```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E742G-MC

PRODUCT NAME: CXFPAG0 DEC/X11 FP11 (11/40 & 45) MODULE

DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT:

FPA IS A BKMOD THAT EXERCISES THE FP11-C
AND THE FIS OPTION IN THE 11/40 OR LSI-11.

2. REQUIREMENTS:

HARDWARE: PDP11/45 WITH THE FPP INSTRUCTION SET, OR AN 11/40 WITH FIS.
STORAGE:: FPA REQUIRES:
1. DECIMAL WORDS: 360
2. OCTAL WORDS: 0550
3. OCTAL BYTES: 1320

3. PASS DEFINITION:

ONE PASS OF THE FPA MODULE CONSISTS OF EXECUTING EACH
INSTRUCTION 25,000 TIMES.

4. EXECUTION TIME:

FPA MODULE RUNNING ALONE ON A PDP11 PROCESSOR TAKES
APPROXIMATELY 30 SECONDS.

5. CONFIGURATION REQUIREMENTS:

SET SR1 EQUAL TO A 1 IF AN 11/40 IS USED OR EQUAL TO 0
IF AN 11/45 IS USED.

6. DEVICE/OPTION SETUP:

MAKE SURE FP11-C OR FIS IS INSTALLED.

7. MODULE OPERATION:

A. SETUP CYCLE COUNTER.
B. TEST SR1,IF EQUAL TO 1 GO TO D.
C. ELSE TEST ALL FPP INSTRUCTIONS, GO TO E.
D. TEST ALL FIS INSTRUCTIONS.
E. IF NOT EOP, GO TO B.
F. ELSE DO EOP, GO TO A.

8. OPERATING OPTIONS:

SR1=0 11/45,11/70 FP11-C OPTION INSTRUCTIONS
SR1=1 11/40 OR LSI ,FIS OPTION INSTRUCTIONS

9. NON-STANDARD PRINTOUTS:

NONE

```
000000'  -          PKMOD <FPAG >,,,,,25000,16
000000'             MODULE  40020,FPAG ,,,,,25000,16
                    .TITLE  FPAG DEC/X11 SYSTEM EXERCISER MODULE
                ;   DDXCOM  VERSION  6      23-MAY-78
                                    .LIST   BIN
                                    ;*****************************************************************************
000000'                             BEGIN:
000000' 050106  043501      040     MODNAM: .ASCII  /FPAG / ;MODULE NAME.
000005'    000                      XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000000                      ADDR:   +0                      ;1ST DEVICE ADDR.
000010' 000000                      VECTOR: +0                      ;1ST DEVICE VECTOR.
000012'    000                      BR1:    .BYTE   PRTY+0          ;1ST BR LEVEL.
000013'    000                      BR2:    .BYTE   PRTY+0          ;2ND BR LEVEL.
000014' 000001                      DVID1:  +1                      ;DEVICE INDICATOR 1.
000016' 000000                      SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                      SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                      SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                      SR4:    OPEN                    ;SWITCH REGISTER 4
                                    ;*****************************************************************************
000026' 040020                      STAT:   40020                   ;STATUS WORD.
000030' 000224'                     INIT:   START                   ;MODULE START ADDR.
000032' 000224'                     SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                      PASCNT: 0                       ;PASS COUNTER.
000036' 025000                      ICONT:  25000                   ;# OF ITERATIONS PER PASS=25000
000040' 000000                      ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                      SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                      HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                      SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                      HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                      SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                      RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                             CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                      RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                      RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                      SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                      SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                      SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                      SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                      SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                      SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                      SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                      CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                             SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                      ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                             WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                      ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                             ERRTYP:                         ;TYPE OF ERROR
000106' 000000                      ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                      AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000224'                     RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                      WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                      WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                      INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000016                      IDNUM:  16                      ;MODULE IDENTIFICATION NUMBER=16
        000040                              .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```



```
                                    .NLIST
                                    .WORD   0
                                    .LIST
                                    .ENDR
000224'                             MODSP:
                                    ;*****************************************************************************
```

```
                    000000                  FPS=    %0
                    000000                  AC0=    %0
                    000001                  AC1=    %1
                    000002                  AC2=    %2
                    000003                  AC3=    %3
                    000004                  AC4=    %4
                    000005                  AC5=    %5
000224'                                     START:
000224'                                     RESTRT:
000224' 005767  177566                      TST     SR1             ;TEST FOR AN 11/40, OR 11/45
000230' 001402                              BEQ     START1          ;11/45
000232' 000167  000552                      JMP     START2          ;11/40
000236' 170127  047400              START1: LDFPS   #47400
000242' 170011                              SETD                    ;SET DOUBLE MODE
000244' 177027  077777                      LDCID   #77777,AC0              ;LOAD AC0 WITH 77777
000250' 177127  002525                      LDCID   #2525,AC1               ;LOAD AC1 WITH 2525
000254' 012702  000012                      MOV     #12,R2                  ;SET STEP COUNTER
000260' 174401                      EXLOP:  DIVD    AC1,AC0                 ;DIVIDE 2525 INTO 77777
000262' 171001                              MULD    AC1,AC0                 ;MULTIPLY 2525 BY ANSWER
000264' 172001                              ADDD    AC1,AC0                 ;ADD 2525 TO ANSWER
000266' 173001                              SUBD    AC1,AC0                 ;SUBTRACT 2525 FROM ANSWER
000270' 005302                              DEC     R2                      ;DO 10 (DECIMAL) TIMES
000272' 001372                              BNE     EXLOP
000274' 175467  000776                      STCDI   AC0,ANS1                ;SAVE ANSWER
000300' 170200                              STFPS   FPS             ;SAVE FPS
000302' 022767  077777  000766              CMP     #77777,ANS1             ;IS RESULT CORRECT
000310' 001403                              BEQ     EX4
                                            ;*****************************************************************
000312' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000320' 172467  000242              EX4:    LDD     D1010,AC0
000324' 012702  001276'                     MOV     #ANS1,R2
000330' 174022                              STD     AC0,(R2)+
000332' 172567  000230                      LDD     D1010,AC1
000336' 174122                              STD     AC1,(R2)+
000340' 172767  000742                      LDD     ANS5,AC3
000344' 170200                              STFPS   FPS
000346' 173767  000724                      CMPD    ANS1,AC3
000352' 170000                              CFCC
000354' 001403                              BEQ     1$
                                            ;*****************************************************************
000356' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000364' 172567  000200              1$:     LDD     D0101,AC1
000370' 174142                              STD     AC1,-(R2)
000372' 172012                              ADDD    (R2),AC0
000374' 173042                              SUBD    -(R2),AC0
000376' 173001                              SUBD    AC1,AC0
000400' 175467  000672                      STCDI   AC0,ANS1
000404' 022767  000000  000664              CMP     #0,ANS1
000412' 170200                              STFPS   FPS
000414' 001403                              BEQ     MORE
                                            ;*****************************************************************
000416' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000424' 170001                      MORE:   SETF
```



```
000426' 177027  000525                      LDCIF   #525,AC0
000432' 177127  000252                      LDCIF   #252,AC1
000436' 174104                              STF     AC1,AC4
000440' 172104                              ADDF    AC4,AC1
000442' 172701                              LDF     AC1,AC3
000444' 173003                              SUBF    AC3,AC0
000446' 175067  000624                      STEXP   AC0,ANS1
000452' 170200                              STFPS   FPS
000454' 022767  000001  000614              CMP     #1,ANS1
000462' 001403                              BEQ     1$
                                            ;*****************************************************************
000464' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000472' 177027  000021              1$:     LDCIF   #21,AC0
000476' 171000                              MULF    AC0,AC0
000500' 174427  040400                      DIVF    #2,AC0
000504' 171427  040200                      MODF    #1,AC0
000510' 170200                              STFPS   FPS
000512' 175567  000560                      STCFI   AC1,ANS1
000516' 022767  000220  000552              CMP     #220,ANS1
000524' 001403                              BEQ     2$
                                            ;*****************************************************************
000526' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000534' 171427  041040              2$:     MODF    #10.,AC0
000540' 175567  000532                      STCFI   AC1,ANS1
000544' 022767  000005  000524              CMP     #5,ANS1
000552' 001403                              BEQ     3$
                                            ;*****************************************************************
000554' 104405  000000' 000000              ARDERS,BEGIN,NULL       ;
                                            ;*****************************************************************
000562' 000167  000436              3$:     JMP     CYCLE
000566' 177777                      D1010:  -1
000570' 000000  177777  000000      D0101:  0,-1,0
000576' 177777  000000  000000      D1001:  -1,0,0
000604'                             DSMALL:
000604' 177777  000001  000000      WEIRD:  -1,1,0,-1
000612' 177777
000614' 000570'                     AD0101: D0101
000616' 000576'                     AD1001: D1001
000620' 000736'                     AD1000: D1000
000622' 077777  000000  177777      DBIG:   77777,0,-1,0
000630' 000000
000632' 100000  000000  000000      DMZERO: 100000,0,0,0
000640' 000000
000642' 001276'                     AANS1:  ANS1
000644' 040252  125252  125252      DALTA:  40252,125252,125252,125252
000652' 125252
000654' 040325  052525  052525      DALTB:  40325,52525,52525,52525
000662' 052525
000664' 040325  052525  052525      DALTC:  40325,52525,52525,52526
000672' 052526
000674' 040000  000000  000000      D40:    40000,0,0,0
000702' 000000
000704' 037400  000000  000000      D37:    37400,0,0,0
000712' 000000
```

```
                                                ;11/40 FIS FLOATING ADD
                                                ;11/40 FIS FLOATING SUBTRACT
                                                ;11/40 FIS FLOATING MULTIPLY
                                                ;11/40 FIS FLOATING DIVIDE
                                                ;SET STACK POINTER
                                                ;LOAD DATA ONTO STACK
                                                ;CLEAR A LOCATION FOR PSW
                                                ;IF C SET,LEAVE ANS3=0
                                                ;IF V SET,LEAVE ANS3=0
                                                ;IF Z SET,LEAVE ANS3=0
                                                ;IF N = 0,LEAVE ANS3=0
                                                ;IF ABOVE NOT TRUE,THEN PS=10
                                                ;AND N=1,Z=V=C=0,SO MAKE ANS3=10
                                                ;SAVE FIRST HALF OF ANSWER
                                                ;SAVE SECOND HALF OF ANSWER
                                                ;SAVE FINAL STACK POINTER
                                                ;CHECK FINAL PSW
```



```
                                                ;*****************************************************************
001144' 104405  000000' 000000                  ERDER$,BEGIN,NULL       ;CONDITION CODES INCORRECT
                                                ;*****************************************************************
001152' 022767  001274' 000124  1$:     CMP     #FISSTK,ANS4            ;CHECK STACKPOINTER POSITION
001160' 001403                          BEQ     2$
                                                ;*****************************************************************
001162' 104405  000000' 000000                  ERDER$,BEGIN,NULL       ;
                                                ;*****************************************************************
001170' 022767  137201  000100  2$:     CMP     #137201,ANS1            ;CHECK FIRST HALF OF ANSWER
001176' 001403                          BEQ     3$
                                                ;*****************************************************************
001200' 104405  000000' 000000                  ERDER$,BEGIN,NULL       ;
                                                ;*****************************************************************
001206' 022767  115230  000064  3$:     CMP     #115230,ANS2            ;CHECK SECOND HALF OF ANSWER
001214' 001403                          BEQ     4$
                                                ;*****************************************************************
001216' 104405  000000' 000000                  ERDER$,BEGIN,NULL       ;
                                                ;*****************************************************************
001224'                                 4$:

                                        ;+
                                        ; CHECK FOR END OF PASS
                                        ;-

001224'                                 CYCLE:
001224' 104413  000000'                         ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                        ;MONITOR SHALL TEST END OF PASS
001230' 000167  176770                          JMP     RESTRT          ;CONTINUE TESTING


                001274'                 .=.+40
001274' 000000                  FISSTK: 0
001276' 000000                  ANS1:   0                               ;FIPST ANSWER (SEE CODE)
001300' 000000                  ANS2:   0
001302' 000000                  ANS3:   0
001304' 000000                  ANS4:   0
001306' 000000                  ANS5:   0
001310' 000000                  ANS6:   0
001312' 000000                  ANS7:   0
001314' 000000                  ANS8:   0
001316' 000000                  CNT:  0


                000001                  .END
```

```
AANS1   000642R  261#
ACSR    000102R  142#
ADDR    000006R  108#
ADDR22= 001000   160#
AD1001  000614R  254#
AD1000  000620R  256#
AD1001  000616R  255#
AD5T01  000770R  283#
ANS1    001276R  183*  185   191   197   208*  209   222*  224   234*  235   241*  242   261
                 322*  337   364#
ANS2    001300R  323*  342   365#
ANS3    001302R  310*  319*  325   366#
ANS4    001304R  324*  331   367#
ANS5    001306R  195   368#
ANS6    001310R  369#
ANS7    001312R  370#
ANS8    001314R  371#
ASB     000106R  146#
ASTAT   000104R  144#
AWAS    000110R  147#
BEGIN   000000R  105#  188   201   213   227   238   245   329   334   340   345   354
BIT0  = 000001   160#
BIT1  = 000002   160#
BIT10 = 002000   160#
BIT11 = 004000   160#
BIT12 = 010000   160#
BIT13 = 020000   160#
BIT14 = 040000   160#
BIT15 = 100000   160#
BIT2  = 000004   160#
BIT3  = 000010   160#
BIT4  = 000020   160#
BIT5  = 000040   160#
BIT6  = 000100   160#
BIT7  = 000200   160#
BIT8  = 000400   160#
BIT9  = 001000   160#
BREAK$= 104407   160#
BR1     000012R  110#
BR2     000013R  111#
BTOD$ = 104421   160#
CDATA$= 104412   160#
CNT     001316R  372#
CONFIG  000056R  130#
CSRA    000100R  140#
CYCLE   001224R  247   353#
DALTA   000644R  262#
DALTB   000654R  264#
DALTC   000664R  266#
DATCK$= 104411   160#
DATER$= 104404   160#
DBIG    000622R  257#
DMZERO  000632R  259#
DSMALL  000604R  251#
DVID1   000014R  112#
D0100   000734R  276#
```



```
D0100X  000746R  279#
D0101   000570R  203   249#  254
D0111   000754R  280#
D1000   000736R  256   277#
D1001   000576R  250#  255
D1010   000566R  190   193   248#
D20     000724R  274#
D37     000704R  270#
D40     000674R  268#
D46     000714R  272#
D5T01   000764R  282#  283
ENDIT$= 104413   160#  354
END$  = 104410   160#
ERRTYP  000106R  145#
EXIT$ = 104400   160#
EXLOP   000260R  177#  182
EX4     000320R  186   190#
FADD  = 075000   291#
FDIV  = 075030   294#
FISSTK  001274R  298   331   363#
FLTERR  001006R  288#
FMUL  = 075020   293#
FSUB  = 075010   292#
F5T01   000772R  284#
GETPA$= 104415   160#
GWBUF$= 104414   160#
HRDCNT  000044R  125#
HRDER$= 104405   160#  188   201   213   227   238   245   329   334   340   345
HRDPAS  000050R  127#
ICONT   000036R  122#
ICOUNT  000040R  123#
IDNUM   000122R  152#
INIT    000030R  119#
INTR    000120R  151#
MAP22$= 104416   160#
MODNAM  000000R  106#
MOSP    000224R  120   158#
MORE    000424R  211   215#
MSGN$ = 104403   160#
MSGS$ = 104402   160#
MSG$  = 104401   160#
NULL  = 000000   160#  188   201   213   227   238   245   329   334   340   345
OPEN  = 000000   107   113   114   115   116   133   134   135   136   137   138   139   140
                 142   144   146   147   149   150   151   160#

OTOA$ = 104420   160#
PASCNT  000034R  121#
PIRQ$ = 000004   160#
POPSP = 005726   160#
POPSP2= 022626   160#
PRTV  = 000000   110   111   160#
PRTV0 = 000000   160#
PRTV1 = 000040   160#
PRTV2 = 000100   160#
PRTV3 = 000140   160#
PRTV4 = 000200   160#
PRTV5 = 000240   160#
```

```
PRTY6 = 000300              160#
PRTY7 = 000340              160#
PS    = 177776              160#
PSW   = 177776              160#
PUSH  = 005746              160#
PUSH2 = 024646              160#
RANDS = 104417              160#
RANNUM  000054R             129#
RESTRT  000224R             148      168#    356
RES1    000056R             131#
RES2    000060R             132#
RSTRT   000112R             146#
SAVADR  001002R             286#
SAVSTS  001004R             287#
SBADR   000102R             141#
SOFCNT  000042R             124#
SOFERS= 104406              160#
SOFPAS  000046R             126#
SPOINT  000032R             120#
SPSIZ = 000040                1#    153
SR1     000016R             113#    169
SR2     000020R             114#
SR3     000022R             115#
SR4     000024R             116#
START   000224R             119     167#
START1  000236R             170     172#
START2  001010R             171     298#
STAT    000026R             118#
SVR0    000062R             133#
SVR1    000064R             134#
SVR2    000066R             135#
SVR3    000070R             136#
SVR4    000072R             137#
SVR5    000074R             138#
SVR6    000076R             139#
SYSCNT  000052R             128#
TRPDFD= 000022              160#
VECTOR  000010R             109#
WASADR  000104R             143#
WDFR    000116R             150#
WDTO    000114R             149#
WEIRD   000604R             252#
XFLAG   000005R             107#
.     = 001320R             362#

. ABS.  000000      000
        001320      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XFPAG0,XFPAG0/SOL/CRF:SYM=DDXCOM,XFPAG0
 RUN-TIME: 1 1 .2 SECONDS
 RUN-TIME RATIO: 19/3=6.3
```



```
 CORE USED:  7K  (13 PAGES)
```