FP11

DIVF DIVD

MD-11-DCFPG-C

EP DCFPG C DL A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

.SBTTL

.ENABL LC

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPB
PRODUCT NAME: FP11 BASIC INSTRUCTION TESTS
DATE CREATED: MARCH 12, 1973
MAINTAINER: DIAGNOSTIC GROUP
AUTHORS: BOB BRAIN & KEN CHAPMAN

MAINDEC NO.	INSTRUCTIONS TESTED
DCFPA	LDFPS, STFPS, SETI, SETL SETF, SETD, CFCC
DCFPB	STST
DCFPC	LDF, LDD, STF, STD
DCFPD	ADDF, ADDD, SUBF, SUBD
DCFPE	CMPF, CMPD
DCFPF	MULF, MULD
DCFPG	DIVF, DIVD
DCFPH	CLRF, CLRD, TSTF, TSTD ABSF, ABSD, NEGF, NEGD
DCFPI	LDCFD, LDCDF, STCFD, STCDF
DCFPJ	LDCIF, LDCLF, LDCID, LDCLD STCFI, STCFL, STCDI, STCDL
DCFPK	LDEXP, STEXP
DCFPL	MODF, MODD

TABLE OF CONTENTS

CONTENTS

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.

```
                        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
```

DESCRIPTION

DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 .. ALL SWITCHES DOWN IS WORST CASE TESTING. IF AN ERROR OCCURS, THAT TEST WILL BE LOOPED UPON UNTIL COMPLETION OF 256 CONSECUTIVE PASSES WITH NO ERRORS OF THE SUBTEST IF SW<9> SET TO A 1. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... SCOPE LOOP
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 ..... INHIBIT TRACE TRAPPING
SW<11> = 1 ..... INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 ..... BELL ON ERROR
         0 ..... BELL ON PASS COMPLETE
SW<09> = 1 ..... LOOP ON ERROR
SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>
         0 ..... LOAD SW<7:0> INTO UB REGISTER
```

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1.) IF A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 256 CONSECUTIVE GOOD PASSES ARE COMPLETED IF SW<9> IS ON A 1. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRTRAP

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

5.2.4 TRAPCATCHER

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

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 2 SCOPE STATEMENTS) WHICH ARE RUN 256 TIMES BEFORE CONTINUING TO THE NEXT. SW<11> ON A 1 CAUSES EACH SUBTEST TO BE RUN ONLY ONCE. SW<9> ON A 1 ENABLES LOOP ON ERROR. THE ADDRESS ICNT (LOC 1000) AND DISPLAY REGISTER ON THE 11/45 EACH CONTAIN THE ITERATION COUNT IN THE LEFT BYTE AND THE TEST NUMBER IN THE RIGHT BYTE. ALL THE SUBTESTS SHOULD BE RUN SEQUENTIALLY BY STARTING AT 200 NOT BY STARTING AT THE BEGINNING OF THE SUBTEST. TO LOOP ON A PARTICULAR SUBTEST, PUT THE TEST NUMBER (SEE LISTING) IN THE RIGHT BYTE OF THE SWITCH REGISTER AND SW<8> ON A 1. THIS TEST WILL BE LOOPED UPON UNTIL SW<8> IS PUT ON A 0 OR THE RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE PROGRAM WILL BE RUN AS USUAL.

.ENDR

```
                        .TITLE  MAINDEC-11-DCFPG-C      TEST OF DIVF AND DIVD
                        :COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
                        :PROGRAM BY BOB BRAIN & KEN CHAPMAN
                        .REM*

                                SWITCH              USE
                                ------              --------------------
                                   8                0 - LOAD UB REGISTER WITH SW<7:0>
                                                    1 - LOOP ON TEST IN SW<7:0>
                                   9                LOOP ON ERROR
                                  10                0 - BELL ON PASS COMPLETE
                                                    1 - BELL ON ERROR
                                  11                INHIBIT ITERATIONS
                                  12                INHIBIT TRACE TRAP
                                  13                INHIBIT ERROR TYPEOUTS
                                  14                LOOP ON TEST
                                  15                HALT ON ERROR


                        OUTPUT FORM:

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


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

```
                                .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      TTY=    %5
                    000006      SP=     %6
                    000007      PC=     %7
                    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      LDUB=   170003
                    170005      STA0=   170005
                    170007      STQ0=   170007
                    170006      MRS=    170006
                    170004      LDSC=   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  010256                  MOV     #6,YESRT        ;FUDGE IN RTT IF 11/45
001052  000403                                  BR      BEGIN
                
001054  016737  011412  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  011330  000014          MOV     #YESRT,@#14     ;SET TRACE TRAP VECTOR
001102  012777  012170  011370          MOV     #POWDWN,@DWNVEC
001110  012777  000340  011364          MOV     #340,@DWNVEC+2
001116  012737  012370  000020          MOV     #.IOT,@#20      ;SET UP VECTOR 20
001124  012700  000030                  MOV     #30,R0          ;SET R0 TO VECTOR 30
001130  012720  011472                  MOV     #.TRP,(0)+      ;SET EMT VECTOR
001134  012720  000340                  MOV     #340,(0)+
001140  012720  011332                  MOV     #.EMT,(0)+      ;SET TRAP VECTOR
001144  012710  000340                  MOV     #340,(0)
001150  012777  000760  011316          MOV     #FLTERR,@FPVECT ;LOAD INTERUPT VECTOR
001156  012777  000340  011312          MOV     #340,@FPVECT+2  ;LOCK UP PROCESSOR
001164  005067  177610                  CLR     ICNT
001170  005067  011320                  CLR     LAD
```

```
                                ;*****************************************************************
                                ;TEST 1         TEST OF DIVF FPU INSTRUCTION
                                ;    0 / 1 = 0
                                ;       USING AC0       FPS = 47404  FEC = N/A
                                ;*****************************************************************
001174  104400                          SCOPE
001176  170127  047400                  LDFPS   #47404&57760
001202  172467  000024                  LDF     N1,0            ;LOAD 0 INTO AC0
001206  174467  000024                  DIVF    M1,0            ;DIVIDE 0 BY 1
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  040200  000000          M1:     .FLT2   1
001242  000000  000000          AN1:    .FLT2   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 DIVF FPU INSTRUCTION
                                ;    1 / 1 = 1
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
001272  104400                          SCOPE
001274  170127  047400                  LDFPS   #47400&57760
001300  172467  000024                  LDF     N2,0            ;LOAD 1 INTO AC0
001304  174467  000024                  DIVF    M2,0            ;DIVIDE 1 BY 1
001310  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001312  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
001316  001401                          BEQ     .+4             ;BRANCH IF OK
001320  104000                          HLT                     ;FPS NOT EQUAL TO 47400

001322  174067  177454                  STF     0,ANS1          ;STORE RESULT
001326  000406                          BR      02
001330  040200  000000          N2:     .FLT2   1
001334  040200  000000          M2:     .FLT2   1
001340  040200  000000          AN2:    .FLT2   1
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 DIVF FPU INSTRUCTION
                                ;       2 / 1 = 2
                                ;          USING AC0        FPS = 47400  FEC = N/A
                                ;*****************************************************************
001370  104400                          SCOPE
001372  170127  047400                  LDFPS   #47400&57760
001376  172467  000024                  LDF     N3,0                ;LOAD 2 INTO AC0
001402  174467  000024                  DIVF    M3,0                ;DIVIDE 2 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      03
001426  040400  000000          N3:     .FLT2   2
001432  040200  000000          M3:     .FLT2   1
001436  040400  000000          AN3:    .FLT2   2
001442  026767  177770  177332  03:     CMP     AN3,ANS1            ;CHECK LEFT HALF
001450  001401                          BEQ     .+4
001452  104002                          HLT+2                       ;LEFT HALF IS WRONG
001454  026767  177760  177322          CMP     AN3+2,ANS2          ;CHECK RIGHT HALF
001462  001401                          BEQ     .+4
001464  104002                          HLT+2                       ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 4          TEST OF DIVF FPU INSTRUCTION
                                ;       3 / 1 = 3
                                ;          USING AC0        FPS = 47400  FEC = N/A
                                ;*****************************************************************
001466  104400                          SCOPE
001470  170127  047400                  LDFPS   #47400&57760
001474  172467  000024                  LDF     N4,0                ;LOAD 3 INTO AC0
001500  174467  000024                  DIVF    M4,0                ;DIVIDE 3 BY 1
001504  170200                          STFPS   FPS                 ;STORE FLOATING POINT STATUS
001506  022700  047400                  CMP     #47400,FPS          ;CHECK FLOATING POINT STATUS
001512  001401                          BEQ     .+4                 ;BRANCH IF OK
001514  104000                          HLT                         ;FPS NOT EQUAL TO 47400

001516  174067  177260                  STF     0,ANS1              ;STORE RESULT
001522  000406                          BR      04
001524  040500  000000          N4:     .FLT2   3
001530  040200  000000          M4:     .FLT2   1
001534  040500  000000          AN4:    .FLT2   3
001540  026767  177770  177234  04:     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 DIVF FPU INSTRUCTION
                                ;     4 / 1 = 4
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
001564  104400                          SCOPE
001566  170127  047400                  LDFPS   #47400&57760
001572  172467  000024                  LDF     N5,0            ;LOAD 4 INTO AC0
001576  174467  000024                  DIVF    M5,0            ;DIVIDE 4 BY 1
001602  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001604  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
001610  001401                          BEQ     .+4             ;BRANCH IF OK
001612  104000                          HLT                     ;FPS NOT EQUAL TO 47400

001614  174067  177162                  STF     0,ANS1          ;STORE RESULT
001620  000406                          BR      05
001622  040600  000000          N5:     .FLT2   4
001626  040200  000000          M5:     .FLT2   1
001632  040600  000000          AN5:    .FLT2   4
001636  026767  177770  177136  05:     CMP     AN5,ANS1        ;CHECK LEFT HALF
001644  001401                          BEQ     .+4
001646  104002                          HLT+2                   ;LEFT HALF IS WRONG
001650  026767  177760  177126          CMP     AN5+2,ANS2      ;CHECK RIGHT HALF
001656  001401                          BEQ     .+4
001660  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 6         TEST OF DIVF FPU INSTRUCTION
                                ;     5 / 1 = 5
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
001662  104400                          SCOPE
001664  170127  047400                  LDFPS   #47400&57760
001670  172467  000024                  LDF     N6,0            ;LOAD 5 INTO AC0
001674  174467  000024                  DIVF    M6,0            ;DIVIDE 5 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  040640  000000          N6:     .FLT2   5
001724  040200  000000          M6:     .FLT2   1
001730  040640  000000          AN6:    .FLT2   5
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 9          TEST OF DIVF FPU INSTRUCTION
                                        ;     6 / 1 = 6
                                        ;        USING ACO       FPS = 47400  FEC = N/A
                                        ;*****************************************************************
                                                SCOPE
002046  104400                                  SCOPE
002050  170127  047400                          LDFPS   #47400&57760
002054  172467  000024                          LDF     N7.0            ;LOAD 6 INTO ACO
002060  174467  000024                          DIVF    M7.0            ;DIVIDE 6 BY 1
002064  170200                                  STFPS   FPS             ;STORE FLOATING POINT STATUS
002066  022700  047400                          CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002072  001401                                  BEQ     .+4             ;BRANCH IF OK
002074  104000                                  HLT                     ;FPS NOT EQUAL TO 47400

002076  174067  176766                          STF     0,ANS1          ;STORE RESULT
002102  000406                                  BR      OK
002104  040740  000000          N7:             .FLT2   6
002110  040200  000000          M7:             .FLT2   1
002114  040740  000000          AN7:            .FLT2   6
002120  026767  177770  176742  OK:             CMP     AN7,ANS1        ;CHECK LEFT HALF
002126  001401                                  BEQ     .+4
002130  104002                                  HLT+2                   ;LEFT HALF IS WRONG
002132  026767  177760  176732                  CMP     AN7+2,ANS2      ;CHECK RIGHT HALF
002140  001401                                  BEQ     .+4
002142  104002                                  HLT+2                   ;RIGHT HALF IS WRONG

                                        ;*****************************************************************
                                        ;TEST 10         TEST OF DIVF FPU INSTRUCTION
                                        ;     7 / 1 = 7
                                        ;        USING ACO       FPS = 47400  FEC = N/A
                                        ;*****************************************************************
002056  104400                                  SCOPE
002060  170127  047400                          LDFPS   #47400&57760
002064  172467  000024                          LDF     N10.0           ;LOAD 7 INTO ACO
002070  174467  000024                          DIVF    M10.0           ;DIVIDE 7 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      O10
002114  040740  000000          N10:            .FLT2   7
002120  040200  000000          M10:            .FLT2   1
002124  040740  000000          AN10:           .FLT2   7
002130  026767  177770  176644  O10:            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 DIVF FPU INSTRUCTION
                                ;       7 / 2 = 3.5
                                ;         USING AC1     FPS = 47400  FEC = N/A
                                ;*****************************************************************
002154  104400                          SCOPE
002156  170127  047400                  LDFPS   #47400&57760
002162  172567  000024                  LDF     N11,1           ;LOAD 7 INTO AC1
002166  174567  000024                  DIVF    M11,1           ;DIVIDE 7 BY 2
002172  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002174  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002200  001401                          BEQ     .+4             ;BRANCH IF OK
002202  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002204  174167  176572                  STF     1,ANS1          ;STORE RESULT
002210  000406                          BR      011
002212  040740  000000          N11:    .FLT2   7
002216  040400  000000          M11:    .FLT2   2
002222  040540  000000          AN11:   .FLT2   3.5
002226  026767  177770  176546  011:    CMP     AN11,ANS1       ;CHECK LEFT HALF
002234  001401                          BEQ     .+4
002236  104002                          HLT+2                   ;LEFT HALF IS WRONG
002240  026767  177760  176536          CMP     AN11+2,ANS2     ;CHECK RIGHT HALF
002246  001401                          BEQ     .+4
002250  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 12                TEST OF DIVF FPU INSTRUCTION
                                ;       7 / 3 = 2.333333333333333
                                ;         USING AC3     FPS = 47400  FEC = N/A
                                ;*****************************************************************
002252  104400                          SCOPE
002254  170127  047400                  LDFPS   #47400&57760
002260  172767  000024                  LDF     N12,3           ;LOAD 7 INTO AC3
002264  174767  000024                  DIVF    M12,3           ;DIVIDE 7 BY 3
002270  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002272  022700  047400                  CMP     #47400,FPS      ;CHECK FLOATING POINT STATUS
002276  001401                          BEQ     .+4             ;BRANCH IF OK
002300  104000                          HLT                     ;FPS NOT EQUAL TO 47400

002302  174367  176474                  STF     3,ANS1          ;STORE RESULT
002306  000406                          BR      012
002310  040740  000000          N12:    .FLT2   7
002314  040500  000000          M12:    .FLT2   3
002320  040425  052525          AN12:   .FLT2   2.333333333333333
002324  026767  177770  176450  012:    CMP     AN12,ANS1       ;CHECK LEFT HALF
002332  001401                          BEQ     .+4
002334  104002                          HLT+2                   ;LEFT HALF IS WRONG
002336  026767  177760  176440          CMP     AN12+2,ANS2     ;CHECK RIGHT HALF
002344  001401                          BEQ     .+4
002346  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 13                TEST OF DIVF FPU INSTRUCTION
                                ;       7 / 4 = 1.75
                                ;       USING AC2       FPS = 47400  FEC = N/A
                                ;*****************************************************************
002350  104400                          SCOPE
002352  170127  047400                  LDFPS   #47400&57760
002356  172667  000024                  LDF     N13,2           ;LOAD 7 INTO AC2
002362  174667  000024                  DIVF    M13,2           ;DIVIDE 7 BY 4
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  174267  176376                  STF     2,ANS1          ;STORE RESULT
002404  000406                          BR      013
002406  040740  000000          N13:    .FLT2   7
002412  040600  000000          M13:    .FLT2   4
002416  040340  000000          AN13:   .FLT2   1.75
002422  026767  177770  176352  013:    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 DIVF FPU INSTRUCTION
                                ;       7 / 5 = 1.4
                                ;       USING AC1       FPS = 47400  FEC = N/A
                                ;*****************************************************************
002446  104400                          SCOPE
002450  170127  047400                  LDFPS   #47400&57760
002454  172567  000024                  LDF     N14,1           ;LOAD 7 INTO AC1
002460  174567  000024                  DIVF    M14,1           ;DIVIDE 7 BY 5
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  174167  176300                  STF     1,ANS1          ;STORE RESULT
002502  000406                          BR      014
002504  040740  000000          N14:    .FLT2   7
002510  040640  000000          M14:    .FLT2   5
002514  040263  031463          AN14:   .FLT2   1.4
002520  026767  177770  176254  014:    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 DIVF FPU INSTRUCTION
                        ;    7 / 6 = 1.1666666666666666667
                        ;       USING AC3       FPS = 47400  FEC = N/A
                        ;*****************************************************************
002544  104400                  SCOPE
002546  170127  047400          LDFPS   #47400&57760
002552  172767  000024          LDF     N15,3           ;LOAD 7 INTO AC3
002556  174767  000024          DIVF    M15,3           ;DIVIDE 7 BY 6
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  174367  176202          STF     3,ANS1          ;STORE RESULT
002600  000406                  BR      O15
002602  040740  000000  N15:    .FLT2   7
002606  040700  000000  M15:    .FLT2   6
002612  040225  052525  AN15:   .FLT2   1.1666666666666666667
002616  026767  177770  176156  O15:    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 DIVF FPU INSTRUCTION
                        ;    6 / 5 = 1.2
                        ;       USING AC0       FPS = 47400  FEC = N/A
                        ;*****************************************************************
002642  104400                  SCOPE
002644  170127  047400          LDFPS   #47400&57760
002650  172467  000024          LDF     N16,0           ;LOAD 6 INTO AC0
002654  174467  000024          DIVF    M16,0           ;DIVIDE 6 BY 5
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  174067  176104          STF     0,ANS1          ;STORE RESULT
002676  000406                  BR      O16
002700  040700  000000  N16:    .FLT2   6
002704  040640  000000  M16:    .FLT2   5
002710  040231  114632  AN16:   .FLT2   1.2
002714  026767  177770  176060  O16:    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 DIVF FPU INSTRUCTION
                                ;    6 / 4 = 1.5
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
002740  104400                          SCOPE
002742  170127  047400                  LDFPS   #47400&57760
002746  172467  000024                  LDF     N17,0           ;LOAD 6 INTO AC0
002752  174467  000024                  DIVF    M17,0           ;DIVIDE 6 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                          HLT                     ;FPS NOT EQUAL TO 47400

002770  174067  176006                  STF     0,ANS1          ;STORE RESULT
002774  000406                          BR      017
002776  040700  000000          N17:    .FLT2   6
003002  040600  000000          M17:    .FLT2   4
003006  040300  000000          AN17:   .FLT2   1.5
003012  026767  177770  175762  017:    CMP     AN17,ANS1       ;CHECK LEFT HALF
003020  001401                          BEQ     .+4
003022  104002                          HLT+2                   ;LEFT HALF IS WRONG
003024  026767  177760  175752          CMP     AN17+2,ANS2     ;CHECK RIGHT HALF
003032  001401                          BEQ     .+4
003034  104002                          HLT+2                   ;RIGHT HALF IS WRONG

                                ;*****************************************************************
                                ;TEST 20                TEST OF DIVF FPU INSTRUCTION
                                ;    6 / 3 = 2
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003036  104400                          SCOPE
003040  170127  047400                  LDFPS   #47400&57760
003044  172467  000024                  LDF     N20,0           ;LOAD 6 INTO AC0
003050  174467  000024                  DIVF    M20,0           ;DIVIDE 6 BY 3
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                          HLT                     ;FPS NOT EQUAL TO 47400

003066  174067  175710                  STF     0,ANS1          ;STORE RESULT
003072  000406                          BR      020
003074  040700  000000          N20:    .FLT2   6
003100  040500  000000          M20:    .FLT2   3
003104  040400  000000          AN20:   .FLT2   2
003110  026767  177770  175664  020:    CMP     AN20,ANS1       ;CHECK LEFT HALF
003116  001401                          BEQ     .+4
003120  104002                          HLT+2                   ;LEFT HALF IS WRONG
003122  026767  177760  175654          CMP     AN20+2,ANS2     ;CHECK RIGHT HALF
003130  001401                          BEQ     .+4
003132  104002                          HLT+2                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 21                TEST OF DIVF FPU INSTRUCTION
                                ;       6 / 2 = 3
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003134  104400                          SCOPE
003136  170127  047400                  LDFPS   #47400&57760
003142  172467  000024                  LDF     N21,0           ;LOAD 6 INTO AC0
003146  174467  000024                  DIVF    M21,0           ;DIVIDE 6 BY 2
003152  170200                          STFPS   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  174067  175612                  STF     0,ANS1          ;STORE RESULT
003170  000406                          BR      021
003172  040700  000000          N21:    .FLT2   6
003176  040400  000000          M21:    .FLT2   2
003202  040500  000000          AN21:   .FLT2   3
003206  026767  177770  175566  021:    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 DIVF FPU INSTRUCTION
                                ;       5 / 2 = 2.5
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003232  104400                          SCOPE
003234  170127  047400                  LDFPS   #47400&57760
003240  172467  000024                  LDF     N22,0           ;LOAD 5 INTO AC0
003244  174467  000024                  DIVF    M22,0           ;DIVIDE 5 BY 2
003250  170200                          STFPS   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  174067  175514                  STF     0,ANS1          ;STORE RESULT
003266  000406                          BR      022
003270  040640  000000          N22:    .FLT2   5
003274  040400  000000          M22:    .FLT2   2
003300  040440  000000          AN22:   .FLT2   2.5
003304  026767  177770  175470  022:    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 DIVF FPU INSTRUCTION
                                ;    5 / 3 = 1.6666666666666666667
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003330  104400                          SCOPE
003332  170127  047400                  LDFPS   #47400&57760
003336  172467  000024                  LDF     N23,0           ;LOAD 5 INTO AC0
003342  174467  000024                  DIVF    M23,0           ;DIVIDE 5 BY 3
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  174067  175416                  STF     0,ANS1          ;STORE RESULT
003364  000406                          BR      023
003366  040640  000000          N23:    .FLT2   5
003372  040500  000000          M23:    .FLT2   3
003376  040325  052525          AN23:   .FLT2   1.6666666666666666667
003402  026767  177770  175372  023:    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 DIVF FPU INSTRUCTION
                                ;    5 / 4 = 1.25
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003426  104400                          SCOPE
003430  170127  047400                  LDFPS   #47400&57760
003434  172467  000024                  LDF     N24,0           ;LOAD 5 INTO AC0
003440  174467  000024                  DIVF    M24,0           ;DIVIDE 5 BY 4
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  174067  175320                  STF     0,ANS1          ;STORE RESULT
003462  000406                          BR      024
003464  040640  000000          N24:    .FLT2   5
003470  040600  000000          M24:    .FLT2   4
003474  040240  000000          AN24:   .FLT2   1.25
003500  026767  177770  175274  024:    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 DIVF FPU INSTRUCTION
                                ;     4 / 3 = 1.3333333333333333333
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003524  104400                          SCOPE
003526  170127  047400                  LDFPS   #47400&57760
003532  172467  000024                  LDF     N25,0           ;LOAD 4 INTO AC0
003536  174467  000024                  DIVF    M25,0           ;DIVIDE 4 BY 3
003542  170200                          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  174067  175222                  STF     0,ANS1          ;STORE RESULT
003560  000406                          BR      025
003562  040600  000000          N25:    .FLT2   4
003566  040500  000000          M25:    .FLT2   3
003572  040252  125253          AN25:   .FLT2   1.3333333333333333333
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 DIVF FPU INSTRUCTION
                                ;     3 / 2 = 1.5
                                ;       USING AC0       FPS = 47400  FEC = N/A
                                ;*****************************************************************
003622  104400                          SCOPE
003624  170127  047400                  LDFPS   #47400&57760
003630  172467  000024                  LDF     N26,0           ;LOAD 3 INTO AC0
003634  174467  000024                  DIVF    M26,0           ;DIVIDE 3 BY 2
003640  170200                          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  174067  175124                  STF     0,ANS1          ;STORE RESULT
003656  000406                          BR      026
003660  040500  000000          N26:    .FLT2   3
003664  040400  000000          M26:    .FLT2   2
003670  040300  000000          AN26:   .FLT2   1.5
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
```

```
1032                            ;*****************************************************************
1033                            ;TEST 27                TEST OF DIVD FPU INSTRUCTION
1034                            ;    0 / 1 = 0
1035                            ;       USING AC0       FPC = 47604     FEC = N/A
1036                            ;*****************************************************************
1037
1038  003720  104400                        SCOPE
1039  003722  170127  047600                LDFPS   #47604&57760
1040  003726  172467  000024                LDD     N27,0           ;LOAD 0 INTO AC0
1041  003732  174467  000030                DIVD    M27,0           ;DIVIDE 0 BY 1
1042  003736  170200                        STFPS   FPS             ;STORE FLOATING POINT STATUS
1043  003740  022700  047604                CMP     #47604,FPS      ;CHECK FLOATING POINT STATUS
1044  003744  001401                        BEQ     .+4             ;BRANCH IF OK
1045  003746  104000                        HLT                     ;FPS NOT EQUAL TO 47604
1046
1047  003750  174067  175026                STD     0,ANS1
1048  003754  000414                        BR      027
1049
1050  003756  000000  000000  000000  N27:   .FLT4   0
1051  003764  000000
1052  003766  040200  000000  000000  M27:   .FLT4   1
1053  003774  000000
1054  003776  000000  000000  000000  AN27:  .FLT4   0
1055  004004  000000
1056
1057  004006  026767  177764  174766  C27:   CMP     AN27,ANS1       ;CHECK LEFT HALF
1058  004014  001401                        BEQ     .+4
1059  004016  104004                        HLT+4                   ;LEFT HALF IS WRONG
1060
1061  004020  026767  177754  174756        CMP     AN27+2,ANS2     ;CHECK LEFT HALF
1062  004026  001401                        BEQ     .+4
1063  004030  104004                        HLT+4                   ;LEFT HALF IS WRONG
1064
1065  004032  026767  177744  174746        CMP     AN27+4,ANS3     ;CHECK RIGHT HALF
1066  004040  001401                        BEQ     .+4
1067  004042  104004                        HLT+4                   ;RIGHT HALF IS WRONG
1068
1069  004044  026767  177734  174736        CMP     AN27+6,ANS4     ;CHECK RIGHT HALF
1070  004052  001401                        BEQ     .+4
1071  004054  104004                        HLT+4                   ;RIGHT HALF IS WRONG
1072
```

```
1073                                    ;*****************************************************************
1074                                    ;TEST 30                TEST OF DIVD FPU INSTRUCTION
1075                                    ;    1 / 1 = 1
1076                                    ;       USING AC0       FPC = 47600     FEC = N/A
1077                                    ;*****************************************************************
1078
1079  004056  104400                            SCOPE
1080  004060  170127  047600                    LDFPS   #47600&57760
1081  004064  172467  000024                    LDD     N30,0           ;LOAD 1 INTO AC0
1082  004070  174467  000030                    DIVD    M30,0           ;DIVIDE 1 BY 1
1083  004074  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
1084  004076  022700  047600                    CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
1085  004102  001401                            BEQ     .+4             ;BRANCH IF OK
1086  004104  104000                            HLT                     ;FPS NOT EQUAL TO 47600
1087
1088  004106  174067  174670                    STD     0,ANS1
1089  004112  000414                            BR      C30
1090
1091  004114  040200  000000  000000  N30:      .FLT4   1
1092  004122  000000
1093  004124  040200  000000  000000  M30:      .FLT4   1
1094  004132  000000
1095  004134  040200  000000  000000  AN30:     .FLT4   1
1096  004142  000000
1097
1098  004144  026767  177764  174630  C30:      CMP     AN30,ANS1       ;CHECK LEFT HALF
1099  004152  001401                            BEQ     .+4
1100  004154  104004                            HLT+4                   ;LEFT HALF IS WRONG
1101
1102  004156  026767  177754  174620            CMP     AN30+2,ANS2     ;CHECK LEFT HALF
1103  004164  001401                            BEQ     .+4
1104  004166  104004                            HLT+4                   ;LEFT HALF IS WRONG
1105
1106  004170  026767  177744  174610            CMP     AN30+4,ANS3     ;CHECK RIGHT HALF
1107  004176  001401                            BEQ     .+4
1108  004200  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1109
1110  004202  026767  177734  174600            CMP     AN30+6,ANS4     ;CHECK RIGHT HALF
1111  004210  001401                            BEQ     .+4
1112  004212  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1113
```

```
                        ;*****************************************************************
                        ;TEST 31                TEST OF DIVD FPU INSTRUCTION
                        ;    2 / 1 = 2
                        ;       USING AC0       FPC = 47600     FEC = N/A
                        ;*****************************************************************

004214  104400                  SCOF
004216  170127  047600          LDFPS   #47600&57760
004222  172467  000024          LDD     N31,0           ;LOAD 2 INTO AC0
004226  174467  000030          DIVD    M31,0           ;DIVIDE 2 BY 1
004232  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
004234  022700  047600          CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
004240  001401                  BEQ     .+4             ;BRANCH IF OK
004242  104000                  HLT                     ;FPS NOT EQUAL TO 47600

004244  174067  174532          STD     0,ANS1
004250  000414                  BR      031

004252  040400  000000  000000  N31:    .FLT4   2
004260  000000
004262  040200  000000  000000  M31:    .FLT4   1
004270  000000
004272  040400  000000  000000  AN31:   .FLT4   2
004300  000000

004302  026767  177764  174472  031:    CMP     AN31,ANS1       ;CHECK LEFT HALF
004310  001401                  BEQ     .+4
004312  104004                  HLT+4                   ;LEFT HALF IS WRONG

004314  026767  177754  174462          CMP     AN31+2,ANS2     ;CHECK LEFT HALF
004322  001401                  BEQ     .+4
004324  104004                  HLT+4                   ;LEFT HALF IS WRONG

004326  026767  177744  174452          CMP     AN31+4,ANS3     ;CHECK RIGHT HALF
004334  001401                  BEQ     .+4
004336  104004                  HLT+4                   ;RIGHT HALF IS WRONG

004340  026767  177734  174442          CMP     AN31+6,ANS4     ;CHECK RIGHT HALF
004346  001401                  BEQ     .+4
004350  104004                  HLT+4                   ;RIGHT HALF IS WRONG
```

```
1155                                    ;*****************************************************************
1156                                    ;TEST 32                TEST OF DIVD FPU INSTRUCTION
1157                                    ;       3 / 1 = 3
1158                                    ;           USING AC0       FPC = 47600     FEC = N/A
1159                                    ;*****************************************************************
1160
1161  004352  104400                            SCOPE
1162  004354  170127  047600                    LDFPS   #47600&57760
1163  004360  172467  000024                    LDD     N32,0           ;LOAD 3 INTO AC0
1164  004364  174467  000030                    DIVD    M32,0           ;DIVIDE 3 BY 1
1165  004370  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
1166  004372  022700  047600                    CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
1167  004376  001401                            BEQ     .+4             ;BRANCH IF OK
1168  004400  104000                            HLT                     ;FPS NOT EQUAL TO 47600
1169
1170  004402  174067  174374                    STD     0,ANS1
1171  004406  000414                            BR      032
1172
1173  004410  040500  000000  000000  N32:      .FLT4   3
1174  004416  000000
1175  004420  040200  000000  000000  M32:      .FLT4   1
1176  004426  000000
1177  004430  040500  000000  000000  AN32:     .FLT4   3
1178  004436  000000
1179
1180  004440  026767  177764  174334  032:      CMP     AN32,ANS1       ;CHECK LEFT HALF
1181  004446  001401                            BEQ     .+4
1182  004450  104004                            HLT+4                   ;LEFT HALF IS WRONG
1183
1184  004452  026767  177754  174324            CMP     AN32+2,ANS2     ;CHECK LEFT HALF
1185  004460  001401                            BEQ     .+4
1186  004462  104004                            HLT+4                   ;LEFT HALF IS WRONG
1187
1188  004464  026767  177744  174314            CMP     AN32+4,ANS3     ;CHECK RIGHT HALF
1189  004472  001401                            BEQ     .+4
1190  004474  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1191
1192  004476  026767  177734  174304            CMP     AN32+6,ANS4     ;CHECK RIGHT HALF
1193  004504  001401                            BEQ     .+4
1194  004506  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1195
```

```
                                ;*****************************************************************
                                ;TEST 33                TEST OF DIVD FPU INSTRUCTION
                                ;    4 / 1 = 4
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

004510  104400                          SCOPE
004512  170127  047600                  LDFPS   #47600&57760
004516  172467  000024                  LDD     N33,0           ;LOAD 4 INTO AC0
004522  174467  000030                  DIVD    M33,0           ;DIVIDE 4 BY 1
004526  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
004530  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
004534  001401                          BEQ     .+4             ;BRANCH IF OK
004536  104000                          HLT                     ;FPS NOT EQUAL TO 47600

004540  174067  174236                  STD     0,ANS1
004544  000414                          BR      033

004546  040600  000000  000000  N33:    .FLT4   4
004554  000000
004556  040200  000000  000000  M33:    .FLT4   1
004564  000000
004566  040600  000000  000000  AN33:   .FLT4   4
004574  000000

004576  026767  177764  174176  033:    CMP     AN33,ANS1       ;CHECK LEFT HALF
004604  001401                          BEQ     .+4
004606  104004                          HLT+4                   ;LEFT HALF IS WRONG

004610  026767  177754  174166          CMP     AN33+2,ANS2     ;CHECK LEFT HALF
004616  001401                          BEQ     .+4
004620  104004                          HLT+4                   ;LEFT HALF IS WRONG

004622  026767  177744  174156          CMP     AN33+4,ANS3     ;CHECK RIGHT HALF
004630  001401                          BEQ     .+4
004632  104004                          HLT+4                   ;RIGHT HALF IS WRONG

004634  026767  177734  174146          CMP     AN33+6,ANS4     ;CHECK RIGHT HALF
004642  001401                          BEQ     .+4
004644  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************
                        ;TEST 34                TEST OF DIVD FPU INSTRUCTION
                        ;     5 / 1 = 5
                        ;     USING ACC         FPC = 47600     FEC = N/A
                        ;*****************************************************

004644  104400                  SCOPE
004646  170127 047600           LDFPS   #47600&57760
004652  172467 000034           LDD     N34,0           ;LOAD 5 INTO ACC
004656  174467 000030           DIVD    M34,0           ;DIVIDE 5 BY 1
004662  170267 000022           STFPS   FPS             ;STORE FLOATING POINT STATUS
004666  033727 000022 047600     BIT     #47600,FPS      ;CHECK FLOATING POINT STATUS
004674  001401                  BEQ     .+4             ;BRANCH IF OK
004676  104000                  HLT                     ;FPS NOT EQUAL TO 47600

004700  174267 174100           STD     ,ANS1
004704  000414                  BR      C34

004706  040640 000000 000000 N34:      .FLT4   5
004714  000000
004716  040200 000000 000000 M34:      .FLT4   1
004724  000000
004726  040640 000000 000000 AN34:     .FLT4   5
004734  000000

004736  026767 177764 174040 C34:      CMP     AN34,ANS1       ;CHECK LEFT HALF
004744  001401                  BEQ     .+4
004746  104004                  HLT+4                   ;LEFT HALF IS WRONG

004750  026767 177754 174030          CMP     AN34+2,ANS2     ;CHECK LEFT HALF
004756  001401                  BEQ     .+4
004760  104004                  HLT+4                   ;LEFT HALF IS WRONG

004762  026767 177744 174020          CMP     AN34+4,ANS3     ;CHECK RIGHT HALF
004770  001401                  BEQ     .+4
004772  104004                  HLT+4                   ;RIGHT HALF IS WRONG

004774  026767 177734 174010          CMP     AN34+6,ANS4     ;CHECK RIGHT HALF
005002  001401                  BEQ     .+4
005004  104004                  HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 35                TEST OF DIVD FPU INSTRUCTION
                                ;    6 / 1 = 6
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005004  104400                          SCOPE
005006  170127  047600                  LDFPS   #47600&57760
005012  172467  000024                  LDD     N35,0           ;LOAD 6 INTO AC0
005016  174467  000030                  DIVD    M35,0           ;DIVIDE 6 BY 1
005022  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005024  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005030  001401                          BEQ     .+4             ;BRANCH IF OK
005032  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005034  174067  173742                  STD     0,ANS1
005040  000414                          BR      C35

005042  040700  000000  000000  N35:    .FLT4   6
005050  000000
005052  040200  000000  000000  M35:    .FLT4   1
005060  000000
005062  040700  000000  000000  AN35:   .FLT4   6
005070  000000

005072  026767  177764  173702  C35:    CMP     AN35,ANS1       ;CHECK LEFT HALF
005100  001401                          BEQ     .+4
005102  104004                          HLT+4                   ;LEFT HALF IS WRONG

005104  026767  177754  173672          CMP     AN35+2,ANS2     ;CHECK LEFT HALF
005112  001401                          BEQ     .+4
005114  104004                          HLT+4                   ;LEFT HALF IS WRONG

005116  026767  177744  173662          CMP     AN35+4,ANS3     ;CHECK RIGHT HALF
005124  001401                          BEQ     .+4
005126  104004                          HLT+4                   ;RIGHT HALF IS WRONG

005130  026767  177734  173652          CMP     AN35+6,ANS4     ;CHECK RIGHT HALF
005136  001401                          BEQ     .+4
005140  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 36                TEST OF DIVD FPU INSTRUCTION
                                ;    7 / 1 = 7
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005142  104400                          SCOPE
005144  170127  047600                  LDFPS   #47600!57760
005150  172467  000024                  LDD     N36,0           ;LOAD 7 INTO AC0
005154  174467  000030                  DIVD    M36,0           ;DIVIDE 7 BY 1
005160  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005162  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005166  001401                          BEQ     .+4             ;BRANCH IF OK
005170  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005172  174067  173604                  STD     0,ANS1
005176  000414                          BR      C36

005200  040740  000000  000000  N36:    .FLT4   7
005206  000000
005210  040200  000000  000000  M36:    .FLT4   1
005216  000000
005220  040740  000000  000000  AN36:   .FLT4   7
005226  000000

005230  026767  177764  173544  C36:    CMP     AN36,ANS1       ;CHECK LEFT HALF
005236  001401                          BEQ     .+4
005240  104004                          HLT+4                   ;LEFT HALF IS WRONG

005242  026767  177754  173534          CMP     AN36+2,ANS2     ;CHECK LEFT HALF
005250  001401                          BEQ     .+4
005252  104004                          HLT+4                   ;LEFT HALF IS WRONG

005254  026767  177744  173524          CMP     AN36+4,ANS3     ;CHECK RIGHT HALF
005262  001401                          BEQ     .+4
005264  104004                          HLT+4                   ;RIGHT HALF IS WRONG

005266  026767  177734  173514          CMP     AN36+6,ANS4     ;CHECK RIGHT HALF
005274  001401                          BEQ     .+4
005276  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 37                TEST OF DIVD FPU INSTRUCTION
                                ;    7 / 2 = 3.5
                                ;       USING AC2       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005300  104400                          SCOPE
005302  170127  047600                  LDFPS   #47600&57760
005306  172667  000024                  LDD     N37,2           ;LOAD 7 INTO AC2
005312  174667  000030                  DIVD    M37,2           ;DIVIDE 7 BY 2
005316  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005320  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005324  001401                          BEQ     .+4             ;BRANCH IF OK
005326  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005330  174267  173446                  STD     2,ANS1
005334  000414                          BR      037

005336  040740  000000  000000  N37:    .FLT4   7
005344  000000
005346  040400  000000  000000  M37:    .FLT4   2
005354  000000
005356  040540  000000  000000  AN37:   .FLT4   3.5
005364  000000

005366  026767  177764  173406  037:    CMP     AN37,ANS1       ;CHECK LEFT HALF
005374  001401                          BEQ     .+4
005376  104004                          HLT+4                   ;LEFT HALF IS WRONG

005400  026767  177754  173376          CMP     AN37+2,ANS2     ;CHECK LEFT HALF
005406  001401                          BEQ     .+4
005410  104004                          HLT+4                   ;LEFT HALF IS WRONG

005412  026767  177744  173366          CMP     AN37+4,ANS3     ;CHECK RIGHT HALF
005420  001401                          BEQ     .+4
005422  104004                          HLT+4                   ;RIGHT HALF IS WRONG

005424  026767  177734  173356          CMP     AN37+6,ANS4     ;CHECK RIGHT HALF
005432  001401                          BEQ     .+4
005434  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 40                TEST OF DIVD FPU INSTRUCTION
                                ;    7 / 3 = 2.333333333333333333333333
                                ;       USING AC1       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005436  104400                          SCOPE
005440  170127  047600                  LDFPS   #47600&57760
005444  172567  000024                  LDD     N40,1           ;LOAD 7 INTO AC1
005450  174567  000030                  DIVD    M40,1           ;DIVIDE 7 BY 3
005454  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005456  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005462  001401                          BEQ     .+4             ;BRANCH IF OK
005464  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005466  174167  173310                  STD     1,ANS1
005472  000414                          BR      C40

005474  040740  000000  000000  N40:    .FLT4   7
005502  000000
005504  040500  000000  000000  M40:    .FLT4   3
005512  000000
005514  040425  052525  052525  AN40:   .FLT4   2.333333333333333333333333
005522  052525

005524  026767  177764  173250  C40:    CMP     AN40,ANS1       ;CHECK LEFT HALF
005532  001401                          BEQ     .+4
005534  104004                          HLT+4                   ;LEFT HALF IS WRONG

005536  026767  177754  173240          CMP     AN40+2,ANS2     ;CHECK LEFT HALF
005544  001401                          BEQ     .+4
005546  104004                          HLT+4                   ;LEFT HALF IS WRONG

005550  026767  177744  173230          CMP     AN40+4,ANS3     ;CHECK RIGHT HALF
005556  001401                          BEQ     .+4
005560  104004                          HLT+4                   ;RIGHT HALF IS WRONG

005562  026767  177734  173220          CMP     AN40+6,ANS4     ;CHECK RIGHT HALF
005570  001401                          BEQ     .+4
005572  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 41                TEST OF DIVD FPU INSTRUCTION
                                ;     7 / 4 = 1.75
                                ;       USING AC1       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005574  104400                          SCOPE
005576  170127  047600                  LDFPS   #47600&57760
005602  172567  000024                  LDD     N41,1           ;LOAD 7 INTO AC1
005606  174567  000030                  DIVD    M41,1           ;DIVIDE 7 BY 4
005612  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005614  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005620  001401                          BEQ     .+4             ;BRANCH IF OK
005622  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005624  174167  173152                  STD     1,ANS1
005630  000414                          BR      041

005632  040740  000000  000000  N41:    .FLT4   7
005640  000000
005642  040600  000000  000000  M41:    .FLT4   4
005650  000000
005652  040340  000000  000000  AN41:   .FLT4   1.75
005660  000000

005662  026767  177764  173112  041:    CMP     AN41,ANS1      ;CHECK LEFT HALF
005670  001401                          BEQ     .+4
005672  104004                          HLT+4                  ;LEFT HALF IS WRONG

005674  026767  177754  173102          CMP     AN41+2,ANS2    ;CHECK LEFT HALF
005702  001401                          BEQ     .+4
005704  104004                          HLT+4                  ;LEFT HALF IS WRONG

005706  026767  177744  173072          CMP     AN41+4,ANS3    ;CHECK RIGHT HALF
005714  001401                          BEQ     .+4
005716  104004                          HLT+4                  ;RIGHT HALF IS WRONG

005720  026767  177734  173062          CMP     AN41+6,ANS4    ;CHECK RIGHT HALF
005726  001401                          BEQ     .+4
005730  104004                          HLT+4                  ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 42                TEST OF DIVD FPU INSTRUCTION
                                ;    7 / 5 = 1.4
                                ;       USING AC3       FPC = 47600     FEC = N/A
                                ;*****************************************************************

005732  104400                          SCOPE
005734  170127  047600                  LDFPS   #47600&57760
005740  172767  000024                  LDD     N42,3           ;LOAD 7 INTO AC3
005744  174767  000030                  DIVD    M42,3           ;DIVIDE 7 BY 5
005750  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
005752  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
005756  001401                          BEQ     .+4             ;BRANCH IF OK
005760  104000                          HLT                     ;FPS NOT EQUAL TO 47600

005762  174367  173014                  STD     3,ANS1
005766  000414                          BR      042

005770  040740  000000  000000  N42:    .FLT4   7
005776  000000
006000  040640  000000  000000  M42:    .FLT4   5
006006  000000
006010  040263  031463  031463  AN42:   .FLT4   1.4
006016  031463

006020  026767  177764  172754  042:    CMP     AN42,ANS1       ;CHECK LEFT HALF
006026  001401                          BEQ     .+4
006030  104004                          HLT+4                   ;LEFT HALF IS WRONG

006032  026767  177754  172744          CMP     AN42+2,ANS2     ;CHECK LEFT HALF
006040  001401                          BEQ     .+4
006042  104004                          HLT+4                   ;LEFT HALF IS WRONG

006044  026767  177744  172734          CMP     AN42+4,ANS3     ;CHECK RIGHT HALF
006052  001401                          BEQ     .+4
006054  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006056  026767  177734  172724          CMP     AN42+6,ANS4     ;CHECK RIGHT HALF
006064  001401                          BEQ     .+4
006066  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 43                TEST OF DIVD FPU INSTRUCTION
                                ;       7 / 6 = 1.1666666666666666667
                                ;       USING AC1       FPC = 47600     FEC = N/A
                                ;*****************************************************************

006070  104400                          SCOPE
006072  170127  047600                  LDFPS   #47600&57760
006076  172567  000024                  LDD     N43,1           ;LOAD 7 INTO AC1
006102  174567  000030                  DIVD    M43,1           ;DIVIDE 7 BY 6
006106  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006110  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
006114  001401                          BEQ     .+4             ;BRANCH IF OK
006116  104000                          HLT                     ;FPS NOT EQUAL TO 47600

006120  174167  172656                  STD     1,ANS1
006124  000414                          BR      043

006126  040740  000000  000000  N43:    .FLT4   7
006134  000000
006136  040700  000000  000000  M43:    .FLT4   6
006144  000000
006146  040225  052525  052525  AN43:   .FLT4   1.1666666666666666667
006154  052525

006156  026767  177764  172616  043:    CMP     AN43,ANS1       ;CHECK LEFT HALF
006164  001401                          BEQ     .+4
006166  104004                          HLT+4                   ;LEFT HALF IS WRONG

006170  026767  177754  172606          CMP     AN43+2,ANS2     ;CHECK LEFT HALF
006176  001401                          BEQ     .+4
006200  104004                          HLT+4                   ;LEFT HALF IS WRONG

006202  026767  177744  172576          CMP     AN43+4,ANS3     ;CHECK RIGHT HALF
006210  001401                          BEQ     .+4
006212  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006214  026767  177734  172566          CMP     AN43+6,ANS4     ;CHECK RIGHT HALF
006222  001401                          BEQ     .+4
006224  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 44                TEST OF DIVD FPU INSTRUCTION
                                ;    6 / 5 = 1.2
                                ;       USING AC1       FPC = 47600     FEC = N/A
                                ;*****************************************************************

006226  104400                          SCOPE
006230  170127  047600                  LDFPS   #47600&57760
006234  172567  000024                  LDD     N44,1           ;LOAD 6 INTO AC1
006240  174567  000030                  DIVD    M44,1           ;DIVIDE 6 BY 5
006244  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006246  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
006252  001401                          BEQ     .+4             ;BRANCH IF OK
006254  104000                          HLT                     ;FPS NOT EQUAL TO 47600

006256  174167  172520                  STD     1,ANS1
006262  000414                          BR      C44

006264  040700  000000  000000  N44:    .FLT4   6
006272  000000
006274  040640  000000  000000  M44:    .FLT4   5
006302  000000
006304  040231  114631  114631  AN44:   .FLT4   1.2
006312  114632

006314  026767  177764  172460  C44:    CMP     AN44,ANS1       ;CHECK LEFT HALF
006322  001401                          BEQ     .+4
006324  104004                          HLT+4                   ;LEFT HALF IS WRONG

006326  026767  177754  172450          CMP     AN44+2,ANS2     ;CHECK LEFT HALF
006334  001401                          BEQ     .+4
006336  104004                          HLT+4                   ;LEFT HALF IS WRONG

006340  026767  177744  172440          CMP     AN44+4,ANS3     ;CHECK RIGHT HALF
006346  001401                          BEQ     .+4
006350  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006352  026767  177734  172430          CMP     AN44+6,ANS4     ;CHECK RIGHT HALF
006360  001401                          BEQ     .+4
006362  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 45                TEST OF DIVD FPU INSTRUCTION
                                ;    6 / 4 = 1.5
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

006364  104400                          SCOPE
006366  170127  047600                  LDFPS   #47600&57760
006372  172467  000024                  LDD     N45,0           ;LOAD 6 INTO AC0
006376  174467  000030                  DIVD    M45,0           ;DIVIDE 6 BY 4
006402  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
006404  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
006410  001401                          BEQ     .+4             ;BRANCH IF OK
006412  104000                          HLT                     ;FPS NOT EQUAL TO 47600

006414  174067  172362                  STD     0,ANS1
006420  000414                          BR      045

006422  040700  000000  000000  N45:    .FLT4   6
006430  000000
006432  040600  000000  000000  M45:    .FLT4   4
006440  000000
006442  040300  000000  000000  AN45:   .FLT4   1.5
006450  000000

006452  026767  177764  172322  045:    CMP     AN45,ANS1       ;CHECK LEFT HALF
006460  001401                          BEQ     .+4
006462  104004                          HLT+4                   ;LEFT HALF IS WRONG

006464  026767  177754  172312          CMP     AN45+2,ANS2     ;CHECK LEFT HALF
006472  001401                          BEQ     .+4
006474  104004                          HLT+4                   ;LEFT HALF IS WRONG

006476  026767  177744  172302          CMP     AN45+4,ANS3     ;CHECK RIGHT HALF
006504  001401                          BEQ     .+4
006506  104004                          HLT+4                   ;RIGHT HALF IS WRONG

006510  026767  177734  172272          CMP     AN45+6,ANS4     ;CHECK RIGHT HALF
006516  001401                          BEQ     .+4
006520  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
1647                                    ;*****************************************************************
1648                                    ;TEST 46                TEST OF DIVD FPU INSTRUCTION
1649                                    ;    6 / 3 = 2
1650                                    ;       USING AC0       FPC = 47600     FEC = N/A
1651                                    ;*****************************************************************
1652
1653  006522  104400                            SCOPE
1654  006524  170127  047600                    LDFPS   #47600&57760
1655  006530  172467  000024                    LDD     N46,0           ;LOAD 6 INTO AC0
1656  006534  174467  000030                    DIVD    M46,0           ;DIVIDE 6 BY 3
1657  006540  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
1658  006542  022700  047600                    CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
1659  006546  001401                            BEQ     .+4             ;BRANCH IF OK
1660  006550  104000                            HLT                     ;FPS NOT EQUAL TO 47600
1661
1662  006552  174067  172224                    STD     0,ANS1
1663  006556  000414                            BR      046
1664
1665  006560  040700  000000  000000  N46:      .FLT4   6
1666  006566  000000
1667  006570  040500  000000  000000  M46:      .FLT4   3
1668  006576  000000
1669  006600  040400  000000  000000  AN46:     .FLT4   2
1670  006606  000000
1671
1672  006610  026767  177764  172164  046:      CMP     AN46,ANS1       ;CHECK LEFT HALF
1673  006616  001401                            BEQ     .+4
1674  006620  104004                            HLT+4                   ;LEFT HALF IS WRONG
1675
1676  006622  026767  177754  172154            CMP     AN46+2,ANS2     ;CHECK LEFT HALF
1677  006630  001401                            BEQ     .+4
1678  006632  104004                            HLT+4                   ;LEFT HALF IS WRONG
1679
1680  006634  026767  177744  172144            CMP     AN46+4,ANS3     ;CHECK RIGHT HALF
1681  006642  001401                            BEQ     .+4
1682  006644  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1683
1684  006646  026767  177734  172134            CMP     AN46+6,ANS4     ;CHECK RIGHT HALF
1685  006654  001401                            BEQ     .+4
1686  006656  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1687
```

```
1688                                    ;*****************************************************************
1689                                    ;TEST 47                TEST OF DIVD FPU INSTRUCTION
1690                                    ;    6 / 2 = 3
1691                                    ;       USING AC0       FPC = 47600     FEC = N/A
1692                                    ;*****************************************************************
1693
1694  006660  104400                            SCOPE
1695  006662  170127  047600                    LDFPS   #47600&57760
1696  006666  172467  000024                    LDD     N47,0           ;LOAD 6 INTO AC0
1697  006672  174467  000030                    DIVD    M47,0           ;DIVIDE 6 BY 2
1698  006676  170200                            STFPS   FPS             ;STORE FLOATING POINT STATUS
1699  006700  022700  047600                    CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
1700  006704  001401                            BEQ     .+4             ;BRANCH IF OK
1701  006706  104000                            HLT                     ;FPS NOT EQUAL TO 47600
1702
1703  006710  174067  172066                    STD     0,ANS1
1704  006714  000414                            BR      047
1705
1706  006716  040700  000000  000000  N47:      .FLT4   6
1707  006724  000000
1708  006726  040400  000000  000000  M47:      .FLT4   2
1709  006734  000000
1710  006736  040500  000000  000000  AN47:     .FLT4   3
1711  006744  000000
1712
1713  006746  026767  177764  172026  047:      CMP     AN47,ANS1       ;CHECK LEFT HALF
1714  006754  001401                            BEQ     .+4
1715  006756  104004                            HLT+4                   ;LEFT HALF IS WRONG
1716
1717  006760  026767  177754  172016            CMP     AN47+2,ANS2     ;CHECK LEFT HALF
1718  006766  001401                            BEQ     .+4
1719  006770  104004                            HLT+4                   ;LEFT HALF IS WRONG
1720
1721  006772  026767  177744  172006            CMP     AN47+4,ANS3     ;CHECK RIGHT HALF
1722  007000  001401                            BEQ     .+4
1723  007002  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1724
1725  007004  026767  177734  171776            CMP     AN47+6,ANS4     ;CHECK RIGHT HALF
1726  007012  001401                            BEQ     .+4
1727  007014  104004                            HLT+4                   ;RIGHT HALF IS WRONG
1728
```

```
                                ;*****************************************************************
                                ;TEST 50                TEST OF DIVD FPU INSTRUCTION
                                ;       5 / 2 = 2.5
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

007016  104400                          SCOPE
007020  170127  047600                  LDFPS   #47600&57760
007024  172467  000024                  LDD     N50,0           ;LOAD 5 INTO AC0
007030  174467  000030                  DIVD    M50,0           ;DIVIDE 5 BY 2
007034  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007036  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007042  001401                          BEQ     .+4             ;BRANCH IF OK
007044  104000                          HLT                     ;FPS NOT EQUAL TO 47600

007046  174067  171730                  STD     0,ANS1
007052  000414                          BR      050

007054  040640  000000  000000  N50:    .FLT4   5
007062  000000
007064  040400  000000  000000  M50:    .FLT4   2
007072  000000
007074  040440  000000  000000  AN50:   .FLT4   2.5
007102  000000

007104  026767  177764  171670  050:    CMP     AN50,ANS1       ;CHECK LEFT HALF
007112  001401                          BEQ     .+4
007114  104004                          HLT+4                   ;LEFT HALF IS WRONG

007116  026767  177754  171660          CMP     AN50+2,ANS2     ;CHECK LEFT HALF
007124  001401                          BEQ     .+4
007126  104004                          HLT+4                   ;LEFT HALF IS WRONG

007130  026767  177744  171650          CMP     AN50+4,ANS3     ;CHECK RIGHT HALF
007136  001401                          BEQ     .+4
007140  104004                          HLT+4                   ;RIGHT HALF IS WRONG

007142  026767  177734  171640          CMP     AN50+6,ANS4     ;CHECK RIGHT HALF
007150  001401                          BEQ     .+4
007152  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST_51:                TEST OF DIVD FPU INSTRUCTION
                                ;     5 / 3 = 1.6666666666666666667
                                ;     USING ACO          FPC = 47600     FEC = N/A
                                ;*****************************************************************

007164  104400                          SCOPE
007166  170106  047600                  LDFPS   #47600857760
007172  172406  000005                  LDD     NS1,0           ;LOAD 5 INTO ACO
007176  174406  000030                  DIVD    MS1,0           ;DIVIDE 5 BY 3
007202  170206                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007204  027706  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007210  001401                          BEQ     .+4             ;BRANCH IF OK
007212  104000                          HLT                     ;FPS NOT EQUAL TO 47600

007214  174067  171572                  STD     0,ANS1
007220  000414                          BR      CS1

007222  040640  000000  000000  NS1:    .FLT4   5
007230  000000
007232  040500  000000  000000  MS1:    .FLT4   3
007240  000000
007242  040326  052525  052525  ANS1:   .FLT4   1.6666666666666666667
007250  052526

007252  026767  177764  171532  CS1:    CMP     ANS1,ANS1       ;CHECK LEFT HALF
007260  001401                          BEQ     .+4
007262  104004                          HLT+4                   ;LEFT HALF IS WRONG

007264  026767  177754  171522          CMP     ANS1+2,ANS2     ;CHECK LEFT HALF
007272  001401                          BEQ     .+4
007274  104004                          HLT+4                   ;LEFT HALF IS WRONG

007276  026767  177744  171512          CMP     ANS1+4,ANS3     ;CHECK RIGHT HALF
007304  001401                          BEQ     .+4
007306  104004                          HLT+4                   ;RIGHT HALF IS WRONG

007310  026767  177734  171502          CMP     ANS1+6,ANS4     ;CHECK RIGHT HALF
007316  001401                          BEQ     .+4
007320  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 52                TEST OF DIVD FPU INSTRUCTION
                                ;       5 / 4 = 1.25
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

007312  104400                          SCOPE
007314  170127  047600                  LDFPS   #47600&57760
007320  172467  000024                  LDD     N52,0           ;LOAD 5 INTO AC0
007324  174467  000030                  DIVD    M52,0           ;DIVIDE 5 BY 4
007330  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007332  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007336  001401                          BEQ     .+4             ;BRANCH IF OK
007340  104000                          HLT                     ;FPS NOT EQUAL TO 47600

007342  174067  171434                  STD     0,ANS1
007346  000414                          BR      052

007350  040640  000000  000000  N52:    .FLT4   5
007356  000000
007360  040600  000000  000000  M52:    .FLT4   4
007366  000000
007370  040240  000000  000000  AN52:   .FLT4   1.25
007376  000000

007400  026767  177764  171374  052:    CMP     AN52,ANS1       ;CHECK LEFT HALF
007406  001401                          BEQ     .+4
007410  104004                          HLT+4                   ;LEFT HALF IS WRONG

007412  026767  177754  171364          CMP     AN52+2,ANS2     ;CHECK LEFT HALF
007420  001401                          BEQ     .+4
007422  104004                          HLT+4                   ;LEFT HALF IS WRONG

007424  026767  177744  171354          CMP     AN52+4,ANS3     ;CHECK RIGHT HALF
007432  001401                          BEQ     .+4
007434  104004                          HLT+4                   ;RIGHT HALF IS WRONG

007436  026767  177734  171344          CMP     AN52+6,ANS4     ;CHECK RIGHT HALF
007444  001401                          BEQ     .+4
007446  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                        ;*****************************************************************
                        ;TEST 53                TEST OF DIVD FPU INSTRUCTION
                        ;    4 / 3 = 1.33333333333333333333
                        ;       USING AC0       FPC = 47600     FEC = N/A
                        ;*****************************************************************

007450  104400                    SCOPE
007452  170127  047600            LDFPS   #47600$57760
007456  172467  000024            LDD     N53,0           ;LOAD 4 INTO AC0
007462  174467  000030            DIVD    M53,0           ;DIVIDE 4 BY 3
007466  170200                    STFPS   FPS             ;STORE FLOATING POINT STATUS
007470  022700  047600            CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007474  001401                    BEQ     .+4             ;BRANCH IF OK
007476  104000                    HLT                     ;FPS NOT EQUAL TO 47600

007500  174067  171276            STD     0,ANS1
007504  000414                    BR      053

007506  040600  000000  000000  N53:    .FLT4   4
007514  000000
007516  040500  000000  000000  M53:    .FLT4   3
007524  000000
007526  040252  125252  125252  ANS3:   .FLT4   1.33333333333333333333
007534  125253

007536  026767  177764  171236  053:    CMP     AN53,ANS1       ;CHECK LEFT HALF
007544  001401                    BEQ     .+4
007546  104004                    HLT+4                   ;LEFT HALF IS WRONG

007550  026767  177754  171226          CMP     AN53+2,ANS2     ;CHECK LEFT HALF
007556  001401                    BEQ     .+4
007560  104004                    HLT+4                   ;LEFT HALF IS WRONG

007562  026767  177744  171216          CMP     AN53+4,ANS3     ;CHECK RIGHT HALF
007570  001401                    BEQ     .+4
007572  104004                    HLT+4                   ;RIGHT HALF IS WRONG

007574  026767  177734  171206          CMP     AN53+6,ANS4     ;CHECK RIGHT HALF
007602  001401                    BEQ     .+4
007604  104004                    HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 54                TEST OF DIVD FPU INSTRUCTION
                                ;    3 / 2 = 1.5
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

007606  104400                          SCOPE
007610  170127  047600                  LDFPS   #47600&57760
007614  172467  000024                  LDD     N54,0           ;LOAD 3 INTO AC0
007620  174467  000030                  DIVD    M54,0           ;DIVIDE 3 BY 2
007624  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007626  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007632  001401                          BEQ     .+4             ;BRANCH IF OK
007634  104000                          HLT                     ;FPS NOT EQUAL TO 47600

007636  174067  171140                  STD     0,ANS1
007642  000414                          BR      C54

007644  040500  000000  000000  N54:    .FLT4   3
007652  000000
007654  040400  000000  000000  M54:    .FLT4   2
007662  000000
007664  040300  000000  000000  AN54:   .FLT4   1.5
007672  000000

007674  026767  177764  171100  C54:    CMP     AN54,ANS1       ;CHECK LEFT HALF
007702  001401                          BEQ     .+4
007704  104004                          HLT+4                   ;LEFT HALF IS WRONG

007706  026767  177754  171070          CMP     AN54+2,ANS2     ;CHECK LEFT HALF
007714  001401                          BEQ     .+4
007716  104004                          HLT+4                   ;LEFT HALF IS WRONG

007720  026767  177744  171060          CMP     AN54+4,ANS3     ;CHECK RIGHT HALF
007726  001401                          BEQ     .+4
007730  104004                          HLT+4                   ;RIGHT HALF IS WRONG

007732  026767  177734  171050          CMP     AN54+6,ANS4     ;CHECK RIGHT HALF
007740  001401                          BEQ     .+4
007742  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 55                TEST OF DIVD FPU INSTRUCTION
                                ;    .6666666666666666667 / 2 = .333333333333333333
                                ;      USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

007744  104400                          SCOPE
007746  170127  047600                  LDFPS   #47600&57760
007752  172467  000024                  LDD     N55,0           ;LOAD .6666666666666666667 INTO AC0
007756  174467  000030                  DIVD    M55,0           ;DIVIDE .6666666666666666667 BY 2
007762  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
007764  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
007770  001401                          BEQ     .+4             ;BRANCH IF OK
007772  104000                          HLT                     ;FPS NOT EQUAL TO 47600

007774  174067  171002                  STD     0,ANS1
010000  000414                          BR      C55

010002  040052  125252  125252  N55:    .FLT4   .6666666666666666667
010010  125253
010012  040400  000000  000000  M55:    .FLT4   2
010020  000000
010022  037652  125252  125252  AN55:   .FLT4   .333333333333333333
010030  125253

010032  026767  177764  170742  C55:    CMP     AN55,ANS1       ;CHECK LEFT HALF
010040  001401                          BEQ     .+4
010042  104004                          HLT+4                   ;LEFT HALF IS WRONG

010044  026767  177754  170732          CMP     AN55+2,ANS2     ;CHECK LEFT HALF
010052  001401                          BEQ     .+4
010054  104004                          HLT+4                   ;LEFT HALF IS WRONG

010056  026767  177744  170722          CMP     AN55+4,ANS3     ;CHECK RIGHT HALF
010064  001401                          BEQ     .+4
010066  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010070  026767  177734  170712          CMP     AN55+6,ANS4     ;CHECK RIGHT HALF
010076  001401                          BEQ     .+4
010100  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 56                TEST OF DIVD FPU INSTRUCTION
                                ;    1.3333333333333333 / 2 = .6666666666666666667
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

010102  104400                          SCOPE
010104  170127  047600                  LDFPS   #47600&57760
010110  172467  000024                  LDD     N56,0           ;LOAD 1.3333333333333333 INTO AC0
010114  174467  000030                  DIVD    M56,0           ;DIVIDE 1.3333333333333333 BY 2
010120  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010122  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
010126  001401                          BEQ     .+4             ;BRANCH IF OK
010130  104000                          HLT                     ;FPS NOT EQUAL TO 47600

010132  174067  170644                  STD     0,ANS1
010136  000414                          BR      056

010140  040252  125252  125252  N56:    .FLT4   1.33333333333333333
010146  125253
010150  040400  000000  000000  M56:    .FLT4   2
010156  000000
010160  040052  125252  125252  AN56:   .FLT4   .6666666666666666667
010166  125253

010170  026767  177764  170604  056:    CMP     AN56,ANS1       ;CHECK LEFT HALF
010176  001401                          BEQ     .+4
010200  104004                          HLT+4                   ;LEFT HALF IS WRONG

010202  026767  177754  170574          CMP     AN56+2,ANS2     ;CHECK LEFT HALF
010210  001401                          BEQ     .+4
010212  104004                          HLT+4                   ;LEFT HALF IS WRONG

010214  026767  177744  170564          CMP     AN56+4,ANS3     ;CHECK RIGHT HALF
010222  001401                          BEQ     .+4
010224  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010226  026767  177734  170554          CMP     AN56+6,ANS4     ;CHECK RIGHT HALF
010234  001401                          BEQ     .+4
010236  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 57                TEST OF DIVD FPU INSTRUCTION
                                ;    .3333333333333333 / .5 = .6666666666666667
                                ;       USING AC0       FPC = 47600     FEC = N/A
                                ;*****************************************************************

010240  104400                          SCOPE
010242  170127  047600                  LDFPS   #47600
                                                ;57600=57760
010246  172467  000024                  LDD     N57,0           ;LOAD .3333333333333333 INTO AC0
010252  174467  000030                  DIVD    M57,0           ;DIVIDE .3333333333333333 BY .5
010256  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010260  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
010264  001401                          BEQ     .+4             ;BRANCH IF OK
010266  104000                          HLT                     ;FPS NOT EQUAL TO 47600

010270  174067  170506                  STD     0,ANS1
010274  000414                          BR      C57

010276  037652  125252  125252  N57:    .FLT4   .3333333333333333
010304  125253
010306  040000  000000  000000  M57:    .FLT4   .5
010314  000000
010316  040052  125252  125252  AN57:   .FLT4   .6666666666666667
010324  125253

010326  026767  177764  170446  C57:    CMP     AN57,ANS1       ;CHECK LEFT HALF
010334  001401                          BEQ     .+4
010336  104004                          HLT+4                   ;LEFT HALF IS WRONG

010340  026767  177754  170436          CMP     AN57+2,ANS2     ;CHECK LEFT HALF
010346  001401                          BEQ     .+4
010350  104004                          HLT+4                   ;LEFT HALF IS WRONG

010352  026767  177744  170426          CMP     AN57+4,ANS3     ;CHECK RIGHT HALF
010360  001401                          BEQ     .+4
010362  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010364  026767  177734  170416          CMP     AN57+6,ANS4     ;CHECK RIGHT HALF
010372  001401                          BEQ     .+4
010374  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 60                TEST OF DIVD FPU INSTRUCTION
                                ;    .6666666666666667 / .5 = 1.3333333333333333
                                ;      USING AC0        FPC = 47600     FEC = N/A
                                ;*****************************************************************

010376  104400                          SCOPE
010400  170127  047600                  LDFPS   #47600&57760
010404  172467  000024                  LDD     N60,0           ;LOAD .6666666666666667 INTO AC0
010410  174467  000030                  DIVD    M60,0           ;DIVIDE .6666666666666667 BY .5
010414  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010416  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
010422  001401                          BEQ     .+4             ;BRANCH IF OK
010424  104000                          HLT                     ;FPS NOT EQUAL TO 47600

010426  174067  170350                  STD     0,ANS1
010432  000414                          BR      060

010434  040052  125252  125252  N60:    .FLT4   .6666666666666667
010442  125253
010444  040000  000000  000000  M60:    .FLT4   .5
010452  000000
010454  040252  125252  125252  AN60:   .FLT4   1.33333333333333333
010462  125253

010464  026767  177764  170310  060:    CMP     AN60,ANS1       ;CHECK LEFT HALF
010472  001401                          BEQ     .+4
010474  104004                          HLT+4                   ;LEFT HALF IS WRONG

010476  026767  177754  170300          CMP     AN60+2,ANS2     ;CHECK LEFT HALF
010504  001401                          BEQ     .+4
010506  104004                          HLT+4                   ;LEFT HALF IS WRONG

010510  026767  177744  170270          CMP     AN60+4,ANS3     ;CHECK RIGHT HALF
010516  001401                          BEQ     .+4
010520  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010522  026767  177734  170260          CMP     AN60+6,ANS4     ;CHECK RIGHT HALF
010530  001401                          BEQ     .+4
010532  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 61                    TEST OF DIVD  AND DIVF
                                ;    .6666666666666666667 / 2 = .33333333333333333
                                ;    USING AC0 & AC4              FPS = 47600  FEC = N/A
                                ;*****************************************************************

010534  104400                          SCOPE
010536  170127  047600                  LDFPS   #47600&57760
010542  172467  000040                  LDD     M61,0           ;LOAD 2 INTO AC0
010546  174004                          STD     0,%4            ;LOAD 2 INTO AC4
010550  172467  000022                  LDD     N61,0           ;LOAD .6666666666666666667 INTO AC0
010554  174404                          DIVD    %4,0            ;DIVIDE .6666666666666666667 BY 2
010556  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010560  022700  047600                  CMP     #47600,FPS      ;CHECK FLOATING POINT STATUS
010564  001401                          BEQ     .+4             ;BRANCH IF OK
010566  104000                          HLT                     ;FPS NOT EQUAL TO 47600

010570  174067  170206                  STD     0,ANS1          ;STORE RESULT
010574  000414                          BR      061

010576  040052  125252  125252  N61:    .FLT4   .6666666666666666667
010604  125253
010606  040400  000000  000000  M61:    .FLT4   2
010614  000000
010616  037652  125252  125252  AN61:   .FLT4   .33333333333333333             ;RESULT = .33333333333333333
010624  125253

010626  026767  177764  170146  061:    CMP     AN61,ANS1       ;CHECK LEFT HALF
010634  001401                          BEQ     .+4
010636  104004                          HLT+4                   ;LEFT HALF IS WRONG

010640  026767  177754  170136          CMP     AN61+2,ANS2     ;CHECK LEFT HALF
010646  001401                          BEQ     .+4
010650  104004                          HLT+4                   ;LEFT HALF IS WRONG

010652  026767  177744  170126          CMP     AN61+4,ANS3     ;CHECK RIGHT HALF
010660  001401                          BEQ     .+4
010662  104004                          HLT+4                   ;RIGHT HALF IS WRONG

010664  026767  177734  170116          CMP     AN61+6,ANS4     ;CHECK RIGHT HALF
010672  001401                          BEQ     .+4
010674  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 62                TEST OF DIVD FPU INSTRUCTION
                                ;    0 / 0 = 0
                                ;       USING AC0       FPC = 147604    FEC = 4
                                ;*****************************************************************

010676  104400                          SCOPE
010700  170127  047600                  LDFPS   #147604&57760
010704  172467  000042                  LDD     N62,0           ;LOAD 0 INTO AC0
010710  174467  000046                  DIVD    M62,0           ;DIVIDE 0 BY 0
010714  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
010716  170367  170100                  STST FEC        ;STORE EXCEPTION CODES
010722  022700  147604                  CMP     #147604,FPS     ;CHECK FLOATING POINT STATUS
010726  001401                          BEQ     .+4             ;BRANCH IF OK
010730  104000                          HLT                     ;FPS NOT EQUAL TO 147604

010732  022767  000004  170062          CMP     #4,     FEC     ;CHECK FLOATING EXCEPTION CODE
010740  001401                          BEQ     .+4             ;BRANCH IF OK
010742  104000                          HLT                     ;FEC NOT EQUAL TO 4

010744  174067  170032                  STD     0,ANS1
010750  000414                          BR      062

010752  000000  000000  000000  N62:    .FLT4   0
010760  000000
010762  000000  000000  000000  M62:    .FLT4   0
010770  000000
010772  000000  000000  000000  AN62:   .FLT4   0
011000  000000

011002  026767  177764  167772  062:    CMP     AN62,ANS1       ;CHECK LEFT HALF
011010  001401                          BEQ     .+4
011012  104004                          HLT+4                   ;LEFT HALF IS WRONG

011014  026767  177754  167762          CMP     AN62+2,ANS2     ;CHECK LEFT HALF
011022  001401                          BEQ     .+4
011024  104004                          HLT+4                   ;LEFT HALF IS WRONG

011026  026767  177744  167752          CMP     AN62+4,ANS3     ;CHECK RIGHT HALF
011034  001401                          BEQ     .+4
011036  104004                          HLT+4                   ;RIGHT HALF IS WRONG

011040  026767  177734  167742          CMP     AN62+6,ANS4     ;CHECK RIGHT HALF
011046  001401                          BEQ     .+4
011050  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
                                ;*****************************************************************
                                ;TEST 63                TEST OF DIVD FPU INSTRUCTION
                                ;    .3333333333333333333 / 0 = .33333333333333333333
                                ;       USING AC0       FPC = 147600    FEC = 4
                                ;*****************************************************************

011052  104400                          SCOPE
011054  170127  047600                  LDFPS   #147600&57760
011060  172467  000042                  LDD     N63,0           ;LOAD .3333333333333333333 INTO AC0
011064  174467  000046                  DIVD    M63,0           ;DIVIDE .3333333333333333333 BY 0
011070  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
011072  170367  167724                  STST FEC        ;STORE EXCEPTION CODES
011076  022700  147600                  CMP     #147600,FPS     ;CHECK FLOATING POINT STATUS
011102  001401                          BEQ     .+4             ;BRANCH IF OK
011104  104000                          HLT                     ;FPS NOT EQUAL TO 147600

011106  022767  000004  167706          CMP     #4,     FEC     ;CHECK FLOATING EXCEPTION CODE
011114  001401                          BEQ     .+4             ;BRANCH IF OK
011116  104000                          HLT                     ;FEC NOT EQUAL TO 4

011120  174067  167656                  STD     0,ANS1
011124  000414                          BR      063

011126  037652  125252  125252  N63:    .FLT4   .3333333333333333333
011134  125253
011136  000000  000000  000000  M63:    .FLT4   0
011144  000000
011146  037652  125252  125252  AN63:   .FLT4   .33333333333333333333
011154  125253

011156  026767  177764  167616  063:    CMP     AN63,ANS1       ;CHECK LEFT HALF
011164  001401                          BEQ     .+4
011166  104004                          HLT+4                   ;LEFT HALF IS WRONG

011170  026767  177754  167606          CMP     AN63+2,ANS2     ;CHECK LEFT HALF
011176  001401                          BEQ     .+4
011200  104004                          HLT+4                   ;LEFT HALF IS WRONG

011202  026767  177744  167576          CMP     AN63+4,ANS3     ;CHECK RIGHT HALF
011210  001401                          BEQ     .+4
011212  104004                          HLT+4                   ;RIGHT HALF IS WRONG

011214  026767  177734  167566          CMP     AN63+6,ANS4     ;CHECK RIGHT HALF
011222  001401                          BEQ     .+4
011224  104004                          HLT+4                   ;RIGHT HALF IS WRONG
```

```
2233  011226  104400                    DONE:   SCOPE
2234  011230  032737  002000  177570            BIT     #SW10,@#SWR     ;RING THE BELL?
2235  011236  001005                            BNE     1$              ;NO!
2236  011240  012767  000007  001242            MOV     #BELL,.TYPE     ;TYPE A BELL
2237  011246  000004  012510                    TYPE    .TYPE
2238  011252  005046                    1$:     CLR     -(6)            ;CLEAR TRACE TRAP
2239  011254  032737  010000  177570            BIT     #SW12,@#SWR     ;RUN WITH TRT?
2240  011262  001010                            BNE     2$
2241  011264  005167  001222                    COM     TRPB
2242  011270  100005                            BPL     2$
2243  011272  052716  000020                    BIS     #20,(6)         ;SET TRACE TRAP
2244  011276  012746  001062                    MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
2245  011302  000412                            BR      YESRT
2246  011304  012746  001062            2$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
2247  011310  013700  000042                    MOV     @#42,R0         ;GET MONITOR ADDRESS
2248  011314  001404                            BEQ     3$              ;IF NONE
2249  011316  004710                            JSR     7,(0)           ;GO TO MONITOR
2250  011320  000240                            NOP
2251  011322  000240                            NOP
2252  011324  000240                            NOP
2253  011326  000002                    3$:     RTI
2254  011330  000002                    YESRT:  RTI                     ;RETURN TO PROGRAM FROM TRAP
2255
2256  011332  032737  000400  177570    .EMT:   BIT     #SW08,@#SWR     ;KILL LDUB OR LOOP ON SPEC. TEST
2257  011340  001404                            BEQ     1$
2258  011342  123767  177570  167430            CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   *SW7-0*
2259  011350  001437                            BEQ     OVER
2260  011352  113703  177570            1$:     MOVB    @#SWR,R3        ;GET UB BITS
2261  011356  170003                            LDUB
2262  011360  032737  040000  177570            BIT     #SW14,@#SWR     ;LOOP ON TEST
2263  011366  001026                            BNE     KIT
2264  011370  032737  004000  177570            BIT     #SW11,@#SWR     ;KILL ITERATIONS
2265  011376  001012                            BNE     SAVLAD
2266  011400  105767  167375                    TSTB    ICNT+1
2267  011404  001404                            BEQ     2$              ;BRANCH IF FIRST
2268  011406  126767  001106  167365            CMPB    TIMES,ICNT+1    ;DONE?
2269  011414  001013                            BNE     KIT             ;BRANCH IF NOT
2270  011416  112767  000001  167355    2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
2271  011424  105267  167350            SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
2272  011430  011667  001060                    MOV     (6),LAD         ;SAVE LOOP ADDRESS
2273  011434  016737  167340  177570            MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
2274  011442  000002                            RTI                     ;RETURN
2275
2276  011444  105267  167331            KIT:    INCB    ICNT+1
2277  011450  016737  167324  177570    OVER:   MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
2278  011456  005767  001032                    TST     LAD             ;FIRST ONE?
2279  011462  001760                            BEQ     SAVLAD
2280  011464  016716  001024                    MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
2281  011470  000002                            RTI                     ;FIXES PS
```

```
2282  011472  032737  002000  177570  .TRP:   BIT     #SW10,@#SWR     ;BELL ON ERROR?
2283  011500  001405                          BEQ     1$              ;NO - SKIP
2284  011502  012767  000007  001000          MOV     #BELL,.TYPE     ;TYPE A BELL
2285  011510  000004  012510                  TYPE    .TYPE
2286  011514  004767  000406          1$:     JSR     PC,ERROR        ;COUNT THE NUMBER OF ERRORS
2287  011520  010446                          MOV     R4,-(6)
2288  011522  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
2289  011530  001072                          BNE     4$
2290  011532  000004  012456                  TYPE    ,RETURN
2291  011536  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
2292  011542  162716  000002                  SUB     #2,(6)
2293  011546  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
2294  011550  004767  000212                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
2295  011554  000004  012464                  TYPE    ,SPACE+3
2296  011560  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
2297  011562  004767  000200                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
2298  011566  000004  012465                  TYPE    ,SPACE+4
2299  011572  012703  001002                  MOV     #ANS1,R3        ;ADDRESS OF DATA
2300  011576  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
2301  011600  001426                          BEQ     3$
2302  011602  100016                          BPL     2$              ;TYPE STACK?
2303  011604  016667  000006  167170          MOV     6(6),ANS1
2304  011612  016667  000010  167164          MOV     10(6),ANS2
2305  011620  016667  000012  167160          MOV     12(6),ANS3
2306  011626  016667  000014  167154          MOV     14(6),ANS4
2307  011634  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
2308  011640  000004  012465          2$:     TYPE    ,SPACE+4
2309  011644  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
2310  011646  004767  000114                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
2311  011652  005304                          DEC     R4
2312  011654  001371                          BNE     2$
2313  011656  005700                  3$:     TST     FPS
2314  011660  100016                          BPL     4$
2315  011662  000004  012461                  TYPE    ,SPACE
2316  011666  170367  167130                  STST    FEC
2317  011672  016705  167124                  MOV     FEC,TTY         ;TYPE FEC IN OCTAL
2318  011676  004767  000064                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
2319  011702  000004  012464                  TYPE    ,SPACE+3
2320  011706  016705  167112                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
2321  011712  004767  000050                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
2322  011716  012604                  4$:     MOV     (6)+,R4
2323  011720  005737  177570                  TST     @#SWR           ;HALT ON ERROR
2324  011724  100001                          BPL     .+4             ;SKIP IF CONTINUE
2325  011726  000000                          HALT                    ;HALT ON ERROR!
2326  011730  032737  001000  177570          BIT     #SW09,@#SWR     ;CHECK FOR INHIBIT LOOP ON ERROR
2327  011736  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
2328  011740  000002                          RTI
2329  011742  105067  167033                  CLRB    ICNT+1
2330  011746  032737  000400  177570          BIT     #SW08,@#SWR     ;CHECK FOR LOAD MICROBREAK
2331  011754  001233                          BNE     KIT             ;BRANCH IF NOT
2332  011756  113703  177570                  MOVB    @#SWR,R3        ;PUT MICROBREAK ADDRESS IN R3
2333  011762  170003                          LDUB                    ;LOAD MICROBREAK
2334  011764  000627                          BR      KIT             ;LOOP ON TEST UNTIL NO ERRORS
```

```
                     000001  000100  PRINTR: MOVB    #1,R4           ;SET ZERO FILL SWITCH
                     000000          PRINTS: CLR     R5              ;SUPRESS LEADING ZERO'S
                     000115                  MOV     @#176,R4+1      ;GET COUNT
                                             MOV     R4,-(SP)        ;SAVE R4
                     000241                  BIC     #241,PS         ;CLEAR CARRY TO AVOID ROTATE TROUBLE
                                             CLR     -(SP)           ;CLEAR FIRST BYTE
                                             ROLB    (SP)            ;ROTATE FIRST BIT
                                     10:     ROLB    (R1)+           ;CLEAR BYTE OF CHARACTER
                                             ROL     (SP)            ;ROTATE BIT INTO C
                                             ROLB    (R1)            ;PACK IT
                                             ROL     (SP)            ;ROTATE BIT INTO C
                                             ROLB    1(R1)           ;PACK IT
                                     20:     ROL     (SP)            ;ROTATE BIT INTO C
                                             ROLB    2(R1)           ;PACK IT
                     000004                  BIT     #4,@#176        ;CHECK FILL SWITCH
                     000060                  BEQ     30$             ;
                     000037                  ADD     #37-40,(R1)+    ;MAKE INTO ASCII CHAR
                     012114                  MOV     (R1),R4         ;REPEAT
                     000060                  ADD     #'0,R4          ;
                     012114                  TSTB    (R4)            ;TYPE IT
                     000207                  MOV     (SP)+,R4        ;RESTORE R4
                                             RTS     PC              ;
                     012114  000004  30:     .BLKW   4
                     012124  000000  R4$:    0
                     012126  005267  000364 ERROR:  INC     ERRORS          ;COUNT ERRORS
                     012130  132737  000001  000041        BITB    #1,@#41         ;AUTO MODE?
                     012140  001412                        BEQ     1$              ;NO!
                     012146  022767  000010  000346        CMP     #10.,ERRORS     ;TOO MANY?
                     012156  001006                        BNE     1$              ;NOT YET
                     012160  013700  000042                MOV     @#42,R0         ;GET ADDRESS
                     012164  001403                        BEQ     1$              ;FORGET IT IF ZERO
                     012166  005037  000042                CLR     @#42            ;ZAP 42
                     012172  004710                        JSR     PC,(0)          ;CALL THE MONITOR
                     012176  000207          1$:     RTS     PC              ;RETURN
```

```
 2379  012170  012777  012364  000306  POWDWN: MOV     #ILLUP,@UPVEC   ;SET FOR FAST UP
 2380  012176  012777  000340  000302          MOV     #340,@UPVEC+2   ;PRIO:7
 2381  012204  170246                          STFPS   -(6)            ;GET THE FPS
 2382  012206  170011                          SETD
 2383  012210  174046                          STD     AC0,-(6)        ;SAVE AC'S
 2384  012212  174146                          STD     AC1,-(6)
 2385  012214  174246                          STD     AC2,-(6)
 2386  012216  174346                          STD     AC3,-(6)
 2387  012220  172404                          LDD     AC4,AC0
 2388  012222  174046                          STD     AC0,-(6)
 2389  012224  172405                          LDD     AC5,AC0
 2390  012226  174046                          STD     AC0,-(6)
 2391  012230  010046                          MOV     R0,-(6)         ;SAVE REGISTERS
 2392  012232  010146                          MOV     R1,-(6)
 2393  012234  010246                          MOV     R2,-(6)
 2394  012236  010346                          MOV     R3,-(6)
 2395  012240  010446                          MOV     R4,-(6)
 2396  012242  010546                          MOV     R5,-(6)
 2397  012244  010667  000220                  MOV     SP,SAVE6        ;SAVE SP
 2398  012250  012777  012260  000236          MOV     #POWUP,@UPVEC   ;SET UP VECTOR
 2399  012256  000000                          HALT
 2400
 2401  012260  016706  000204          POWUP:  MOV     SAVE6,SP        ;GET SP
 2402  012264  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
 2403  012266  005201                  1$:     INC     R1
 2404  012270  001376                          BNE     1$
 2405  012272  012605                          MOV     (6)+,R5         ;GET THE REGISTERS
 2406  012274  012604                          MOV     (6)+,R4
 2407  012276  012603                          MOV     (6)+,R3
 2408  012300  012602                          MOV     (6)+,R2
 2409  012302  012601                          MOV     (6)+,R1
 2410  012304  012600                          MOV     (6)+,R0
 2411  012306  170011                          SETD
 2412  012310  172426                          LDD     (6)+,AC0        ;RESTORE THE AC'S
 2413  012312  174005                          STD     AC0,AC5
 2414  012314  172426                          LDD     (6)+,AC0
 2415  012316  174004                          STD     AC0,AC4
 2416  012320  172726                          LDD     (6)+,AC3
 2417  012322  172626                          LDD     (6)+,AC2
 2418  012324  172526                          LDD     (6)+,AC1
 2419  012326  172426                          LDD     (6)+,AC0
 2420  012330  170126                          LDFPS   (6)+            ;RESTORE FPS
 2421  012332  012777  012170  000140          MOV     #POWDWN,@DWNVEC ;SET UP THE POWER DOWN VECTOR
 2422  012340  012777  000340  000134          MOV     #340,@DWNVEC+2
 2423  012346  000004  012352                  TYPE    ..+2            ;.ASCIZ <15><12>"POWER"
 2424  012362  000002                          RTI
 2425
 2426  012364  000000                  ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
 2427  012366  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
```

```
2428  012370  010546                .IOT:   MOV     TTY,-(6)        ;SAVE TTY
2429  012372  017605  000002                MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
2430  012376  105715                1$:     TSTB    (TTY)           ;TERMINATOR?
2431  012400  001406                        BEQ     2$
2432  012402  112537  177566                MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
2433  012406  105737  177564                TSTB    @#177564        ;IS THE PRINTER READY
2434  012412  100375                        BPL     .-4
2435  012414  000770                        BR      1$              ;GET THE NEXT CHARACTER
2436  012416  017646  000002          2$:   MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
2437  012422  062766  000002  000004        ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
2438  012430  022666  000002                CMP     (6)+,2(6)       ;IS IT .+2?
2439  012434  301006                        BNE     3$              ;NO
2440  012436  062705  000002                ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
2441  012442  042705  000001                BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
2442  012446  010566  000002                MOV     TTY,2(6)        ;RESTORE ADDRESS
2443  012452  012605                3$:     MOV     (6)+,TTY        ;RESTORE TTY
2444  012454  000002                        RTI                     ;RETURN
2445
2446  012456  005015      000       RETURN: .ASCIZ  <15><12>        ;RETURN AND LINEFEED
2447  012461      015  020012  020040  SPACE:  .ASCIZ  <15><12>"   "   ;RETURN AND 3 SPACES
2448  012466      000
2449          012470                .EVEN
2450  012470  000000                SAVE6:  0
2451  012472  172160                FPTADR: 172160                  ;FLOATING POINT ADDRESS ON THE 11/20
2452  012474  000244  000246        FPVECT: 244,246                 ;FLOATING POINT VECTOR ADDRESS
2453  012500  000024  000026        DWNVEC: 24,26                   ;POWER DOWN VECTOR ADDRESS
2454  012504  000024  000026        UPVEC:  24,26                   ;POWER UP VECTOR ADDRESS
2455  012510  000000                .TYPE:  0
2456  012512  000000                TRPB:   0
2457  012514  000000                LAD:    0                       ;LOOP ADDRESS
2458  012516  000000                ERRORS: 0                       ;ERROR COUNT
2459  012520  000377                TIMES:  377                     ;ITERATION COUNT
2460  000001                        .END
```

DCFPG.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
AC0     =%000000          390#  2383   2387*  2388   2389*  2390   2412*  2413   2414*  2415   2419*
AC1     =%000001          391#  2384   2418*
AC2     =%000002          392#  2385   2417*
AC3     =%000003          393#  2386   2416*
AC4     =%000004          394#  2387   2415*
AC5     =%000005          395#  2388   2413*
ANS1    001002            424#   473*   478    499*   504    525*   530    551*   556    577*   582    603*   608
                          649*   654    655*   660    681*   686    707*   712    733*   738    759*   764    785*
                          806*   811*   816    837*   842    863*   868    889*   894    915*   920    941*   946
                          967*   972*   973*   998*  1019*  1024   1047*  1053   1070*  1098   1129*  1139   1170*
                         1180   1211*  1221   1252*  1262   1293*  1303   1334*  1344   1375*  1385   1416*  1426
                         1457*  1467   1498*  1508   1539*  1549   1580*  1590   1621*  1631   1662*  1672   1703*
                         1713   1744*  1754   1785*  1795   1826*  1836   1867*  1877   1908*  1918   1949*  1959
                         1990*  2000   2031*  2041   2072*  2082   2115*  2125   2161*  2171   2207*  2217   2299
                         2303*
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   1061   1102   1143
                         1184   1225   1266   1307   1348   1389   1430   1471   1512   1553   1594   1635   1676
                         1717   1758   1799   1840   1881   1922   1963   2004   2045   2086   2129   2175   2221
                         2304*
ANS3    001006            426#  1065   1106   1147   1188   1229   1270   1311   1352   1393   1434   1475   1516
                         1557   1598   1639   1680   1721   1762   1803   1844   1885   1926   1967   2008   2049
                         2090   2133   2179   2225   2305*
ANS4    001010            427#  1069   1110   1151   1192   1233   1274   1315   1356   1397   1438   1479   1520
                         1561   1602   1643   1684   1725   1766   1807   1848   1889   1930   1971   2012   2053
                         2094   2137   2183   2229   2306*
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    003776           1054#  1057   1061   1065   1069
AN3     001436            529#   530    533
AN30    004134           1095#  1098   1102   1106   1110
AN31    004272           1136#  1139   1143   1147   1151
AN32    004430           1177#  1180   1184   1188   1192
AN33    004566           1218#  1221   1225   1229   1233
AN34    004724           1259#  1262   1266   1270   1274
AN35    005062           1300#  1303   1307   1311   1315
AN36    005220           1341#  1344   1348   1352   1356
```

```
AN37    005356          1382#   1385    1389    1393    1397
AN4     001534           555#    556     559
AN40    005514          1423#   1426    1430    1434    1438
AN41    005652          1464#   1467    1471    1475    1479
AN42    006010          1505#   1508    1512    1516    1520
AN43    006146          1546#   1549    1553    1557    1561
AN44    006304          1587#   1590    1594    1598    1602
AN45    006442          1628#   1631    1635    1639    1643
AN46    006600          1669#   1672    1676    1680    1684
AN47    006736          1710#   1713    1717    1721    1725
AN5     001632           581#    582     585
AN50    007074          1751#   1754    1758    1762    1766
AN51    007232          1792#   1795    1799    1803    1807
AN52    007370          1833#   1836    1840    1844    1848
AN53    007526          1874#   1877    1881    1885    1889
AN54    007664          1915#   1919    1922    1926    1930
AN55    010022          1956#   1959    1963    1967    1971
AN56    010160          1997#   2000    2004    2008    2012
AN57    010316          2038#   2041    2045    2049    2053
AN6     001730           607#    609     611
AN60    010454          2079#   2082    2086    2090    2094
AN61    010616          2122#   2125    2129    2133    2137
AN62    010772          2168#   2171    2175    2179    2183
AN63    011146          2214#   2217    2221    2225    2229
AN7     002026           633#    634     637
A##     012124          2335*   2337*   2338*   2352*   2353    2356*   2367#
BEG     001026           414     435#
BEGIN   001062           439     444#   2244    2246
BELL  = 000007           379#   2236    2284
DISPLA= 177570           375#   2273*   2277*
DONE    011226          2233#
DWNVEC  012500           447*    448*   2421*   2422*   2453#
ERROR   012126          2286    2369#
ERRORS  012516          2369*   2372    2458#
FEA     001024           433#   2320
FEC     001022           418*    432#   2152*   2157    2198*   2203    2316*   2317
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*
                         780*    781     806*    807     832*    833     858*    859     884*    885     910*    911*
                         937     962*    963     988*    989    1014*   1015    1042*   1043    1068*   1069*   1094*
                        1165*   1166    1206*   1207    1247*   1248    1288*   1289    1329*   1330    1370*   1371*
                        1412    1452*   1453    1493*   1494    1534*   1535    1575*   1576    1616*   1657*   1698*
                        1698*   1699    1739*   1740    1780*   1781    1821*   1822    1862*   1863    1903*   1904*
                        1945    1985*   1986    2026*   2027    2067*   2068    2110*   2111    2151*   2153    2197*
                        2313
FPTADR  012472           441    2451#
FPVECT  012474           455*    456*   2452#
HLT   = 104000           377#    471     480     483     497     506     509     523     532     535     549     555
                         575     584     587     601     610     613     627     636     639     653     668     683
                         688     691     705     714     717     731     740     743     757     766     781     796
                         795     809     818     821     835     844     847     861     870     873     891     900
                         913     922     925     939     948     951     965     974     977     991    1000    1004
                        1026    1029    1045    1059    1063    1067    1071    1086    1100    1104    1108    1112
                        1141    1145    1149    1153    1168    1182    1186    1190    1194    1209    1213    1217    1232
                        1235    1250    1264    1268    1272    1276    1291    1305    1309    1313    1317    1332    1336
```

```
                                1350    1354    1358    1373    1387    1391    1395    1399    1414    1428    1432    1436    1440
                                1455    1469    1473    1477    1481    1496    1510    1514    1518    1522    1537    1551    1555
                                1559    1563    1578    1592    1596    1600    1604    1619    1633    1637    1641    1645    1660
                                1674    1678    1682    1686    1701    1715    1719    1723    1727    1742    1756    1760    1764
                                1768    1783    1797    1801    1805    1809    1824    1838    1842    1846    1850    1865    1873
                                1883    1887    1891    1906    1920    1924    1928    1932    1947    1951    1965    1969    1973
                                1988    2002    2006    2010    2014    2029    2043    2047    2051    2055    2070    2094    2098
                                2092    2096    2113    2127    2131    2135    2139    2155    2159    2173    2177    2181    2185
                                2201    2205    2219    2223    2227    2231
ICNT    001000                  452#     457*   2258    2266    2269    2270*   2271*   2273    2276*   2277    2329*
ILLUP   012364                  2379    2426#
KII     011444                  2263    2269    2276#   2331    2334
LAD     012614                  4559*   2272*   2278    2280    2457#
LDSG  = 170004                  408#
LDUB  = 170000                  404#
MRS   = 170006                  407#
M1      001236                  467     476#
M10     002120                  6595    6588#
M11     002216                  6775    6884#
M11B    001054                  4436    4414#
M12     002314                  7014    7108#
M12B    002412                  7277    7366#
M13     002510                  7533    7622#
M13B    003606                  7789    7888#
M14     002704                  8045    8140#
M14B    003002                  8301    8402#
M15     001334                  4893    5022#
M15B    003100                  8557    8662#
M16     003176                  8813    8922#
M16B    003274                  9069    9182#
M17     003372                  9325    9442#
M17B    003470                  9581    9702#
M2      003566                  9837    9962#
M20     003664                 10013   10122#
M21     003762                 10241   10362#
M21B    001432                  5139    5208#
M22     004124                 10805   10932#
M22B    004222                 11063   11134#
M23     004320                 11165   11758#
M23B    004456                 12005   12162#
M24     004554                 12045   12572#
M24B    004712                 12301   12982#
M25     005060                 12557   12982#
M25B    005202                 13021   13392#
M26     005340                 13565   13802#
M26B    001530                  5455    5544#
M27     005504                 14101   14212#
M27B    005642                 14451   14622#
M28     006000                 14805   15032#
M28B    006136                 15333   15442#
M29     006274                 15577   15852#
M29B    006432                 16115   16262#
M3      006570                 16569   16672#
M30     006726                 16937   17082#
M30B    001626                  5711    5802#
M31     007066                 17221   17492#
M32     007222                 17779   17902#
```

```
M5       007360          1820    1831#
M50      007516          1861    1872#
M500     007654          1902    1913#
M5000    010012          1943    1954#
M50000   010160          1984    1995#
M500000  010306          2025    2036#
M6       001724           587     606#
M60      010444          2066    2077#
M600     010606          2106    2120#
M6000    010764          2150    2166#
M60000   011136          2196    2212#
N        002032           623     632#
         000064 =        372#    459     485#    511#    537#    563#    589#    615#    641#    667#    693#    719#    745#
                         771#    797#    823#    849#    875#    901#    927#    953#    979#   1005#   1031#   1058    1073#
                        1114#   1155#   1196#   1237#   1278#   1319#   1360#   1401#   1442#   1483#   1524#   1565#   1606#
                        1647#   1688#   1729#   1770#   1811#   1852#   1893#   1934#   1975#   2016#   2057#   2098#   2141#
                        2187#   2233#
N1       001232           466     475#
N10      002114           648     657#
N100     002212           674     683#
N1000    002310           700     709#
N10000   002406           726     735#
N100000  002504           752     761#
N11      002602           778     787#
N110     002700           804     813#
N1100    002776           830     839#
N11000   001330           490     501#
N1100000 003074           856     865#
N1101    003172           882     891#
N1102    003270           908     917#
N1103    003366           934     943#
N1104    003464           960     969#
N1105    003562           986     995#
N1106    003660          1012    1021#
N1107    003756          1040    1050#
N111     001426           518     527#
N1110    004114          1081    1091#
N1111    004252          1122    1132#
N1112    004410          1163    1173#
N1113    004546          1204    1214#
N1114    004704          1245    1255#
N1115    005042          1286    1296#
N1116    005200          1327    1337#
N1117    005336          1368    1378#
N112     001524           544     553#
N1120    005474          1409    1419#
N1121    005632          1450    1460#
N1122    005770          1491    1501#
N1123    006126          1532    1542#
N1124    006264          1573    1583#
N1125    006422          1614    1624#
N1126    006560          1655    1665#
N1127    006716          1696    1706#
N113     001622           570     579#
N1130    007054          1737    1747#
N1131    007212          1778    1788#
```

DCFPG.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
056     010170          1991    2000#
057     010326          2032    2041#
06      001734           604     608#
060     010464          2073    2082#
061     010626          2115    2125#
062     011002          2162    2171#
063     011156          2208    2217#
07      002032           630     634#
PC    =%000007           389#    2286*   2364*   2377*   2378*
POWDWN  012170           447#    2379#   2421
POWUP   012260          2398    2401#
PRINTR  011766          2294    2297    2310    2318    2321    2335#
PRINTS  011776          2337#
PS    = 177776           373#
RETURN  012456          2290    2446#
R0    =%000000           381#    450*    2247*   2296    2374*   2391    2410*
R1    =%000001           382#    2392    2402*   2403*   2409*
R2    =%000002           383#    2393    2408*
R3    =%000003           384#    2260*   2299*   2332*   2394    2407*
R4    =%000004           385#    2287    2300*   2307*   2311*   2322*   2339    2340*   2358    2363*   2395    2406*
R5    =%000005           386#    2396    2405*
SAVE6   012470          2397*   2401    2450#
SAVLAD  011424          2265    2271#   2279
SCOPE = 104400           376#    464     490     516     542     568     594     620     646     672     698     724     750
                         776     802     828     854     880     906     932     958     984     1010    1038    1079    1120
                        1161    1202    1243    1284    1325    1366    1407    1448    1489    1530    1571    1612    1653
                        1694    1735    1776    1817    1858    1899    1940    1981    2022    2063    2104    2147    2193
                        2233
SP    =%000006           388#    435*    445*    2397    2401*
SPACE   012461          2295    2298    2308    2315    2319    2447#
STA0  = 170005           405#
STQ0  = 170007           406#
SWR   = 177570           374#    375     2234    2239    2256    2258    2260    2262    2264    2282    2288    2323    2326
                        2330    2332
SW08  = 000400           403#    2256    2330
SW09  = 001000           402#    2326
SW10  = 002000           401#    2234    2282
SW11  = 004000           400#    2264
SW12  = 010000           399#    2239
SW13  = 020000           398#    2288
SW14  = 040000           397#    2262
SW15  = 100000           396#
TIMES   012520          2268    2459#
TRPB    012512          2241*   2456#
TTY   =%000005           387#    2293*   2296*   2309*   2317*   2320*   2344*   2346*   2348*   2428    2429*   2430    2432
                        2440*   2441*   2442    2443*
TYPE  = 000004           378#    2237    2285    2290    2295    2298    2308    2315    2319    2362    2423
UPVEC   012504          2379*   2380*   2398*   2454#
YESRT.  011330           438*    446     2245    2254#
.     = 012522           410#    411     412#    416#    421#    470     479     482     496     505     508     522     531
                         534     548     557     560     574     583     586     600     609     612     626     635     638
                         652     661     664     678     687     690     704     713     716     730     739     742     756
                         765     768     782     791     794     808     817     820     834     843     846     860     865
                         872     886     895     898     912     921     924     938     947     950     964     973     976
                         990     999    1002    1016    1025    1028    1044    1059    1062    1066    1070    1085    1089
                        1103    1107    1111    1126    1140    1144    1148    1152    1167    1181    1185    1189    1193
```

```
                    1208    1222    1226    1230    1234    1249    1263    1267    1271    1275    1290    1304    1308
                    1312    1316    1331    1345    1349    1353    1357    1372    1386    1390    1394    1398    1413
                    1427    1431    1435    1439    1454    1468    1472    1476    1480    1495    1509    1513    1517
                    1521    1536    1550    1554    1558    1562    1577    1591    1595    1599    1603    1618    1632
                    1636    1640    1644    1659    1673    1677    1681    1685    1700    1714    1718    1726    1726
                    1741    1755    1759    1763    1767    1782    1796    1800    1804    1808    1823    1837    1841
                    1845    1849    1864    1878    1882    1886    1890    1905    1919    1923    1927    1931    1946
                    1960    1964    1968    1972    1987    2001    2005    2009    2013    2028    2042    2046    2050
                    2054    2069    2083    2087    2091    2095    2112    2126    2130    2134    2138    2154    2158
                    2172    2176    2180    2184    2200    2204    2218    2222    2226    2230    2324    2327    2336
                    2351    2354    2359    2366#   2423    2427    2434    2449#
.EMT    011332       453    2256#
.IOT    012370       449    2428#
.TRP    011472       451    2292#
.TYPE   012510      2236*   2237    2284*   2285    2455#
```

```
DUMP      371#   2293   2296   2309   2317   2320
PRINT     371#   2423
SDUMP     371#
STATUS    371#    468    494    520    546    572    598    624    650    676    702    728    754    780    806
           832    858    884    910    936    962    988   1014   1042   1083   1124   1165   1206   1247   1288
          1329   1370   1411   1453   1493   1534   1575   1616   1657   1698   1739   1780   1821   1862   1903
          1944   1985   2026   2067   2110   2151   2197
TYPEM     371#   2236   2284
$DIVD     459#   1032   1073   1114   1155   1196   1237   1278   1319   1360   1401   1442   1483   1524   1565
          1606   1647   1688   1729   1770   1811   1852   1893   1934   1975   2016   2057   2141   2187
$DIVDR    459#   2098
$DIVF     459#    485    511    537    563    589    615    641    667    693    719    745    771    797    823
           849    875    901    927    953    979   1005
```

DCFPG.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD     2437  2440
BEQ      470   479   482   496   505   508   522   531   534   548   557   560   574   583   586
         600   609   612   626   635   638   652   661   664   678   687   690   704   713   716
         730   739   742   756   765   768   782   791   794   808   817   820   834   843   846
         860   869   872   886   895   898   912   921   924   938   947   950   964   973   976
         990   999  1002  1016  1025  1028  1044  1058  1062  1066  1070  1085  1099  1103  1107
        1111  1126  1140  1144  1148  1152  1167  1181  1185  1189  1193  1208  1222  1226  1230
        1234  1249  1263  1267  1271  1275  1290  1304  1308  1312  1316  1331  1345  1349  1353
        1357  1372  1386  1390  1394  1398  1413  1427  1431  1435  1439  1454  1468  1472  1476
        1480  1495  1509  1513  1517  1521  1536  1550  1554  1558  1562  1577  1591  1595  1599
        1603  1618  1632  1636  1640  1644  1659  1673  1677  1681  1685  1700  1714  1718  1722
        1726  1741  1755  1759  1763  1767  1782  1796  1800  1804  1808  1823  1837  1841  1845
        1849  1864  1878  1882  1886  1890  1905  1919  1923  1927  1931  1946  1960  1964  1968
        1972  1987  2001  2005  2009  2013  2028  2042  2046  2050  2054  2069  2083  2087  2091
        2095  2112  2126  2130  2134  2138  2154  2158  2172  2176  2180  2184  2200  2204  2218
        2222  2226  2230  2248  2257  2259  2267  2279  2283  2301  2351  2354  2371  2375  2431
BIC     2307  2441
BIS     2243
BISB    2355
BIT     2234  2239  2256  2262  2264  2282  2286  2326  2330
BITB    2370
BNE     2235  2240  2263  2265  2269  2289  2312  2327  2331  2357  2359  2373  2404  2439
BPL     2242  2302  2314  2324  2434
BR       439   474   500   526   552   578   604   630   656   682   708   734   760   786   812
         938   864   890   916   942   968   994  1020  1048  1089  1130  1171  1212  1253  1294
        1335  1376  1417  1458  1499  1540  1581  1622  1663  1704  1745  1786  1827  1868  1909
        1950  1991  2032  2073  2116  2162  2208  2245  2334  2336  2342  2427  2435
CLR      457   458  2238  2337  2376  2402
CLRB    2329  2341  2343  2361
CMP      469   478   481   495   504   507   521   530   533   547   556   559   573   582   585
         599   608   611   625   634   637   651   660   663   677   686   689   703   712   715
         729   738   741   755   764   767   781   790   793   807   816   819   833   842   845
         859   868   871   885   894   897   911   920   923   937   946   949   963   972   975
         989   998  1001  1015  1024  1027  1043  1057  1061  1065  1069  1084  1098  1102  1106
        1110  1125  1139  1143  1147  1151  1166  1180  1184  1188  1192  1207  1221  1225  1229
        1233  1248  1262  1266  1270  1274  1289  1303  1307  1311  1315  1330  1344  1348  1352
        1356  1371  1385  1389  1393  1397  1412  1426  1430  1434  1438  1453  1467  1471  1475
        1479  1494  1508  1512  1516  1520  1535  1549  1553  1557  1561  1576  1590  1594  1598
        1602  1617  1631  1635  1639  1643  1658  1672  1676  1680  1684  1699  1713  1717  1721
        1725  1740  1754  1758  1762  1766  1781  1795  1799  1803  1807  1822  1836  1840  1944
        1848  1863  1877  1881  1885  1889  1904  1918  1922  1926  1930  1945  1959  1963  1967
        1971  1986  2000  2004  2008  2012  2027  2041  2045  2049  2053  2068  2082  2086  2090
        2094  2111  2125  2129  2133  2137  2153  2157  2171  2175  2179  2183  2199  2203  2217
        2221  2225  2229  2358  2372  2438
CMPB    2258  2268
COM     2241
DEC     2311
DIVD    1041  1082  1123  1164  1205  1246  1287  1328  1369  1410  1451  1492  1533  1574  1615
        1656  1697  1738  1779  1820  1861  1902  1943  1984  2025  2066  2109  2150  2196
DIVF     467   493   519   545   571   597   623   649   675   701   727   753   779   805   831
         857   883   909   935   961   987  1013
EMT      377
HALT     411   419  2325  2399  2426
INC     2369  2403
INCB    2271  2276  2352  2356
IOT      378
```

DCFPG.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
JMP      414
JSR     2249  2286  2294  2297  2310  2318  2321  2377
LDD     1040  1081  1122  1163  1204  1245  1286  1327  1368  1409  1450  1491  1532  1573  1614
        1655  1696  1737  1778  1819  1860  1901  1942  1983  2024  2065  2106  2108  2149  2195
        2387  2389  2412  2414  2416  2417  2418  2419
LDF      466   492   518   544   570   596   622   648   674   700   726   752   778   804   830
         856   882   908   934   960   986  1012
LDFPS    465   491   517   543   569   595   621   647   673   699   725   751   777   803   829
         855   881   907   933   959   985  1011  1039  1080  1121  1162  1203  1244  1285  1326
        1367  1408  1449  1490  1531  1572  1613  1654  1695  1736  1777  1818  1859  1900  1941
        1982  2023  2064  2105  2148  2194  2420
LDUB    2261  2333
MOV      435   436   438   441   444   445   446   447   448   449   450   451   452   453   454
         455   456  2236  2244  2246  2247  2272  2273  2277  2280  2284  2287  2291  2293  2296
        2299  2303  2304  2305  2306  2309  2317  2320  2322  2339  2340  2363  2374  2379  2380
        2391  2392  2393  2394  2395  2396  2397  2398  2401  2405  2406  2407  2408  2409  2410
        2421  2422  2428  2429  2436  2442  2443
MOVB    2260  2270  2300  2332  2335  2338  2360  2432
NOP     2250  2251  2252
ROL     2344  2346  2348
ROLB    2345  2347  2349
RTI      420  2253  2254  2274  2281  2328  2424  2444
RTS     2364  2378
SETD    2382  2411
STD     1047  1088  1129  1170  1211  1252  1293  1334  1375  1416  1457  1498  1539  1580  1621
        1662  1703  1744  1785  1826  1867  1908  1949  1990  2031  2072  2107  2115  2161  2207
        2383  2384  2385  2386  2388  2390  2413  2415
STF      473   499   525   551   577   603   629   655   681   707   733   759   785   811   837
         863   889   915   941   967   993  1019
STFPS    417   468   494   520   546   572   598   624   650   676   702   728   754   780   806
         832   858   884   910   936   962   988  1014  1042  1083  1124  1165  1206  1247  1288
        1329  1370  1411  1452  1493  1534  1575  1616  1657  1698  1739  1780  1821  1862  1903
        1944  1985  2026  2067  2110  2151  2197  2381
STST     418  2152  2198  2316
SUB     2292
TRAP     376
TST      437  2278  2313  2323
TSTB    2266  2350  2353  2430  2433
.ASCIZ  2424  2446  2447
.BLKW   2366
.ENABL   371
.END    2460
.ENDC    469   473   495   499   521   525   547   551   573   577   599   603   625   629   651
         655   677   681   703   707   729   733   755   759   781   785   807   811   833   837
         859   863   885   889   911   915   937   941   963   967   989   993  1015  1019  1043
        1047  1084  1088  1125  1129  1166  1170  1207  1211  1248  1252  1289  1293  1330  1334
        1371  1375  1412  1416  1453  1457  1494  1498  1535  1539  1576  1580  1617  1621  1658
        1662  1699  1703  1740  1744  1781  1785  1822  1826  1863  1867  1904  1908  1945  1949
        1986  1990  2027  2031  2068  2072  2111  2115  2153  2161  2199  2207
.EVEN   2424  2449
.FLT2    475   476   477   501   502   503   527   528   529   553   554   555   579   580   581
         605   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
.FLT4   1050  1052  1054  1091  1093  1095  1132  1134  1136  1173  1175  1177  1214  1216  1218
```

Spooler runtime 9 Seconds, 40 KCS, 230 disk reads, 3 disk writes, 66 pages