   641   667   693   719
                 745   771   797   823   849   875   901   927   953   979  1005  1031  1057  1083  1109
                1146  1182  1218  1254  1290  1326  1362  1398  1434  1470  1506  1542  1578  1614  1650
                1686  1722  1758  1794  1830  1866  1902  1938  1974  2010  2046  2082  2118  2154  2190
                2226  2262  2298  2336  2372  2421  2474  2518  2563  2567
.MACRO           371   459
.MEXIT           371
.NLIST           321   371   411   421   459   485   511   537   563   589   615   641   667   693   719
                 745   771   797   823   849   875   901   927   953   979  1005  1031  1057  1083  1109
                1146  1182  1218  1254  1290  1326  1362  1398  1434  1470  1506  1542  1578  1614  1650
                1686  1722  1758  1794  1830  1866  1902  1938  1974  2010  2046  2082  2118  2154  2190
                2226  2262  2298  2336  2372  2421  2474  2518  2563  2567
.PAGE            459   511   563   615   667   719   771   823   875   927   979  1031  1083  1135  1187
                1218  1254  1290  1326  1362  1398  1434  1470  1506  1542  1578  1614  1650  1686  1722
                1758  1794  1830  1866  1902  1938  1974  2010  2046  2082  2118  2154  2190  2226  2262
                2298  2336
.REM             330
.REPT            411
.SBTTL           327   371   421   459  2372  2421  2474  2518  2567
.TITLE           327
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DCFPFB,DCFPFB.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DCFPFB.P11
RUN-TIME: 12 20 4 SECONDS
RUN-TIME RATIO: 2787/38=72.1
CORE USED: 9K (18 PAGES)

KOБ


Spooler running 8 Seconds, 32 KCS, 8251 disk reads, 4 disk writes, 71 pages
0000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111112222222223333
4567890123456789012