FP11

MAINTENANCE INSTRUCTION
MD-11-DCFPM-B
TESTS

EP-DCFPM-B-DL

FICHE 1 OF 1

AUG 1978
digital
MADE IN USA

```
                        .REPT   2
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCFPM-B-D

PRODUCT NAME: FP11 MAINTENANCE INSTRUCTION TESTS

DATE CREATED: NOVEMBER 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: BOB BRAIN

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

CONTENTS

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


                                1.          ABSTRACT

                                            THE PURPOSE OF THIS PROGRAM IS THREE FOLD.   FIRST  TO  TEST
                                            THE  MAINTENANCE  FEATURES, SECOND TO TRACE MULD THROUGH ITS
                                            STEPS, THIRD TO TRACE DIVD.


                                2.          REQUIREMENTS

                                2.1         EQUIPMENT

                                            PDP11/45 STANDARD COMPUTER WITH FP11 OPTION


                                2.2         STORAGE

                                            PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

                                2.3         PRELIMINARY PROGRAMS

                                            NONE


                                3.          LOADING PROCEDURE

                                            USE STANDARD PROCEDURE FOR ABS TAPES.


                                4.          STARTING PROCEDURE

                                4.1         CONTROL SWITCH SETTINGS

                                            SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)


                                4.2         STARTING ADDRESS

                                            THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.


                                4.3         PROGRAM AND/OR OPERATOR ACTION

                                4.3.1       BASIC MAINTENANCE TEST

                                            1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
```

```
                                        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 DATA DISPLAY SWITCH TO THE DISPLAY POSITION.

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

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

4.3.3 DIVIDE TRACE

SAME AS 4.3.2 EXCEPT START AT 210

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

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

5.1.1 SWITCH SETTINGS ARE:

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

5.2 SUBROUTINE ABSTRACTS

DESCRIPTION

5.2.1 SCOPE

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

5.2.2 HLT

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

5.2.3 TRTRAP

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

5.2.4 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR 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 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 FOR THE BASIC TEST IS AS FOLLOWS:

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

DESCRIPTION

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.

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

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

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

```
                                MAINDEC-11-DCFPM-B-D    FP11 MAINTENANCE TEST       PAGE   7
                                DESCRIPTION


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

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

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

                                SWITCH                  USE
                                ------          --------------------
                                  8             LOOP ON TEST IN SW<7:0>
                                  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 (-0)
                                  7     DOUBLE PRECISION MODE             16    UBREAK TRAP
                                  8     INTERUPT ON CONVERSION ERROR
                                  9     INTERUPT ON OVERFLOW
                                  10    INTERUPT ON UNDERFLOW
                                  11    INTERJPT 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      STAB=   170005
                    170007      STOB=   170007
                    170006      MRS=    170006
                    170004      LOSC=   170004

                    000000      .=      0                       ;TRAP CATCHER FROM 0 - 776
                    000200      .=      200
000200  000167  000622          JMP     BEG
000204  000167  003234          JMP     TSTMUL
000210  000167  003464          JMP     TSTDIV

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

```
                001000                  .=      1000

                000000          ICNT:   0                       ;ITERATION COUNT = LH   TEST NO. = RH
                000000          ANS1:   0                       ;FIRST ANSWER (SEE CODE)
                000000          ANS2:   0
                000000          ANS3:   0
                000000          ANS4:   0
                000000          ANS5:   0
                000000          ANS6:   0
                000000          ANS7:   0
                000000          ANS8:   0
                000000          FEC:    0                       ;FLOATING EXCEPTION CODES
                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  003476          MOV     #6,YESRT        ;FUDGE IN RTT IF 11/45
001052  000403                          BR      BEGIN

001054  016737  004612  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  001550  000014          MOV     #YESRT,@#14     ;SET TRACE TRAP VECTOR
001102  012777  005370  004570          MOV     #PDWDWN,@DWNVEC
001110  012777  000340  004564          MOV     #340,@DWNVEC+2
001116  012737  005570  000020          MOV     #.IOT,@#20      ;SET UP VECTOR 20
001124  012700  000030                  MOV     #30,R0          ;SET R0 TO VECTOR 30
001130  012720  004704                  MOV     #.TRP,(0)+      ;SET EMT VECTOR
001134  012720  000340                  MOV     #340,(0)+
001140  012720  004552                  MOV     #.EMT,(0)+      ;SET TRAP VECTOR
001144  012710  000340                  MOV     #340,(0)
001150  012777  000760  004516          MOV     #FLTERR,@FPVECT ;LOAD INTERUPT VECTOR
001156  012777  000340  004512          MOV     #340,@FPVECT+2  ;LOCK UP PROCESSOR
001164  005067  177610                  CLR     ICNT
001170  005067  004520                  CLR     LAD
001174  170127  040000                  LDFPS   #40000
001200  005003                          CLR     R3              ;CLEAR
001202  170003                          LDUB                    ;UBREAK REG
```

```
                                ;*****************************************************************
                                ;TEST 1          TRAP BEFORE LOAD [5]
                                ;*****************************************************************
001224  104420                  TST1:   SCOPE

001226  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
001232  172467  004066                  LDD     ALT1A,AC0       ;LOAD ALT 1'S
001236  012703  000005                  MOV     #5,R3           ;LOAD 5 INTO
001242  170003                          LDUB                    ;UBREAK REG.
001244  172467  004064                  LDD     ALT1B,AC0       ;LOAD OTHER 1'S
001250  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001252  170367  177564                  STST    FEC             ;STORE EXCEPTION CODES
001256  022700  140230                  CMP     #140230,FPS     ;CHECK FLOATING POINT STATUS
001262  001401                          BEQ     .+4             ;BRANCH IF OK
001264  104000                          HLT                     ;FPS NOT EQUAL TO 140230

001266  022767  000016  177546          CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
001274  001401                          BEQ     .+4             ;BRANCH IF OK
001276  104000                          HLT                     ;FEC NOT EQUAL TO 16

001300  174067  177516                  STD     AC0,ANS1        ;GET RESULT
001304  173467  004014                  CMPD    ALT1A,AC0       ;CHECK RESULT
001310  170000                          CFCC                    ;GET CC BITS
001312  001401                          BEQ     .+4             ;SKIP IF OK
001314  104004                          HLT+4                   ;SECOND LDD DIDN'T TRAP

001316  005003                          CLR     R3              ;CLEAR
001320  170003                          LDUB                    ;UBREAK REG

                                ;*****************************************************************
                                ;TEST 2          TRAP AFTER LOADING 32 BITS [21]
                                ;*****************************************************************
001322  104400                  TST2:   SCOPE

001324  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
001330  012703  000021                  MOV     #21,R3          ;LOAD 21 INTO
001334  170003                          LDUB                    ;UBREAK REG.
001336  172467  003772                  LDD     ALT1B,AC0       ;LOAD FIRST 32 BITS
001342  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
001344  170367  177472                  STST    FEC             ;STORE EXCEPTION CODES
001350  022700  140230                  CMP     #140230,FPS     ;CHECK FLOATING POINT STATUS
001354  001401                          BEQ     .+4             ;BRANCH IF OK
001356  104000                          HLT                     ;FPS NOT EQUAL TO 140230

001360  022767  000016  177454          CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
001366  001401                          BEQ     .+4             ;BRANCH IF OK
001370  104000                          HLT                     ;FEC NOT EQUAL TO 16

001372  174067  177424                  STD     AC0,ANS1        ;GET ANSWER
001376  173467  003736                  CMPD    ALT1C,AC0       ;RESULT RIGHT?
001402  170000                          CFCC                    ;GET CC
001404  001401                          BEQ     .+4             ;SKIP IF OK
001406  104004                          HLT+4                   ;WEIRD ANSWER
```

A2

```
001370  005003                  CLR     R3              ;CLEAR
001372  170003                  LDUB                    ;UBREAK REG

                        ;*****************************************************************
                        ;TEST 3         TEST OF STA0 AND STQ?  DOUBLE MODE
                        ;*****************************************************************
001374  104400          TST3:   SCOPE

001376  170127  040220          LDFPS   #40220          ;DOUBLE/MAINT BITS
001402  172467  003722          LDD     ALTA,AC0        ;LOAD ALT 1'S
001406  012703  000115          MOV     #115,R3         ;LOAD 115 INTO
001412  170003                  LDUB                    ;UBREAK REG.
001414  172067  003720          ADDD    ALTB,AC0        ;ADD ALT 1'S
001420  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001422  170367  177374          STST    FEC             ;STORE EXCEPTION CODES
001426  022700  140220          CMP     #140220,FPS     ;CHECK FLOATING POINT STATUS
001432  001401                  BEQ     .+4             ;BRANCH IF OK
001434  104000                  HLT                     ;FPS NOT EQUAL TO 140220

001436  022767  000016  177356  CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
001444  001401                  BEQ     .+4             ;BRANCH IF OK
001446  104000                  HLT                     ;FEC NOT EQUAL TO 16

001450  170005                  STA0                    ;GET AR INTO AC0
001452  174067  177324          STD     AC0,ANS1        ;GET FOR TYPING
001456  042767  177600  177316  BIC     #177600,ANS1    ;CLEAR JUNK
001464  022767  000125  177310  CMP     #125,ANS1       ;IS ANS1 = 125
001472  001401                  BEQ     .+4             ;SKIP IF SAME
001474  104004                  HLT+4                   ;ANS1 NOT = 125
001476  022767  052525  177300  CMP     #52525,ANS2     ;IS ANS2 = 52525
001504  001401                  BEQ     .+4             ;SKIP IF SAME
001506  104004                  HLT+4                   ;ANS2 NOT = 52525
001510  022767  052525  177270  CMP     #52525,ANS3     ;IS ANS3 = 52525
001516  001401                  BEQ     .+4             ;SKIP IF SAME
001520  104004                  HLT+4                   ;ANS3 NOT = 52525
001522  022767  052525  177260  CMP     #52525,ANS4     ;IS ANS4 = 52525
001530  001401                  BEQ     .+4             ;SKIP IF SAME
001532  104004                  HLT+4                   ;ANS4 NOT = 52525

001534  170007                  STQ0                    ;GET QR
001536  174067  177240          STD     AC0,ANS1        ;GET RESULT
001542  042767  177600  177232  BIC     #177600,ANS1    ;CLEAR JUNK
001550  022767  000052  177224  CMP     #52,ANS1        ;IS ANS1 = 52
001556  001401                  BEQ     .+4             ;SKIP IF SAME
001560  104004                  HLT+4                   ;ANS1 NOT = 52
001562  022767  125252  177214  CMP     #125252,ANS2    ;IS ANS2 = 125252
001570  001401                  BEQ     .+4             ;SKIP IF SAME
001572  104004                  HLT+4                   ;ANS2 NOT = 125252
001574  022767  125252  177204  CMP     #125252,ANS3    ;IS ANS3 = 125252
001602  001401                  BEQ     .+4             ;SKIP IF SAME
001604  104004                  HLT+4                   ;ANS3 NOT = 125252
001606  022767  125252  177174  CMP     #125252,ANS4    ;IS ANS4 = 125252
001614  001401                  BEQ     .+4             ;SKIP IF SAME
001616  104004                  HLT+4                   ;ANS4 NOT = 125252
```

```
                                ;*****************************************************************
                                ;TEST 4          TEST OF STA0 AND STQ0  FLOATING MODE
                                ;*****************************************************************
001624  104400          TST4:   SCOPE

001626  170127  040020          LDFPS   #40020          ;FLOATING/MAINT BITS
001632  172467  003502          LDF     ALTB,AC0        ;LOAD ALT 1'S
001636  012703  000115          MOV     #115,R3         ;LOAD 115 INTO
001642  170003                  LDUB                    ;UBREAK REG,
001644  172067  003460          ADDF    ALTA,AC0        ;ADD ALT 1'S
001650  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
001652  170367  177144          STST    FEC             ;STORE EXCEPTION CODES
001656  022700  140020          CMP     #140020,FPS     ;CHECK FLOATING POINT STATUS
001662  001401                  BEQ     .+4             ;BRANCH IF OK
001664  104000                  HLT                     ;FPS NOT EQUAL TO 140020

001666  022767  000016  177126  CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
001674  001401                  BEQ     .+4             ;BRANCH IF OK
001676  104000                  HLT                     ;FEC NOT EQUAL TO 16

001700  170007                  STQ0                    ;GET QR INTO AC0
001702  174067  177074          STF     AC0,ANS1        ;GET FOR TYPING
001706  042767  177600  177066  BIC     #177600,ANS1    ;CLEAR JUNK
001714  022767  000125  177060  CMP     #125,ANS1       ;IS ANS1 = 125
001722  001401                  BEQ     .+4             ;SKIP IF SAME
001724  104004                  HLT+4                   ;ANS1 NOT = 125
001726  022767  052525  177050  CMP     #52525,ANS2     ;IS ANS2 = 52525
001734  001401                  BEQ     .+4             ;SKIP IF SAME
001736  104004                  HLT+4                   ;ANS2 NOT = 52525
001740  022767  125252  177040  CMP     #125252,ANS3    ;IS ANS3 = 125252
001746  001401                  BEQ     .+4             ;SKIP IF SAME
001750  104004                  HLT+4                   ;ANS3 NOT = 125252
001752  022767  125252  177030  CMP     #125252,ANS4    ;IS ANS4 = 125252
001760  001401                  BEQ     .+4             ;SKIP IF SAME
001762  104004                  HLT+4                   ;ANS4 NOT = 125252

001764  170005                  STA0                    ;GET AR
001766  174067  177010          STF     AC0,ANS1        ;GET RESULT
001772  042767  177600  177002  BIC     #177600,ANS1    ;CLEAR JUNK
002000  022767  000052  176774  CMP     #52,ANS1        ;IS ANS1 = 52
002006  001401                  BEQ     .+4             ;SKIP IF SAME
002010  104002                  HLT+2                   ;ANS1 NOT = 52
002012  022767  125252  176764  CMP     #125252,ANS2    ;IS ANS2 = 125252
002020  001401                  BEQ     .+4             ;SKIP IF SAME
002022  104002                  HLT+2                   ;ANS2 NOT = 125252

002024  005003                  CLR     R3              ;CLEAR
002026  170003                  LDUB                    ;UBREAK REG,

                                ;*****************************************************************
```

C2

```
                                        ;TEST 5         TEST OF MRS
                                        ;*****************************************************************
002030  104400                  TST5:   SCOPE

002032  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
002036  012703  000115                  MOV     #115,R3         ;LOAD 115 INTO
002042  170003                          LDUB                    ;UBREAK REG.
002044  172467  003260                  LDD     ALTA,AC0        ;LOAD ALT 1'S
002050  172067  003264                  ADDD    ALTB,AC0        ;ADD ALT 1'S
002054  170006                          MRS
002056  170007                          STQ0                    ;GET QR INTO AC0
002060  174067  176716                  STD     AC0,ANS1        ;GET RESULT
002064  170200                          STFPS   FPS             ;GET FPS
002066  042767  177600  176706          BIC     #177600,ANS1    ;CLEAR JUNK
002074  022767  000125  176700          CMP     #125,ANS1       ;IS IT 125?
002102  001401                          BEQ     .+4             ;SKIP IF OK
002104  104004                          HLT+4                   ;FIRST WORD IS NOT 125

002106  170005                          STAB                    ;GET AR INTO AC0
002110  174067  176666                  STD     AC0,ANS1        ;GET RESULT
002114  042767  177600  176660          BIC     #177600,ANS1    ;CLEAR JUNK
002122  022767  000152  176652          CMP     #152,ANS1       ;CHECK RESULT
002130  001401                          BEQ     .+4             ;SKIP IF OK
002132  104004                          HLT+4                   ;AR NOT 152

002134  005003                          CLR     R3              ;CLEAR
002136  170003                          LDUB                    ;UBREAK REG

                                        ;*****************************************************************
                                        ;TEST 6         TEST OF LDSC FOR MULD OF 1
                                        ;*****************************************************************
002140  104400                  TST6:   SCOPE

002142  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
002146  012703  000230                  MOV     #230,R3         ;LOAD 230 INTO
002152  170003                          LDUB                    ;UBREAK REG.
002154  012704  000001                  MOV     #1,R4           ;LOAD 1 INTO
002160  170004                          LDSC                    ;SHIFT COUNT REG.
002162  172467  000206                  LDD     1$,AC0          ;LOAD INTO AC0 - 40000,0,0,0
002166  171067  000212                  MULD    2$,AC0          ;MULD BY 40252,125252,125252,125252
002172  170200                          STFPS   FPS             ;STORE FLOATING POINT STATUS
002174  170367  176622                  STST    FEC             ;STORE EXCEPTION CODES
002200  022700  140220                  CMP     #140220,FPS     ;CHECK FLOATING POINT STATUS
002204  001401                          BEQ     .+4             ;BRANCH IF OK
002206  104000                          HLT                     ;FPS NOT EQUAL TO 140220

002210  022767  000016  176604          CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
002216  001401                          BEQ     .+4             ;BRANCH IF OK
002220  104000                          HLT                     ;FEC NOT EQUAL TO 16

002222  170005                          STAB                    ;GET AR
002224  174067  176552                  STD     AC0,ANS1        ;GET FOR TYPING
002230  042767  177600  176544          BIC     #177600,ANS1    ;CLEAR JUNK
002236  170007                          STQ0                    ;GET QR
```

D2

```
   691  002240  174067  176546                STD     AC0,ANS5        ;GET FOR TYPING
   692  002244  042767  177000  176540       BIC     #177000,ANS5    ;CLEAR JUNK
   693  002252  022767  000000  176522       CMP     #0,ANS1         ;IS ANS1 = 0
   694  002260  001401                       BEQ     .+4             ;SKIP IF SAME
   695  002262  104010                       HLT+8.                  ;ANS1 NOT = 0
   696  002264  022767  000000  176512       CMP     #0,ANS2         ;IS ANS2 = 0
   697  002272  001401                       BEQ     .+4             ;SKIP IF SAME
   698  002274  104010                       HLT+8.                  ;ANS2 NOT = 0
   699  002276  022767  000000  176502       CMP     #0,ANS3         ;IS ANS3 = 0
   700  002304  001401                       BEQ     .+4             ;SKIP IF SAME
   701  002306  104010                       HLT+8.                  ;ANS3 NOT = 0
   702  002310  022767  000000  176472       CMP     #0,ANS4         ;IS ANS4 = 0
   703  002316  001401                       BEQ     .+4             ;SKIP IF SAME
   704  002320  104010                       HLT+8.                  ;ANS4 NOT = 0
   705  002322  022767  000125  176462       CMP     #125,ANS5       ;IS ANS5 = 125
   706  002330  001401                       BEQ     .+4             ;SKIP IF SAME
   707  002332  104010                       HLT+8.                  ;ANS5 NOT = 125
   708  002334  022767  052525  176452       CMP     #52525,ANS6     ;IS ANS6 = 52525
   709  002342  001401                       BEQ     .+4             ;SKIP IF SAME
   710  002344  104010                       HLT+8.                  ;ANS6 NOT = 52525
   711  002346  022767  052525  176442       CMP     #52525,ANS7     ;IS ANS7 = 52525
   712  002354  001401                       BEQ     .+4             ;SKIP IF SAME
   713  002356  104010                       HLT+8.                  ;ANS7 NOT = 52525
   714  002360  022767  052525  176432       CMP     #52525,ANS8     ;IS ANS8 = 52525
   715  002366  001401                       BEQ     .+4             ;SKIP IF SAME
   716  002370  104010                       HLT+8.                  ;ANS8 NOT = 52525
   717  002372  000410                       BR      5$              ;SKIP JUNK
   718
   719  002374  040000  000000  000000  1$:  40000,0,0,0
   720  002402  000000
   721  002404  040252  125252  125252  2$:  40252,125252,125252,125252
   722  002412  125252
   723
   724  002414  005003                  5$:  CLR     R3              ;CLEAR
   725  002416  170003                       LDUB                    ;UBREAK REG.
   726
   727
   728                                  ;*****************************************************************
   729                                  ;TEST 7         TEST OF LDSC FOR MULD OF 71
   730                                  ;*****************************************************************
   731  002420  104400                  TST7:SCOPE
   732
   733  002422  170127  040220               LDFPS   #40220          ;DOUBLE/MAINT BITS
   734  002426  012703  000230               MOV     #230,R3         ;LOAD 230 INTO
   735  002432  170003                       LDUB                    ;UBREAK REG.
   736  002434  012704  000071               MOV     #71,R4          ;LOAD 71 INTO
   737  002440  170004                       LDSC                    ;SHIFT COUNT REG.
   738  002442  172467  000206               LDD     1$,AC0          ;LOAD INTO AC0 - 40000,0,0,0
   739  002446  171067  000212               MULD    2$,AC0          ;MULD BY 40252,125252,125252,125252
   740  002452  170200                       STFPS   FPS             ;STORE FLOATING POINT STATUS
   741  002454  170367  176342               STST    FEC             ;STORE EXCEPTION CODES
   742  002460  022700  140220               CMP     #140220,FPS     ;CHECK FLOATING POINT STATUS
   743  002464  001401                       BEQ     .+4             ;BRANCH IF OK
   744  002466  104000                       HLT                     ;FPS NOT EQUAL TO 140220
```

```
                                                    CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
002470  022767  000016  176324                      BEQ     .+4             ;BRANCH IF OK
002476  001401                                      HLT                     ;FEC NOT EQUAL TO 16
002500  104000
                                                    STA0                    ;GET AR
002502  170005                                      STD     AC0,ANS1        ;GET FOR TYPING
002504  174067  176272                              BIC     #177600,ANS1    ;CLEAR JUNK
002510  042767  177600  176264                      STQ0                    ;GET QR
002516  170007                                      STD     AC0,ANS5        ;GET FOR TYPING
002520  174067  176266                              BIC     #177600,ANS5    ;CLEAR JUNK
002524  042767  177600  176260                      CMP     #125,ANS1       ;IS ANS1 = 125
002532  022767  000125  176242                      BEQ     .+4             ;SKIP IF SAME
002540  001401                                      HLT+8.                  ;ANS1 NOT = 125
002542  104010                                      CMP     #52525,ANS2     ;IS ANS2 = 52525
002544  022767  052525  176232                      BEQ     .+4             ;SKIP IF SAME
002552  001401                                      HLT+8.                  ;ANS2 NOT = 52525
002554  104010                                      CMP     #52525,ANS3     ;IS ANS3 = 52525
002556  022767  052525  176222                      BEQ     .+4             ;SKIP IF SAME
002564  001401                                      HLT+8.                  ;ANS3 NOT = 52525
002566  104010                                      CMP     #52525,ANS4     ;IS ANS4 = 52525
002570  022767  052525  176212                      BEQ     .+4             ;SKIP IF SAME
002576  001401                                      HLT+8.                  ;ANS4 NOT = 52525
002600  104010                                      CMP     #0,ANS5         ;IS ANS5 = 0
002602  022767  000000  176202                      BEQ     .+4             ;SKIP IF SAME
002610  001401                                      HLT+8.                  ;ANS5 NOT = 0
002612  104010                                      CMP     #0,ANS6         ;IS ANS6 = 0
002614  022767  000000  176172                      BEQ     .+4             ;SKIP IF SAME
002622  001401                                      HLT+8.                  ;ANS6 NOT = 0
002624  104010                                      CMP     #0,ANS7         ;IS ANS7 = 0
002626  022767  000000  176162                      BEQ     .+4             ;SKIP IF SAME
002634  001401                                      HLT+8.                  ;ANS7 NOT = 0
002636  104010                                      CMP     #0,ANS8         ;IS ANS8 = 0
002640  022767  000000  176152                      BEQ     .+4             ;SKIP IF SAME
002646  001401                                      HLT+8.                  ;ANS8 NOT = 0
002650  104010                                      BR      5$              ;SKIP JUNK
002652  000410

002654  040000  000000  000000  1$:     40000,0,0,0
002662  000000
002664  040252  125252  125252  2$:     40252,125252,125252,125252
002672  125252

002674  005003                  5$:     CLR     R3              ;CLEAR
002676  170003                          LDUB                    ;UBREAK REG.


                                ;*****************************************************************
                                ;TEST 10        TEST OF LDSC FOR DIVD OF 1
                                ;*****************************************************************
002700  104400                  TST10:  SCOPE

002702  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
002706  012703  000314                  MOV     #314,R3         ;LOAD 314 INTO
002712  170003                          LDUB                    ;UBREAK REG.
```

F2

```
   799  002714  012704  000001              MOV     #1,R4           ;LOAD 1 INTO
   800  002720  170004                      LDSC                    ;SHIFT COUNT REG.
   801  002722  172467  000206              LDD     1$,AC0          ;LOAD INTO AC0 = 40000,0,0,0
   802  002726  174467  000212              DIVD    2$,AC0          ;DIVD BY 40252,125252,125252,125252
   803  002732  170200                      STFPS   FPS             ;STORE FLOATING POINT STATUS
   804  002734  170367  176062              STST    FEC             ;STORE EXCEPTION CODES
   805  002740  022700  140220              CMP     #140220,FPS     ;CHECK FLOATING POINT STATUS
   806  002744  001401                      BEQ     .+4             ;BRANCH IF OK
   807  002746  104000                      HLT                     ;FPS NOT EQUAL TO 140220
   808
   809  002750  022767  000016  176044      CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
   810  002756  001401                      BEQ     .+4             ;BRANCH IF OK
   811  002760  104000                      HLT                     ;FEC NOT EQUAL TO 16
   812
   813  002762  170005                      STA0                    ;GET AR
   814  002764  174067  176012              STD     AC0,ANS1        ;GET FOR TYPING
   815  002770  042767  177600  176004      BIC     #177600,ANS1    ;CLEAR JUNK
   816  002776  170007                      STQ0                    ;GET QR
   817  003000  174067  176006              STD     AC0,ANS5        ;GET FOR TYPING
   818  003004  042767  177600  176000      BIC     #177600,ANS5    ;CLEAR JUNK
   819  003012  022767  000052  175762      CMP     #52,ANS1        ;IS ANS1 = 52
   820  003020  001401                      BEQ     .+4             ;SKIP IF SAME
   821  003022  104013                      HLT+8.                  ;ANS1 NOT = 52
   822  003024  022767  125252  175752      CMP     #125252,ANS2    ;IS ANS2 = 125252
   823  003032  001401                      BEQ     .+4             ;SKIP IF SAME
   824  003034  104010                      HLT+8.                  ;ANS2 NOT = 125252
   825  003036  022767  125252  175742      CMP     #125252,ANS3    ;IS ANS3 = 125252
   826  003044  001401                      BEQ     .+4             ;SKIP IF SAME
   827  003046  104010                      HLT+8.                  ;ANS3 NOT = 125252
   828  003050  022767  125254  175732      CMP     #125254,ANS4    ;IS ANS4 = 125254
   829  003056  001401                      BEQ     .+4             ;SKIP IF SAME
   830  003060  104010                      HLT+8.                  ;ANS4 NOT = 125254
   831  003062  022767  000125  175722      CMP     #125,ANS5       ;IS ANS5 = 125
   832  003070  001401                      BEQ     .+4             ;SKIP IF SAME
   833  003072  104010                      HLT+8.                  ;ANS5 NOT = 125
   834  003074  022767  052525  175712      CMP     #52525,ANS6     ;IS ANS6 = 52525
   835  003102  001401                      BEQ     .+4             ;SKIP IF SAME
   836  003104  104010                      HLT+8.                  ;ANS6 NOT = 52525
   837  003106  022767  052525  175702      CMP     #52525,ANS7     ;IS ANS7 = 52525
   838  003114  001401                      BEQ     .+4             ;SKIP IF SAME
   839  003116  104010                      HLT+8.                  ;ANS7 NOT = 52525
   840  003120  022767  052524  175672      CMP     #52524,ANS8     ;IS ANS8 = 52524
   841  003126  001401                      BEQ     .+4             ;SKIP IF SAME
   842  003130  104010                      HLT+8.                  ;ANS8 NOT = 52524
   843  003132  000410                      BR      5$              ;SKIP JUNK
   844
   845  003134  040000  000000  000000  1$:     40000,0,0,0
   846  003142  000000
   847  003144  040252  125252  125252  2$:     40252,125252,125252,125252
   848  003152  125252
   849
   850  003154  005003                  5$:     CLR     R3              ;CLEAR
   851  003156  170003                          LDUB                    ;UBREAK REG.
   852
```

```
                                ;*****************************************************************
                                ;TEST 11        TEST OF LDSC FOR DIVD OF 71
                                ;*****************************************************************
003160  104400                  TST11:  SCOPE

003162  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT BITS
003166  012703  000314                  MOV     #314,R3         ;LOAD 314 INTO
003172  170003                          LDUB                    ;UBREAK REG,
003174  012704  000071                  MOV     #71,R4          ;LOAD 71 INTO
003200  170004                          LDSC                    ;SHIFT COUNT REG,
003202  172467  000206                  LDD     15,AC0          ;LOAD INTO AC0 - 40000,0,0,0
003206  174467  000212                  DIVD    25,AC0          ;DIVD BY 40252,125252,125252,125252
003212  170200                  STFPS   FPS             ;STORE FLOATING POINT STATUS
003214  170367  175602                  STST    FEC             ;STORE EXCEPTION CODES
003220  022700  140220                  CMP     #140220,FPS     ;CHECK FLOATING POINT STATUS
003224  001401                          BEQ     .+4             ;BRANCH IF OK
003226  104000                          HLT                     ;FPS NOT EQUAL TO 140220

003230  022767  000016  175564          CMP     #16,FEC         ;CHECK FLOATING EXCEPTION CODE
003236  001401                          BEQ     .+4             ;BRANCH IF OK
003240  104000                          HLT                     ;FEC NOT EQUAL TO 16

003242  170005                          STA0                    ;GET AR
003244  174867  175532                  STD     AC0,ANS1        ;GET FOR TYPING
003250  042767  177600  175524          BIC     #177600,ANS1    ;CLEAR JUNK
003256  170007                          STQ0                    ;GET QR
003260  174067  175526                  STD     AC0,ANS5        ;GET FOR TYPING
003264  042767  177600  175520          BIC     #177600,ANS5    ;CLEAR JUNK
003272  022767  000052  175502          CMP     #52,ANS1        ;IS ANS1 = 52
003300  001401                          BEQ     .+4             ;SKIP IF SAME
003302  104010                          HLT+8.                  ;ANS1 NOT = 52
003304  022767  125252  175472          CMP     #125252,ANS2    ;IS ANS2 = 125252
003312  001401                          BEQ     .+4             ;SKIP IF SAME
003314  104010                          HLT+8.                  ;ANS2 NOT = 125252
003316  022767  125252  175462          CMP     #125252,ANS3    ;IS ANS3 = 125252
003324  001401                          BEQ     .+4             ;SKIP IF SAME
003326  104010                          HLT+8.                  ;ANS3 NOT = 125252
003330  022767  125254  175452          CMP     #125254,ANS4    ;IS ANS4 = 125254
003336  001401                          BEQ     .+4             ;SKIP IF SAME
003340  104010                          HLT+8.                  ;ANS4 NOT = 125254
003342  022767  000014  175442          CMP     #14,ANS5        ;IS ANS5 = 14
003350  001401                          BEQ     .+4             ;SKIP IF SAME
003352  104010                          HLT+8.                  ;ANS5 NOT = 14
003354  022767  000000  175432          CMP     #0,ANS6         ;IS ANS6 = 0
003362  001401                          BEQ     .+4             ;SKIP IF SAME
003364  104010                          HLT+8.                  ;ANS6 NOT = 0
003366  022767  000000  175422          CMP     #0,ANS7         ;IS ANS7 = 0
003374  001401                          BEQ     .+4             ;SKIP IF SAME
003376  104010                          HLT+8.                  ;ANS7 NOT = 0
003400  022767  000000  175412          CMP     #0,ANS8         ;IS ANS8 = 0
003406  001401                          BEQ     .+4             ;SKIP IF SAME
003410  104010                          HLT+8.                  ;ANS8 NOT = 0
003412  000410                          BR      55              ;SKIP JUNK
```

```
   937
   938  003414  040000  000000  000000  1$:     40000,0,0,0
   939  003422  000000
   940  003424  040252  125252  125252  2$:     40252,125252,125252,125252
   941  003432  125252
   942
   943  003434  005003                  5$:     CLR     R3              ;CLEAR
   944  003436  170003                          LDUB                    ;UBREAK REG.
   945
   946
   947  003440  000167  001002                  JMP     DONE
```

```
  918  003444  005067  175330          TSTMUL: CLR     ICNT
  919  003450  104400                          SCOPE
  920  003452  004767  000456                  JSR     PC,GETTY        ;GET TTY INPUT
  921  003456  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT MODES
  922  003462  012703  000230                  MOV     #230,R3
  923  003466  170003                          LDUB
  924  003470  005004                          CLR     R4
  925  003472  012767  003510  002214          MOV     #1$,LAD
  926  003500  172767  175276                  LDD     ANS1,AC3        ;GET FIRST
  927  003504  172567  175302                  LDD     ANS5,AC1        ;GET SECOND
  928  003510  172603                  1$:     LDD     AC3,AC2         ;RESTORE
  929  003512  005204                          INC     R4              ;INC SHIFT COUNT
  930  003514  170004                          LDSC
  931  003516  171201                          MULD    AC1,AC2         ;MULTIPLY
  932  003520  170005                          STAB
  933  003522  174867  175254                  STD     AC0,ANS1        ;GET RESULT
  934  003526  170007                          STOB
  935  003530  174867  175256                  STD     AC0,ANS5        ;GET RESULT
  936
  937  003534  042767  177600  175240          BIC     #177600,ANS1
  938  003542  042767  177600  175242          BIC     #177600,ANS5
  939  003550  170006                          MRS
  940  003552  170006                          MRS
  941  003554  170005                          STAB
  942  003556  174867  000004                  STD     AC0,INPUT
  943  003562  006167  000600                  ROL     INPUT
  944  003566  006167  000574                  ROL     INPUT
  945  003572  042767  177477  000566          BIC     #177477,INPUT
  946  003600  056767  000562  175174          BIS     INPUT,ANS1
  947  003606  170007                          STOB
  948  003610  174867  000552                  STD     AC0,INPUT
  949  003614  006167  000546                  ROL     INPUT
  950  003620  006167  000542                  ROL     INPUT
  951  003624  042767  177477  000534          BIC     #177477,INPUT
  952  003632  056767  000530  175152          BIS     INPUT,ANS5
  953  003640  010400                          MOV     R4,R0
  954  003642  104010                          HLT+8.
  955  003644  022704  000071                  CMP     #71,R4
  956  003650  001002                          BNE     3$
  957  003652  105067  175123                  CLRB    ICNT+1
  958  003656  104400                  3$:     SCOPE
  959  003660  170127  040200                  LDFPS   #40200
  960  003664  172603                          LDD     AC3,AC2
  961  003666  171201                          MULD    AC1,AC2
  962  003670  170200                          STFPS   FPS
  963  003672  104010                          HLT+8.
  964  003674  000167  177544                  JMP     TSTMUL
```

```
   965  003700  005067  175074          TSTDIV: CLR     ICNT
   966  003704  104400                          SCOPE
   967  003706  004767  000222                  JSR     PC,GETTY        ;GET TTY INPUT
   968  003712  170127  040220                  LDFPS   #40220          ;DOUBLE/MAINT MODES
   969  003716  012703  000314                  MOV     #314,R3
   970  003722  170003                          LDUB
   971  003724  005004                          CLR     R4
   972  003726  012767  003744  001760          MOV     #1$,LAD
   973  003734  172767  175042                  LDD     ANS1,AC3        ;GET FIRST
   974  003740  172567  175046                  LDD     ANS5,AC1        ;GET SECOND
   975  003744  172603                  1$:     LDD     AC3,AC2         ;RESTORE
   976  003746  005204                          INC     R4              ;INC SHIFT COUNT
   977  003750  170004                          LDSC
   978  003752  174601                          DIVD    AC1,AC2         ;DIVIDE
   979  003754  170005                          STAB
   980  003756  174067  175020                  STD     AC0,ANS1        ;GET RESULT
   981  003762  170007                          STQB
   982  003764  174067  175022                  STD     AC0,ANS5        ;GET RESULT
   983
   984  003770  042767  177600  175004          BIC     #177600,ANS1
   985  003776  042767  177600  175006          BIC     #177600,ANS5
   986  004004  170006                          MFS
   987  004006  170006                          MFS
   988  004010  170005                          STAB
   989  004012  174067  000350                  STD     AC0,INPUT
   990  004016  006167  000344                  ROL     INPUT
   991  004022  006167  000340                  ROL     INPUT
   992  004026  042767  177477  000332          BIC     #177477,INPUT
   993  004034  056767  000326  174740          BIS     INPUT,ANS1
   994  004042  170007                          STQB
   995  004044  174067  000316                  STD     AC0,INPUT
   996  004050  006167  000312                  ROL     INPUT
   997  004054  006167  000306                  ROL     INPUT
   998  004060  042767  177477  000300          BIC     #177477,INPUT
   999  004066  056767  000274  174716          BIS     INPUT,ANS5
  1000  004074  013400                          MOV     R4,R0
  1001  004076  104010                          MLT+8.
  1002  004100  022704  000071                  CMP     #71,R4
  1003  004104  001002                          BNE     3$
  1004  004106  105067  174667                  CLRB    ICNT+1
  1005  004112  104400                  3$:     SCOPE
  1006  004114  170127  040200                  LDFPS   #40200
  1007  004120  172603                          LDD     AC3,AC2
  1008  004122  174601                          DIVD    AC1,AC2
  1009  004124  170200                          STFPS   FPS
  1010  004126  104010                          MLT+8.
  1011  004130  000167  177544                  JMP     TSTDIV
```

K2

```
1012  004134                          GETTY:
1013  004134  000004  004140                TYPE    ,.+2            ;.ASCIZ <15><12>"1) "
1014  004140  005027  000000          FLAG: CLR     #0
1015  004144  012702  001002                MOV     #ANS1,R2
1016  004150  004567  000106          5$:   JSR     R5,GOGET
1017  004154  004366                        INPUT
1018  004156  005012                        CLR     (2)
1019  004160  012701  004366                MOV     #INPUT,R1
1020  004164  122711  000015          3$:   CMPB    #15,(1)         ;IS IT A CR?
1021  004170  001420                        BEQ     1$
1022  004172  122711  000040                CMPB    #40,(1)         ;HOW ABOUT A SPACE
1023  004176  001411                        BEQ     2$
1024  004200  105011                        CLRB    (1)
1025  004202  105011                        CLRB    (1)
1026  004204  105011                        CLRB    (1)
1027  004206  006112                        ROL     (2)
1028  004210  105111                        ROLB    (1)
1029  004212  006112                        ROL     (2)
1030  004214  105121                        ROLB    (1)+
1031  004216  006112                        ROL     (2)
1032  004220  000761                        BR      3$
1033  004222  105721                2$:     TSTB    (1)+            ;TRY NEXT BYTE
1034  004224  005722                        TST     (2)+
1035  004226  005012                        CLR     (2)
1036  004230  000755                        BR      3$
1037  004232  005767  177764          1$:   TST     FLAG-2          ;FIRST TIME?
1038  004236  001010                        BNE     6$
1039  004240  000004  004254                TYPE    ,.+2            ;.ASCIZ "2) "
1040  004244  005722                        TST     (2)+
1041  004246  005167  177656                COM     FLAG-2
1042  004252  000734                        BR      5$
1043  004254  000207                6$:     RTS     PC
1044
1045  004256  012503                GOGET:  MOV     (5)+,R3         ;GET ADDRESS
1046  004260  105767  173262          1$:   TSTB    177560
1047  004264  100375                        BPL     .-4
1048  004266  116713  173256                MOVB    177562,(3)      ;GET CHARACTER
1049  004272  142713  000200                BICB    #200,(3)
1050  004276  111367  001374                MOVB    (3),.TYPE       ;SET UP FOR TYPING
1051  004302  000004  005710                TYPE    ,.TYPE
1052  004306  122713  000177                CMPB    #177,(3)
1053  004312  001411                        BEQ     2$
1054  004314  122723  000015                CMPB    #15,(3)+        ;CHECK FOR RETURN
1055  004320  001357                        BNE     1$
1056  004322  012767  000012  001346        MOV     #12,.TYPE       ;TYPE A 12
1057  004330  000004  005710                TYPE    ,.TYPE
1058  004334  000205                        RTS     R5
1059  004336  000004  004342          2$:   TYPE    ,QUES
1060  004342  016503  177776                MOV     -2(5),R3
1061  004346  000744                        BR      1$
1062  004350  006477  000012          QUES: .ASCIZ  "?"<15><12>
1063  004354  000030                INPUT:  .BLKW   30
```

```
1064  004446  104400                DONE:   SCOPE
1065  004450  032737  002000  177570        BIT     @SW10,@#SWR     ;RING THE BELL?
1066  004456  001005                BNE     1$              ;NO!
1067  004460  012767  000007  001222        MOV     #BELL,.TYPE     ;TYPE A BELL
1068  004466  000004  005710                TYPE    ,.TYPE
1069  004472  005046                1$:     CLR     -(6)            ;CLEAR TRACE TRAP
1070  004474  032737  010000  177570        BIT     @SW12,@#SWR     ;RUN WITH TRT?
1071  004502  001010                BNE     2$
1072  004504  005167  001202                COM     TRPB
1073  004510  100005                BPL     2$
1074  004512  052716  000020                BIS     #20,(6)         ;SET TRACE TRAP
1075  004516  012746  001062                MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
1076  004522  000412                BR      YESRT
1077  004524  012746  001062        2$:     MOV     #BEGIN,-(6)     ;JUMP TO START OF TEST
1078  004530  013700  000042                MOV     @#42,R0         ;GET MONITOR ADDRESS
1079  004534  001404                BEQ     3$              ;IF NONE
1080  004536  004710                JSR     7,(0)           ;GO TO MONITOR
1081  004540  000240                NOP
1082  004542  000240                NOP
1083  004544  000240                NOP
1084  004546  000002        3$:     RTI
1085  004550  000002        YESRT:  RTI                             ;RETURN TO PROGRAM FROM TRAP
1086
1087  004552  032737  000400  177570  .EMT:   BIT     @SW08,@#SWR     ;KILL LOOP OR LOOP ON SPEC. TEST
1088  004560  001404                BEQ     1$
1089  004562  123767  177570  174210        CMPB    @#SWR,ICNT      ;ON RIGHT TEST?   *SW7-0*
1090  004570  001434                BEQ     OVER
1091  004572  032737  040000  177570  1$:     BIT     @SW14,@#SWR     ;LOOP ON TEST
1092  004600  001026                BNE     KIT
1093  004602  032737  004000  177570        BIT     @SW11,@#SWR     ;KILL ITERATIONS
1094  004610  001012                BNE     SAVLAD
1095  004612  105767  174163                TSTB    ICNT+1
1096  004616  001404                BEQ     2$              ;BRANCH IF FIRST
1097  004620  126767  001074  174153        CMPB    TIMES,ICNT+1    ;DONE?
1098  004626  001013                BNE     KIT             ;BRANCH IF NOT
1099  004630  112767  000001  174143  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
1100  004636  105267  174136        SAVLAD: INCB    ICNT            ;COUNT TEST NUMBERS
1101  004642  011667  001046                MOV     (6),LAD         ;SAVE LOOP ADDRESS
1102  004646  016737  174126  177570        MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
1103  004654  000002                RTI                             ;RETURN
1104
1105  004656  105267  174117        KIT:    INCB    ICNT+1
1106  004662  016737  174112  177570  OVER:   MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
1107  004670  005767  001020                TST     LAD             ;FIRST ONE?
1108  004674  001760                BEQ     SAVLAD
1109  004676  016716  001012                MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
1110  004702  000002                RTI                             ;FIXES PS
```

```
1111  005724  032737  002000  177570  .TRP:   BIT     #SW17,@#SWR     ;BELL ON ERROR?
1112  005732  001405                          BEQ     1$              ;NO - SKIP
1113  005734  012767  000007  000766          MOV     #BELL,.TYPE     ;TYPE A BELL
1114  005742  007004  005710                  TYPE    ,.TYPE
1115  005746  005267  000764          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
1116  005752  010446                          MOV     R4,-(6)
1117  005754  032737  020000  177570          BIT     #SW13,@#SWR     ;SKIP TYPEOUT IF SET
1118  006002  001072                          BNE     4$
1119  006004  007004  005656                  TYPE    ,RETURN
1120  006010  016646  000002                  MOV     2(6),-(6)       ;PUT ADDRESS OF INSTRUCTION ON STACK
1121  006014  162716  000002                  SUB     #2,(6)
1122  006020  011605                          MOV     (6),TTY         ;TYPE (6) IN OCTAL
1123  006022  004767  000156                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1124  006026  007004  005664                  TYPE    ,SPACE+3
1125  006032  010005                          MOV     R0,TTY          ;TYPE R0 IN OCTAL
1126  006034  004767  000144                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1127  006040  007004  005665                  TYPE    ,SPACE+4
1128  006044  012703  001002                  MOV     #ANS1,R3        ;ADDRESS OF DATA
1129  006050  113604                          MOVB    @(6)+,R4        ;AMOUNT OF DATA IN TABLE
1130  006052  001426                          BEQ     3$
1131  006054  100016                          BPL     2$              ;TYPE STACK?
1132  006056  016667  000006  173756          MOV     6(6),ANS1
1133  006064  016667  000010  173752          MOV     10(6),ANS2
1134  006072  016667  000012  173746          MOV     12(6),ANS3
1135  006100  016667  000014  173742          MOV     14(6),ANS4
1136  006106  042704  177600                  BIC     #177600,R4      ;CLEAR SIGN
1137  006112  007004  005665          2$:     TYPE    ,SPACE+4
1138  006116  012305                          MOV     (3)+,TTY        ;TYPE (3)+ IN OCTAL
1139  006120  004767  000060                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1140  006124  005304                          DEC     R4
1141  006126  001371                          BNE     2$
1142  006130  005700                  3$:     TST     FPS
1143  006132  100016                          BPL     4$
1144  006134  007004  005661                  TYPE    ,SPACE
1145  006140  170367  173716                  STST    FEC
1146  006144  016705  173712                  MOV     FEC,TTY         ;TYPE FEC IN OCTAL
1147  006150  004767  000030                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1148  006154  007004  005664                  TYPE    ,SPACE+3
1149  006160  016705  173700                  MOV     FEA,TTY         ;TYPE FEA IN OCTAL
1150  006164  004767  000014                  JSR     %7,PRINTR       ;TYPE LEADING ZERO'S
1151  006170  012604                  4$:     MOV     (6)+,R4
1152  006172  005737  177570                  TST     @#SWR           ;HALT ON ERROR
1153  006176  100001                          BPL     .+4             ;SKIP IF CONTINUE
1154  006200  000000                          HALT                    ;HALT ON ERROR!
1155  006202  000002                          RTI
```

```
1156  005144  112767  000001  000130  PRINTR: MOVB    #1,A4$          ;SET ZERO FILL SWITCH
1157  005152  000402                  BR      .+6
1158  005154  005067  000122          PRINTS: CLR     A4$             ;SUPRESS LEADING ZERO'S
1159  005160  112767  177772  000115          MOVB    #-6,A4$+1       ;SET COUNT
1160  005166  010446                  MOV     R4,-(6)         ;SAVE R4
1161  005170  012704  005272                  MOV     #3$,R4          ;SET POINTER TO FIRST ASCII CHAR,
1162  005174  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
1163  005176  000405                  BR      2$              ;ROTATE FIRST BIT
1164  005200  105014                  1$:     CLRB    (4)             ;CLEAR BYTE OF CHARACTER
1165  005202  006105                  ROL     TTY             ;ROTATE BIT INTO C
1166  005204  106114                  ROLB    (4)             ;PACK IT
1167  005206  006105                  ROL     TTY             ;ROTATE BIT INTO C
1168  005210  106114                  ROLB    (4)             ;PACK IT
1169  005212  006105                  2$:     ROL     TTY             ;ROTATE BIT INTO C
1170  005214  106114                  ROLB    (4)             ;PACK IT
1171  005216  105714                  TSTB    (4)
1172  005220  001402                  BEQ     .+6
1173  005222  105267  000054          INCB    A4$
1174  005226  105767  000050          TSTB    A4$             ;CHECK FILL SWITCH
1175  005232  001402                  BEQ     .+6
1176  005234  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
1177  005240  105267  000037          INCB    A4$+1
1178  005244  001355                  BNE     1$              ;REPEAT
1179  005246  022704  005272          CMP     #3$,R4
1180  005252  001002                  BNE     .+6
1181  005254  112724  000060          MOVB    #'0,(4)+
1182  005260  105014                  CLRB    (4)
1183  005262  000004  005272          TYPE    ,3$             ;TYPE IT
1184  005266  012604                  MOV     (6)+,R4         ;RESTORE R4
1185  005270  000207                  RTS     PC
1186
1187  005272  000004                  3$:     .BLKW   4
1188  005302  000000                  A4$:    0
1189
1190  005304  052525  052525  052525  ALT1A:  52525,52525,52525,52525
1191  005312  052525
1192  005314  125252  125252          ALT1B:  125252,125252
1193  005320  125252  125252  052525  ALT1C:  125252,125252,52525,52525
1194  005326  052525
1195  005330  040125  052525  052525  ALTA:   40125,52525,52525,52525
1196  005336  052525
1197  005340  040052  125252  125252  ALTB:   40052,125252,125252,125252
1198  005346  125252
1199  005350  000125  052525  052525  ANSA:   125,52525,52525,52525
1200  005356  052525
1201  005360  000052  125252  125252  ANSB:   52,125252,125252,125252
1202  005366  125252
1203
```

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

```
1253  005570  010546                  .IOT:   MOV     TTY,-(6)          ;SAVE TTY
1254  005572  017605  000002                  MOV     @2(6),TTY         ;GET ADDRESS TO BE TYPED
1255  005576  105715                  1$:     TSTB    (TTY)             ;TERMINATOR?
1256  005600  001406                          BEQ     2$
1257  005602  112537  177566                  MOVB    (TTY)+,@#177566   ;LOAD AND TYPE THE CHARACTER
1258  005606  105737  177564                  TSTB    @#177564          ;IS THE PRINTER READY
1259  005612  100375                          BPL     .-4
1260  005614  000770                          BR      1$                ;GET THE NEXT CHARACTER
1261  005616  017646  000002          2$:     MOV     @2(6),-(6)        ;GET ADDRESS TO BE TYPED
1262  005622  062766  000002  000004          ADD     #2,4(6)           ;ADD 2 TO THE ADDRESS
1263  005630  022666  000002                  CMP     (6)+,2(6)         ;IS IT .+2?
1264  005634  001006                          BNE     3$                ;NO
1265  005636  062705  000002                  ADD     #2,TTY            ;ADD 2 TO THE ADDRESS
1266  005642  042705  000001                  BIC     #1,TTY            ;BACK UP TO AN EVEN BYTE
1267  005646  010566  000002                  MOV     TTY,2(6)          ;RESTORE ADDRESS
1268  005652  012605                  3$:     MOV     (6)+,TTY          ;RESTORE TTY
1269  005654  000002                          RTI                       ;RETURN
1270
1271  005656  005015    000           RETURN: .ASCIZ  <15><12>          ;RETURN AND LINEFEED
1272  005661    015     020012  020040  SPACE:  .ASCIZ  <15><12>"   "     ;RETURN AND 3 SPACES
1273  005666    000
1274          005670                    .EVEN
1275  005670  000000                    SAVE6:  0
1276  005672  172160                    FPTADR: 172160                    ;FLOATING POINT ADDRESS ON THE 11/20
1277  005674  000244  000246           FPVECT: 244,246                   ;FLOATING POINT VECTOR ADDRESS
1278  005700  000024  000026           DWNVEC: 24,26                     ;POWER DOWN VECTOR ADDRESS
1279  005704  000024  000026           UPVEC:  24,26                     ;POWER UP VECTOR ADDRESS
1280  005710  000000                    .TYPE:  0
1281  005712  000000                    TRPB:   0
1282  005714  000000                    LAD:    0                         ;LOOP ADDRESS
1283  005716  000000                    ERRORS: 0                         ;ERROR COUNT
1284  005720  000377                    TIMES:  377                       ;ITERATION COUNT
1285          000001                    .END
```

DCFPM.P11 CROSS REFERENCE TABLE

```
AC0     =%000000        403#  481*  484*  495   496   512*  523   524   538*  541*  553   565   953*
                        596*  608   624   644*  645*  648   656   675*  676*  688   691   738*  739*
                        751   754   801*  882*  814   817   864*  865*  877   888   933   935   942
                        948   980   982   989   995   1208  1212*  1213  1214*  1215  1237*  1238  1239*
                       1240   1244*
AC1     =%000001        484#  927*  931   961   974*  978   1008  1209  1243*
AC2     =%000002        485#  928*  931*  960*  961*  975*   978*  1007*  1008*  1210  1242*
AC3     =%000003        486#  926*  928   960   973*  975   1007  1211  1241*
AC4     =%000004        487#  1212  1240*
AC5     =%000005        488#  1214  1238*
ALTA      005330        538   596   644   1195#
ALTB      005340        541   593   645   1197#
ALT1A     005304        481   496   1190#
ALT1B     005314        484   512   1192#
ALT1C     005320        524   1193#
ANSA      005350       1199#
ANSB      005360       1201#
ANS1      001002        437#  495*  523*  553*  554*  555   569*  578*  571   688*  689*  618   624*
                        625*  626   648*  650*  651   656*  657*  658   688*  689*  693   751*  752*
                        756   814*  815*  819   877*  878*  882   926   933*  937*  946*  973   988*
                        984*  993*  1015  1128  1132*
ANS2      001004        438#  558   574   613   629   696   759   822   885   1133*
ANS3      001006        439#  561   577   616   699   762   825   888   1134*
ANS4      001010        440#  564   580   619   702   765   828   891   1135*
ANS5      001012        441#  691*  692*  785   754*  755*  768   817*  818*  831   885*  881*  854
                        927   935*  938*  952*  974   982*  985*  999*
ANS6      001014        442#  708   771   834   897
ANS7      001016        443#  711   774   837   900
ANS8      001020        444#  714   777   840   903
A4S       005302       1156*  1158*  1159*  1173*  1174   1177*  1188#
BEG       001026        425   448#
BEGIN     001062        452   457#  1075  1077
BELL    = 000007        392#  1067  1113
DISPLA  = 177570        388#  1102*  1106*
DONE      004446        917   1064#
DWNVEC    005700        460*  461*  1246*  1247*  1278#
ERRORS    005716       1115*  1283#
FEA       001024        446#  1149
FEC       001022        431*  445#  486*  491   514*  519   543*  548   598*  683   678*  683   741*
                        746   884*  889   867*  872   1145*  1146
FLAG      004146       1014#  1037  1041*
FLTERR    000760        438#  468
FPS     =%000040        393#  430*  485*  487   513*  515   542*  544   597*  599   649*  677*  679
                        740*  742   883*  885   866*  868   962*  1089*  1142
FPTADR    005672        454   1276#
FPVECT    005674        460*  469*  1277#
GETTY     004134        920   967   1012#
GOGET     004270       1016   1045#
HLT     = 104000        398#  489   493   499   517   521   527   546   558   557   56_   563   566
                        573   576   579   582   601   605   612   615   618   621   628   631   653
                        660   681   685   695   698   701   704   707   710   713   716   744   748
                        758   761   764   767   770   773   776   779   807   811   821   824   827
                        838   833   836   839   842   870   874   884   887   890   893   896   869
                        902   905   954   963   1001  1010
```

```
ICNT      001000          436#    470*    918*    957*    965*   1004*   1089    1095    1097    1099*   110.*   1182    1185*
                         1186
ILLUP     005564         1284    1251#
INPUT     004366          942*    943*    944*    945*    946     948*    949*    950*    951*    952     989*    990*    991*
                          992*    993     995*    996*    997*    998*    999    1017    1019    1063#
KIT       004656         1092    1098    1105#
LAD       005714          471*    925*    972*   1181*   1187    1189    1282#
LDSC    = 170004          421#
LDUB    = 170003          417#
MRS     = 170006          420#    646     939     940     986     987
M1120     001054          449     454#
N       = 000012          385#    475     479#    584     588#    532     536#    587     591#    636     64.#    665     669*
                          728     732#    791     795#    854     858#
OVER      004662         1098    1106#
PC      =%000007          402#    920*    967*   1043*   1185*
POWDWN    005370           46#   1284#   1246
POWUP     005460         1223    1226#
PRINTR    005144         1123    1126    1139    1147    1150    1156#
PRINTS    005154         1158#
PS      = 177776          386#
QUES      004362         1059    1062#
RETURN    005656         1119    1271#
R0      =%000000          394#    463*    953*   1008*   1078*   1125    1216    1235*
R1      =%000001          395#   1019*   1217    1227*   1228*   1234*
R2      =%000002          396#   1015*   1218    1233*
R3      =%000003          397#    473*    482*    581*    510*    529*    539*    584*    594*    633*    642*    662*    671*
                          724*    734*    787*    797*    850*    860*    913*    922*    969*   1045*   1063*   1128*   1219
                         1232*
R4      =%000004          398#    673*    736*    799*    862*    924*    929*    953     955     971*    976*    1000    1002
                         1116    1129*   1136*   1148*   1151*   1160    1161*   1179    1184*   1220    1231*
R5      =%000005          399#   1016*   1058*   1221    1230*
SAVE6     005670         1222*   1226    1275#
SAVLAD    004636         1094    1100#   1108
SCOPE   = 104400          389#    478     507     535     590     639     668     731     794     857     919     958     986
                         1005    1064
SP      =%000006          401#    448*    458*   1222    1226*
SPACE     005661         1124    1127    1137    1144    1148    1272#
STA0    = 170005          418#
STQ0    = 170007          419#
SWR     = 177570          387#    388     1065    1070    1087    1089    1091    1093    1111    1117    1152
SW08    = 000400          416#   1087
SW09    = 001000          415#
SW10    = 002000          414#   1065    1111
SW11    = 004000          413#   1093
SW12    = 010000          412#   1070
SW13    = 020000          411#   1117
SW14    = 040000          410#   1091
SW15    = 100000          409#
TIMES     005720         1097    1284#
TRPB      005712         1072*   1281#
TSTDIV    003700          427     965#   1011
TSTMUL    003444          426     918#    964
TST1      001204          478#
TST10     002700          794#
```

```
TST11    003160          857#
TST2     001302          587#
TST3     001374          535#
TST4     001624          590#
TST5     002030          639#
TST6     002140          668#
TST7     002420          731#
TTY     =%000005         488#  1122*  1125*  1138*  1146*  1149*  1165*  1167*  1169*  1253  1254*  1255  1257
                        1265*  1266*  1267   1268*
TYPE    = 000004         391#  1013   1039   1051   1057   1059   1068   1114   1119   1124  1127   1137  1144
                        1148   1183   1248
UPVEC    005704         1284*  1285*  1223*  1279#
YESRT    004550          451*   459   1076   1085#
.       = 005722         423#   424#   429#   434#   488   492   498   516   528   526   545   549   556
                         559    562    565    572    575   578   581   600   604   611   614   617   620
                         627    630    652    659    680   684   694   697   700   703   706   709   712
                         715    743    747    757    760   763   766   769   772   775   778   806   810
                         820    823    826    829    832   835   838   841   869   873   883   886   889
                         892    895    898    901    904   1013  1039  1047  1063#  1153  1157  1172  1175
                        1188   1187#  1248   1252   1259  1274#
.EMT     004552          466   1087#
.IOT     005570          462   1253#
.TRP     004704          464   1111#
.TYPE    005710         1050*  1051   1056*  1057   1067*  1068   1113*  1114   1200#
```

```
NOP      1081  1082  1083
ROL       943   944   949   950   990   991   996   997  1027  1029  1031  1165  1167  1169
ROLB     1028  1030  1166  1168  1170
RORB     1024  1025  1026
RTI       433  1084  1085  1103  1110  1155  1249  1269
RTS      1043  1058  1105
SETD     1207  1236
STA0      552   623   655   687   750   813   876   932   941   979   988
STD       495   523   553   569   648   656   688   691   751   754   814   817   877   880   933
          935   942   948   980   982   989   995  1208  1209  1210  1211  1213  1215  1238  1242
STF       608   624
STFPS     430   485   513   542   597   649   677   740   803   866   962  1089  1206
STQ0      568   607   647   690   753   816   879   934   947   981   994
STST      431   486   514   543   598   678   741   804   867  1145
SUB      1121
TRAP      389
TST       450  1034  1037  1040  1107  1142  1152
TST0     1033  1046  1095  1171  1174  1255  1258
.ASCII   1014  1040  1062  1249  1271  1272
.BLKW    1063  1187
.ENABL    384
.END     1285
.ENDC     487   495   515   523   544   552   599   687   672   679   687   735   742   750   758
          805   813   861   868   876
.EVEN    1014  1040  1249  1274
.IF       486   514   543   598   671   678   734   741   797   804   860   867
.IFF      672   735   797   860
.IFNZ     491   495   519   523   548   552   603   607   683   687   746   750   809   813   872
          876
.LIST     344   384   424   434   475   479   584   588   532   536   587   591   636   640   665
          669   728   732   791   795   854   858  1014  1040  1064  1111  1156  1204  1249  1253
.MACR     384
.MACRO    384
.NLIST    344   384   424   434   475   479   584   588   532   536   587   591   636   640   665
          669   728   732   791   795   854   858  1014  1040  1064  1111  1156  1204  1249  1253
.REM      345
.REPT       2   424
.SBTTL    344   384   434   475   584   532   587   636   665   720   791   854  1064  1111  1156
         1204  1253
.TITLE    343
```

ERRORS DETECTED: 0

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