DECUS

PROGRAM LIBRARY

DECUS NO.	84
TITLE	M.I.T. Floating Point Arithmetic Package
AUTHOR	Bill Gasper and Tom Eggers
COMPANY	Massachusetts Institute of Technology
DATE	
FORMAT	

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

M.I.T. FLOATING POINT ARITHMETIC PACKAGE

Program Library Write-up DECUS No. 84

The Floating Package is a group of arithmetic subroutines in which numbers are represented in the form f x 2^e. f is a one's complement 18-bit fraction with the binary point between bits 0 and 1. e is a one's complement 18-bit integer exponent of 2. The largest magnitude numbers that can be represented are $\simeq 10^{39,000}$.

A number is normalized when $\frac{1}{2} \leq |f| < 1$. All the floating routines, except the two floating unnormalized adds, return a normalized answer. The fraction appears in the ac, the exponent in the io. Description of routines:

Floating Add - jda fad

One argument should appear in the ac-io. The other argument should have the addresses, direct or indirect, of the fraction and exponent in the two registers following the jda fad.

```
lca f1              /load first argument
lio e1
jda fad             /call floating add
  f2                /address of second fraction
  e2                /address of exponent for second
                       fraction
dac                 /control returns to here with
                    /normalized answer in ac-io
```

Floating Multiply - jda fmp

```
lac f1              /Load multiplicand.
lio e1
jda fmp             /Call floating multiply
  f2                /address of fraction of multiplier
  e2                /address of exponent of multiplier
dac                 /Control returns to here with normal-
                    /ized answer in ac-io.
```

Floating Divide - jda fdv

```
lac f1              /Load first argument.
lio e1
jda fdv             /Call divide.
  f2                /address for divisor, hlt will occur
  e2                    /if f2=0.
dac                 /Control returns here with normalized
                    /answer in ac-io.
```

Floating Square Root - jda fsq

Execution time ≃385 μsec.

```
          lac f            /Load argument; argument must be
                               normalized
          lio e
          jda fsq          /Call square root; hlt will occur
                           /unless f ≥ 0.
          dac              /Control returns here with normalized
                           /answer in ac-io.
```

Floating Log, base 2 - jda log

```
          lac f            /Load argument.
          lio e
          jda log          /Call log; hlt will occur unless f > 0.
          dac              /Control returns here with normalized
                           /answer in ac-io.
```

Floating Reciprocal - jda rcp

```
          lac f            /Load argument.
          lio e
          jda rcp          /Call reciprocal; hlt will occur in
                           /fdh if f=0.
          dac              /Control returns here with normalized
                           /answer in ac-io.
```

Floating Input - jda fip

```
          jda fip          /Call input; ac-io don't matter.
          jsp              /This instruction is repeatedly
                           /executed (xct) in order to get the
                           /input characters.  The jsp (or jda)
                           /could call a typewriter or reader
                           /listen loop subroutine which should
                           /return the input characters in the
                           /low bits of the io.
          dac              /Control returns here with the answer
                           /in the ac-io after the first illegal
                           /character.
```

Legal characters for fip

x resets routine and starts forming a new number. Spaces and code deleted characters are ignored. Legal characters are: ., e, 0-9, -, x, and space. The illegal character that terminated the number is in register fip.

Input examples:

```
    6.9el
    690 e-1234
    -6.9 e 17
```

Floating Output - jda fop

```
            lac f               /Load argument.
            lio e
            jda fop             /Call output.
            tyo                 /an executed instruction (xct) for
                                /output
            dac                 /Control returns here with normalized
                                /floating point input quantity.
```

The routine generates parity for each character, so the executed output instruction could be a ppa or a call to an output subroutine.

The output format is .71000 e2, 5 significant figures.

Floating Unnormalized Add - jda fua

```
            lac fl              /Load first argument.
            lio el
            jda fua             /Call unnormalized add.
              f2                /addresses of second argument
              e2
```

The subroutine returns with a 35 bit number in the ac-io with binary point after the bit number equal to the larger exponent of the two arguments. If the addition produces an overflow, the larger exponent is incremented by 1. In any case, the larger exponent, perhaps incremented, appears in fac+1. Examples for subroutine:

```
            lac  (200000        /½
            lio  (0
            jda  fua
              (0                /zero with exponent to cause the number
              (17.              /to be fixed.
```

At return ac,io equals 0,400000.

```
            lac (0              /0
            lio (16.
            jda fua
              (200000           /½
              (0
```

At return ac,io equals 1,0.

Floating Unnormalized Add and Round - jda fur

```
lac fl              /Load argument
lio el
jda fur             /Call unnormalized add and round
  f2                /addresses of second argument
  e2
dac                 /Control returns here with fraction
                    /in ac and exponent in io.
```

This routine is the same as fad except that the answer is not normalized. The larger exponent returns in the io, unless overflow occured. Then the larger exponent +1 returns in the io. Example:

```
lac (300000         /3
lio (2
jda fur             /Call subroutine.
  (0                /zero with exponent to
  (17.              /cause the answer to be fixed.
```

At return, ac,io equals 3,17.

Floating Exponentiation - jda f2x

This subroutine calculates 2^x. Execution time ≃ 1.3 m sec.

```
lac f               /Load argument.
lio e
jda f2x             /Call subroutine.
dac                 /Control returns to here with normalized
                    /answer in ac-io.
```

```
/floating pack with trig func, arctan, and nat log

fac,     0          0

fmp,     0          /floating multiply
         dap fm1
         dap fm2
         idx fm2
         dap frx
         idx frx
         swp
fm2,     add i .
         dac fac 1
fm1,     lac i .
         mul fmp
         dac fac
         jmp fnm
```

```
fnm,        cla            /normalize.  For internal use only
            dap fnr
            add fac
            sza i
             jmp fze
flp,        sub (200000
            sma
             jmp fnr
            idx fnr
            lac fac
            scl 1
fdf,        dac fac
             jmp flp
fze,        law 17.
            dap fnr
            lac fac
            scl 9s
            scl 8s
            add (0
            sza
             jmp fdf
fnr,        law i .
            add fac 1
            dac fac 1

frn,        lac fac        /round.          "
            sar 9s
            sar 8s
            clo
            scl 1
            add fac
            lio fac 1
            szo i
frx,        jmp .
            rar 1
            swp
            idx fac 1
            swp
            jmp frx
```

```
fdv,      0            /floating divide
          dap fd1
          dap fd2
          idx fd2
          dap frx
          idx frx
          swp
fd2,      sub i .
          dac fac 1
          lac fdv
          cli
          spa
          lio (-0
fd1,      div i .
          jmp fdo
          dac fac
          cla
          spi
          cma
          swp
          div i fd1
          hlt          /impossible
          scr 9s
          scr 8s       /or mul (1
          jmp fnm
fdo,      add (0
          sza i
          hlt          /zero divisor
          scr 1
          dac fdv
          idx fac 1
          lac fdv
          jmp fd1
```

```
fad,      0          /floating add
          dap fa1
          dap fa2
          idx fa2
          dap frx
          idx frx
          law fnm
          dap fux
          cla
          add i fa1
          sza
          jmp fnz
          lac fad
fzn,      dac fac
          dio fac 1
          cli
          spa
          lio (-0
          jmp fnm
fnz,      cla
          add fad
          sza
          jmp fah
          lac i fa1
          lio i fa2
          jmp fzn
fah,      dac fua
          jmp fun
```

```
fua,    0           /unnormalized floating add (fix)
        dap fa1
        dap fa2
        idx fa2
        dap fux
        idx fux
fun,    swp
        dac fac 1
fa2,    sub i .
        sma
        jmp fsr
        dac fde
        lac i fa2
        dac fac 1
        lac fua
        dac fa4
fa1,    lac i .
        dac fua
fba,    cli
        spa
        lio (-0
        dio fa4 1
        scr 1
        and (377777
        dac fmp
        swp
        sar 1
        and (377777
        dac fad
        lac fde
        scr 3s
        add (5
        spq
        jmp fou
        add .+1
        dap fsh
        clc
        scl 3s
        add (fsp 7
        dap fxq
        lac fa4
        cli
        spa
        lio (-0
fsh,    jmp .
        repeat 4, scr 8s
fxq,    xct .
        and (377777
        swp
        sar 1
        and (377777
        add fad
        dac fa3 1
        spa
```

```
          idx fmp
          swp
          cma
          sub fmp
          cma
          dac fa3
          sma
          jmp ful
          law i 1
          sub fa3 1
          cma
          dac fa3 1
          sas (400000
          jmp ful
          law i 1
          sub fa3
          cma
          dac fa3
ful,      lac fa3
          lio fa3 1
          ril 1
          rcl 1
          scr 2s
          scl 2s
          dac fac
          xor fa4
          dac fa3
          lac fac
          xor fua
          and fa3
          sma
          jmp fok
          idx fac 1
          lac fac
          scr 1
          xor (400000
          dac fac
fok,      lac fac
          dio fa4 1
fux,      jmp .

fsr,      cma
          dac fde
          lac i fa1
          dac fa4
          lac fua
          jmp fba
fou,      lac fua
          lio fa4 1
          jmp fok-1
fa3,      0          0         fa4,       0          0
fs8=0
fsp,      repeat 8., scr 8s-fs8          fs8=fs8 fs8 1
fde,      0
```

```
fur,      0            /unnormalized rounded add
          dap fu1
          dap fu2
          idx fu2
          dap frx
          idx frx
          lac fur
          jda fua
fu1,      i .
fu2,      i .
          jmp frn

/square root    jda fsq   Inputs must be normalized (or 0)
fsq,      0
          dap zlv
          law i 3      dac zlv 1
          spi
          cma
          rcr 1
          dio zlv 2
          xor zlv 2
          swp
          spi sma-skp
          idx zlv 2
          lac fsq
          spa
          hlt
          sza i
          jmp zlv-1
          spi
          sar 1
          dac fsq
          sar 2s
          add zlv 3
          jmp . 11
          lac fsq
          cli
          scr 2s
          dis zlv 4
          hlt
          add zlv 4
          cli
          rcr 1
          dac zlv 4
          isp zlv 1
          jmp .-12
          lac zlv 4
          scl 1
          sza i
          sub (400000
          lio zlv 2
zlv,      jmp
          0            0
          66314        0
```

```
/log, base 2.  requires normalized arg

log,       0
           dap lgo
           dio lgo 6
           lac log
           spq
           hlt
           sar 1
           add lgo 1
           dac lg
           lac log
           sub lgo 2
           cli
           spa
           lio (-0
           div lg
           hlt          /not norm.
           dac lg       mul lg
           mul lgo 3
           sar 4s
           add lgo 4
           mul lg
           sub lgo 5
           lio (1
           jda fad
           lgo 6        (17.
lgo,       jmp .
           132405       265012    373621
           270517       100002    0
lg,        0
```

```
/floating input: jda fip, input inst
fip,      0
          dap owt     dap wat  idx owt
ini,      dzm hol
          dzm hol 1
          dzm z11
          dzm dig
          lio wat 3  /spa
          dio cns
          dio cnn
          dio 6fg
wat,      xct .
          dio fip
          rcr 7s
          spa
          jmp wat
          sar 2s
          sar 9s
          lio fns 2  /sma
          sad (charac rx
          jmp ini
          sza i
          jmp wat
          sas (charac r-
          jmp 5fg
          dio cns
          jmp wat
5fg,      sas (charac r.
          jmp . 3
          dio cnn
          jmp wat
          sas (charac re
          jmp cnm
          dio 6fg
          lio wat 3
          dio cns
          jmp wat
cnm,      sad har 1
          cla
          sub wah
          sma
          jmp fns
          add wah
cns,      spa .
          cma
          dac z12
          cla
6fg,      spa .
          jmp cxp
cnn,      spa .
          idx dig
          lac hol
          lio hol 1
          jda fmp
          wah
          har 5
          jda fad
          z12
          har 5
          dac hol
          dio hol 1
          jmp wat
```

```
cxp,        lac zl1
            sal 2s
            add zl1
            sal 1s
            add zl2
            dac zl1
            jmp wat
fns,        lac zl1
            sub dig
            sma
            lio . 1
            spa
            cma
            dac zl1
            dio cpr
cpr,        spa .-.
            jsp inv
            law har
            dap tnp
            law har 1
            dap tnp 1
rpt,        lac zl1
            sza i
            jmp ard 4
            scr 1
            dac zl1
            spi i
            jmp ard
            lac hol
            lio hol 1
            jda fmp
            tnp i
            tnp 1 i
            dac hol
            dio hol 1
ard,        idx tnp 1
            dap tnp
            idx tnp 1
            jmp rpt
            xct cpr
            jsp inv
            lac hol
            lio hol 1
owt,        jmp .
inv,        dap qzl
            lac hol
            sza i
            jmp owt-1
            lio hol 1
            jda rcp
            dac hol
            dio hol 1
qzl,        jmp .-.
wah,        10.
zl1,        0            zl2,      0
```

```
/floating output

fop,      0
          dap urp
          dap xit
          idx xit
          law har 30.
          dap tnp
          law har 31.
          dap tnp 1
          cla
          dap ubm
          lac fop
          dio 151
          lio 2hn
          sma
          jmp urp
          cma
          dac fop
          lio (charac r-
urp,      xct .-.
          lac fop
          lio 151
          jda fmp
          (5
          har 5
          dzm loh
          sza i spi-skp
          jmp mzd
          jda rcp
          dac fop
          rcr 9s
          rcr 9s
          sub (1
          dac 151
          law spa-skp sma-skp
          dap ubm
mzd,      dzm dig
          lac fop
          lio 151
          dac lst
          dio lst 1
          jda fdv
tnp,      har 30.
          har 31.
          dac fop
          dio 151
          jda fmp
          (5
          har 5      spi
          jmp tuf
          idx dig
          jmp mzd 1
tuf,      lac lst
          lio lst 1
          dac fop
          dio 151
          lac loh
2hn,      sal 200
          add dig
          dac loh
          law i 2
          add tnp
          dap tnp
          dap tnp 1
          idx tnp 1
          sas tob
          jmp mzd
ubm,      skp .-.
          jmp drp
          lac c1
          lio c1 1
          jda fdv
          fop
          151
          dac fop
          dio 151
drp,      lio (charac r.
          xct urp
          lac fop
          lio 151
          jda fmp
          tn5
          har
          jda fur
          (0
          har 5
          jda dpt
          xct urp
          lio 2hn
          xct urp
          lio (265
          xct urp
          lac loh
          cma
          xct ubm
          cma
          jda dpt
          xct urp
xit,      jmp .-.
har,      20.        20       100.       17.       10000.
17.
97656.    27.        72759.   54.        80778.    107.
99565.    213.       75632.   426.       87283.    851.
116246.   1701.      103097.  3402.      81093.    6804.
100343.   13607.     76818.   27214.     90042.    54427.
123712.   108853.
tob,      har-1      loh,     0
hol,      0          0
lst,      0          0
c1,       314632     -3
dig,      0          151,     0          tn5,      12500.
```

```
/decimal integer print of ac.  jda dpt followed by output instr.

dpt,      0
          dap dpo
          dap dpx
          dzm ddv
          idx dpx
          lio (charac r-
dlp,      lac dpt
          spa
          xct dpo
          spa
          cma
          dac dpt
dl1,      dac dpr
          mul (1
          div (10.
dpr,      0
          sas ddv
          jmp dl1
          sni                      /note sni
          lio (charac r0
dpo,      xct .
          lac dpr
          dac ddv
          sas dpt
          jmp dlp
dpx,      jmp .
ddv,      0

/parity for low 6 io bits, saves ac
pty,      0
          dap ytp
          law i 770
          rcr 6s
          lio (252002
          rcr 9s
          dap . 1
          rir
          spi
          and pty 2
          rcl 6s
          rcl 9s
          lac pty
ytp,      jmp .

/reciprocal routine
rcp,      0
          dap pcr
          dio pcr 1
          cli
          law 1200
          rcl 9s
          jda fdv
          rcp
          pcr 1
pcr,      jmp .
pcr 1,    0
```

```
/antilog, base 2
f2x,      0
          dap fxx
          lac f2x
          dio f2x
          lio (nop
          spa
          lio (jda rcp
          dio fck
          spa
          cma
          lio f2x
          jda fua
          (0
          (17.
          dac fmp
          idx fmp
          law 17.
          sas fac 1
          hlt         /power too big
          lac (200000
          dac fac
          law ftb
          dap fmt
fci,      spi i
          jmp fz0
          dio z11
fmt,      lac .
          mul fac
          scl 1
          dac fac
          spi
          idx fac
          lio z11
fz0,      idx fmt
          sad (lac ftb 16.
          jmp fdu
          rcl 1
          sni i
          jmp fci
fdu,      lac fac
          lio fmp
fck,      .-. .
fxx,      jmp .
ftb,      265017    230160    213454    205526    202633
201312    200544    200262    200131    200054    200026
200013    200006    200003    200001    200001
```

```
/x to the y power
pow,      0
          dap po1
          dap po2
          idx po2
          dap pox
          idx pox
          lac pow
          jda log
          jda fmp
po1,      i .
po2,      i .
          jda f2x
pox,      jmp .
```

```
/floating normal angle,  for internal use only

fna,        0
            dap rt1
            clf 6
            dio fed
            lac fna
            spa
            jmp inc
            lac fna
            lio fed
            jda fad
            (-311040
            (3
            sma
            dac fna
            sma
            dio fed
            sma
            jmp .-12
            lac fna
            lio fed
            jda fad
            (-226630
            (3
            sma
            jmp 4q
            jda fad
            (311040
            (1
            sma
            jmp 3q
            jda fad
            (311040
            (1
            sma
            jmp 2q
            lac fna
            lio fed
rt1,        jmp .
inc,        lac fna
            jda fad
            (311040
            (3
            dac fna
            dio fed
            jmp fna 4
4q,         dac fna
            dio fed
            stf 6
            lac fna

            lio fed
2q,         cma
            jda fad
            (311040
            (1
            dac fna
            dio fed
            jmp rt1-2
3q,         dac fna

            dio fed
```

```
          stf 6
          jmp rt1-2

/floating sin-cos

fcs,      0
          dap rt2
          lac fcs
          jda fad
          (311040
          (1
          jda fna
          dac fsn
          dio fex
          jmp .+5
fsn,      0
          dap rt2
          lac fsn
          jda fna
          dac fsn
          dio f̅ex
          jda fmp
          fsn
          fex
          dac f̅xs
          dio f̅es
          jda fmp
          fsn
          fex
          jda fmp
          (252533
          (-2
          dac f̅dd
          dio f̅ed
          cma
          jda fad
          fsn
          fex
          dac fcs
          dio f̅ee
          lac fdd
          lio fed
          jda fmp
          fxs
          fes
          jda fmp
          (314637
          (-4
          dac fdd
          dio fed
          jda fad
          fcs
          fee
          dac fcs
          dio fee
          lac fdd
          lio fed
          jda fmp
          (303034
          (-5
          cma
          jda fad
          fcs
          fee
          sza i
          lio fex


          sza i
          lac fsn
          szf 6
          cma
rt2,      jmp .

/floating secant - cosecant

fsc,      0
          dap rt3
          lac fsc
          jda fad
          (311040
          (1
          dac fco
          jmp .+3
fco,      0
          dap rt3
          lac fco
          jda fsn
          sza i
          jmp .+3
          jda rcp
rt3,      jmp .
          lio (377777          lai
          jmp rt3
```

```
/floating tan cot

ftn,      0
          dap rt4
          dio fte
          lac ftn
          jda fcs
          dac fsc
          sza i
          jmp rt4 1
          dio fco
          lac ftn
          lio fte
          jda fsn
          jda fdv
          fsc
          fco
rt4,      jmp .
          lio (377777
          lai
          jmp rt4

fct,      0
          dap rt5
          dio fte
          lac fct
          jda fsn
          dac fsc
          sza i
          jmp rt5 1
          dio fco
          lac fct
          lio fte
          jda fcs
          jda fdv
          fsc
          fco
rt5,      jmp .
          lio (377777
          lai
          jmp rt5
```

```
fln,    0          /floating natural log
        dap rt6
        dio eln
        lac fln
        spq
        hlt
        lac (200000
        dac lnc
        lac (1
        dac lce
        lac fln
        jda fad
        lnc
        lce
        dac fln
        dio eln
        jda fad
        (-200000
        (2
        jda fdv
        fln
        eln
        dac fln
        dio eln
        dac lna
        dio lne
        jda fmp
        fln
        eln
        dac fls
        dio els
lrp,    lac lnc
        lio lce
        jda fad
        (200000
        (2
        dac lnc
        dio lce
        lac fln
        lio eln
        jda fmp
        fls
        els
        jda fdv
        lnc
        lce
        jda fad
        lna
        lne
        sad lna
        jmp rt6 1
        dac lna
        dio lne
rt6,    jmp .
        jda fmp     (200000
        (2
        jmp rt6
```

```
atn,        0
            dap atx
            lac atn
            dac att
            dio att 1
            sma
            cma
            jda fad
            (200000
            (1
            sma
            jmp at5
            lac att
            lio att 1
            jda rcp
            dac att
            dio att 1
            law at4
            skp i
at5,        law at 1
            dap at
            lac att
            lio att 1
            jda fmp
            att
            att 1
            dac btt
            dio btt 1
            law gt
            dap at1
            law gt1
            dap at1 1
            lac att
            lio att 1
            jda fmp
            ht
            ht 1
            dac ctt
            dio ctt 1
```

```
at2,      lac att
          lio att 1
          jda fmp
          btt
          btt 1
          dac att
          dio att 1
          jda fmp
at1,      .
          .
          jda fad
          ctt
          ctt 1
          dac ctt
          dio ctt 1
          idx at1
          idx at1 1
          sas (gt1 4
          jmp at2
at,       jmp .
          lac ctt
atx,      jmp .

at4,      lio ctt
          lac (311040
          spi 1
          cma
          lio (1
          jda fad
          ctt
          ctt 1
          cma                  /   pi/2 - arccot(x)
          jmp atx

att,      0          0
btt,      0          0
ctt,      0          0
ht,       377772     0
gt,       -251072    270355   -256271    252535
gt1,      -1         -2       -3         -5



variables
constants
start
```