Better math for AmigaE

Michal Bartczak

COLLABORATORS

	TITLE : Better math for AmigaE		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Michal Bartczak	December 6, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Better math for AmigaE

1.1 main

```
*************************************************************************
*                   Better floating point in "E"                        *
*    Now You can create programs for FPU and for standard libraries     *
*                           _really_ easily.                            *
*                                                                       *
*                       FPU_in_E included as bonus                      *
*                                                                       *
*        This is next version of (my) "FPU_in_E", read HISTORY.        *
*                                                                       *
*                     Written by Michal Bartczak                       *
*                                                                       *
* This program is __MAILWARE__ , so if You like it, then send me EMAIL *
*************************************************************************

Uff. Maybe I should write little doc here ? It can be silly, because my
english is poor... VERY POOR...
```

1.2 requirements

```
1. REQUIREMENTS

  Minimum system requirements for BetterMath:
    "E" compiler :-) (recommended version with preprocessor)
```

1.3 introduction

```
2. INTRODUCTION

  BetterMath  consists of 3 modules.  "math_881_s.m" gives You functions
to  use  ONLY  with  FPU,  and module "math_ieee_s.m" gives You the same
functions, but for mathieee libraries.  Third module - "math_881_test.m"
returns TRUE if FPU is present in current system, FALSE otherwise.
```

1.4 installation

```
3. INSTALLATION

  Simply   copy   all   ".m"   files   somewhere.   The  best  place  is
"emodules:bettermath/", but it is Your choice.
```

1.5 usage

```
4. USAGE

  Ok.   You have BetterMath modules installed.  I think, You know how to
use  them,  so  I  don't  have  to  say  stuffs  like  writing  "MODULE
'bettermath/math_881_s'"  or  other here.  There will be only SIMPLY SET
OF ALL INSTRUCTIONS:

"float"  -  is an floating point number ("E" can handle single precision
floating  point  numbers.   Just write 3.14 or 1.0, or 0.001 - there are
used  by  compiler as floating point numbers.  See chapter of "E" manual
called  "floating  point  support".  BetterMath for AmigaE is compatible
with  "E"  floating  numbers,  so You can use i.e "RealF()", "RealVal()"
functions with BetterMath floating numbers)

Functions for modules: "math_881_s.m" and "math_ieee_s.m"

         finit()               = TRUE if initialising (opening math
                                 libraries) was succesfull. FALSE elsewhere
         fend()                = TRUE if closing libraries was
                                 succesfull. FALSE elsewhere


         fabs(float)           = Absolute Value of "float"
         facos(float)          = Arc Cosine of "float"
         fadd(float1,float2)   = "float1" + "float2"
         fasin(float)          = Arc Sine of "float"
         fatan(float)          = Arc Tangent of "float"
         fcmp(float1,float2)   = Results are integer! -1 if float1<float2
                                                       0 if float1=float2
                                                       1 if float1>float2

         fcos(float)           = Cosine of "float"
         fcosh(float)          = Hyperbolic Cosine of "float"
         fdiv(float1,float2)   = "float1" / "float2"
         fint(float)            = Integer part of "float" - result is FLOAT
         flog10(float)          = Log[10] of "float"
         flogn(float)           = Log[e] of "float"
         fmul(float1,float2)    = "float1" * "float2"
         fneg(float)            = -"float"
         fsin(float)            = Sine of "float"
         fsinh(float)           = Hyperbolic Sine of "float"
         fsqrt(float)           = Square Root of "float"
         fsub(float1,float2)    = "float1" - "float2"
         ftan(float)            = Tangent of "float"
         ftanh(float)           = Hyperbolic Tangent of "float"
         ftoreal(integer)       = Converts integer to "float"
         ftointeger(float)      = Converts float to integer

And in "math_881_test.m" module:
         ftestiffpu()           = result is TRUE  - FPU is present
                                            FALSE - There is NO FPU

Now,  You  should  use  "math_881_s.m"  module  for  FPU version of Your
programs,  and  "math_ieee_s.m"  module for no-fpu version.  FPU version
uses  direct  FPU  chip calls, so it will crash systems without FPU.  In
beginning  of Your FPU version of program You should test if FPU chip is
present - use "math_881_test.m" module function ftestiffpu().

Everything clear ? See examples.
```

1.6 bugs

```
5.   BUGS  -  HEY  !!   I  DON'T  SEE  ANY  BUGS!   (I'm  A-U-T-H-O-R,  not
tester....:->)

  IF YOU FOUND BUG, THEN SEND DESCRIPTION FOR ME ! - I WILL SPRAY IT.
```

1.7 examples

```
6. EXAMPLES

Example ONE (and alone :-):

-------------------------------------------------------------------------------- ↩

-> mini fractal example, also little timing...

OPT PREPROCESS

/* Add comment for this line if You want NOFPU version of this example */
#define FPU

#ifdef FPU
MODULE 'bettermath/math_881_s','bettermath/math_881_test'
#endif
```

```
#ifndef FPU
MODULE 'bettermath/math_ieee_s'
#endif

MODULE 'dos/dos'

CONST TICKS_PER_MINUTE=TICKS_PER_SECOND*60
CONST CALCW=200,HEIGHT=100, DEPTH=25

PROC main()
  DEF w,xmax,ymax,x,y,xr,width=3.5,height=2.8,left,top,ds1:datestamp,ds2:datestamp ↩
      ,
  tct
#ifdef FPU
  IF ftestiffpu()
#endif

#ifdef FPU
    WriteF('This is FPU version of this program.\n')
#endif
#ifndef FPU
    WriteF('This is NOFPU version of this program.\n')
#endif

    finit()
    IF w:=OpenW(0,11,CALCW+40,HEIGHT+30,$200,$E,'MiniFrac!',NIL,1,NIL)
      DateStamp(ds1)
      top:=fsub(0.0,3.2)
      left:=fsub(0.0,2.0)
      xmax:=ftoreal(CALCW)
      ymax:=ftoreal(HEIGHT-1)

      FOR x:=0 TO CALCW-1
      xr:=fadd(fmul(fdiv(ftoreal(x),xmax),width),left)
        FOR y:=0 TO HEIGHT-1 DO Plot(x+20,y+20,calc(xr,fadd(fmul(fdiv(ftoreal(y), ↩
            ymax),height),top)))
      ENDFOR
      DateStamp(ds2)
      tct:=((ds2.minute-ds1.minute)*TICKS_PER_MINUTE)+ds2.tick-ds1.tick
      WaitIMessage(w)
      CloseW(w)
      WriteF('Ticks:\d\n',tct)
    ELSE
      WriteF('Can''t open window !\n')
    ENDIF
    fend()
#ifdef FPU
  ELSE
    WriteF('Sorry, FPU is needed.\n')
  ENDIF
#endif

ENDPROC

PROC calc(x,y)
  DEF xtemp,it=0,xc,yc
  xc:=x; yc:=y
  WHILE (it++<DEPTH) AND (fcmp(fmul(fadd(fmul(x,x),y),y),16.0)=-1)
    xtemp:=x

    x:=fadd(fsub(fmul(x,x),fmul(y,y)),xc)
    y:=fadd(fmul(fadd(xtemp,xtemp),y),yc)

  ENDWHILE
ENDPROC it
----------------------------------------------------------------------------------- ↩
```

```
Ok. That’s all folks!
```

1.8 other

```
7. OTHER

 ToDo list:
   Double precision support
   68040/68060 support (with no &%%!&* software emulations)


 IF YOU HAVE ANY COMMENTS, OR YOU HAVE FOUND SOME BUGS, THEN SEND ME A
                  MAIL! (I HAVE A BUGSPRAY SOMEWHERE...)

 SO: COMMENTS, GRETINGS, BUGS DESCRIPTIONS AND MONEY ( $-) !) ARE WELCOME.
     FUCKINGS, FLAME AND OTHER ARE NOT WERY WELCOME...

 Also see AUTHOR



  THIS  DOCUMENTATION  AND  SOFTWARE  ARE  PROVIDED  "AS  IS"  WITHOUT  ANY
WARRANTY, EITHER EXPRESSED OR IMPLIED.  USE THIS SOFTWARE ON YOUR OWN RISK,
AND  DONT  BLAME  ON ME, IF "BetterMath" EAT YOUR FAWOURITE PIZZA FROM YOUR
FRIDGE...

Amiga "E" is very good programing language for Amiga, written by Wouter van
Oortmerssen.

Amiga is the best computer ever.  I should write here:  trademark of "Amiga
owner"  is  known  for  me,  and  is registered for "Amiga owner" only, but
answer  for  question  "who  owns  rights  for Amiga" changes so quickly...
```

1.9 history

```
8. HISTORY

  07/31/1996 - FPU in E v1.0 - First try to make FPU usage in AmigaE

  08/25/1996 - Name changed to BetterMath. First version of general
```

```
               math modules for AmigaE (direct FPU usage, and libraries
               calls. EASY to use). Guide for FPUinE changed a bit to
               match BetterMath... (I'm LAZY ! :) )
               Check FPU_in_E (included). And compare guides :>.
```

1.10 author

```
9. AUTHOR

                               MICHAL BARTCZAK
                            ul. Lukasinskiego 2/18
                                05-820 PIASTOW
                                    POLAND

  EMAIL: s0402@goblin.pjwstk.waw.pl

  PHONE: 723-52-44 in Warsaw (this is small village in Poland... :->)
     OR: 617-78-12 in Warsaw (this phone is actual to april 1997)


(it's  almost  impossible  to catch me in home...  You know...  GIRLS, Work
(I'm   working   for   Polish   Public   Television   [TVP   S.A]),  school
(Polish-Japaneese Higher Computer Techniques School - maybe it is the right
name  for  it  in  english...), my cats (4), my dogs (2), my sister (1) and
other  funny things, that are more important from sitting in front of phone
in my house...)


Some info about me:
   I'm 19 and half years old, I am "BIG AMIGA" user. And I don't know
                             english...

Why "BIG AMIGA"? See on my config... :-)

MY CONFIG:
one A1200
one 1084S
one Thompsonic Super Stereo Tower (:->)
one CD-ROM (model no. CDA 268-031SE - dual slow)
one (not really mine) STAR LC10 Colour Super Quality Printer

one 16MB 60ns SIMM     -\
one MC68060              >- All of them putted on one Blizzard A1260
one 50000000Hz clock   -/

and  some  other  stuff,  like:   realtime  clock  - also on A1260 + SUPERB
BATTERY  (without warranty) for it, one Super Safe Video Backup System, six
Video  Archive  Tapes,  some CD-ROM (Aminet, MacWorld Cover CD, other Amiga
Stuff,  some  music),  some  books  ("AMIGA  ROM  Kernel Reference Maunal",
Motorola "PROGRAMMER'S REFERENCE MANUAL FOR MC68XXX FAMILY (INCLUDING CPU32
INSTRUCTIONS)",  ZX  SPECTRUM beginner manual, not complete polish-english,
and  english-polish  translation  book  (with  no gramar rules, and without
about 50 pages, that was consumed by my beautifull dog).

Special addition for my A1200 :->:
```

```
I  have  created beautifull hole in trap door of my A1200 (just in shape of
MC68060).   Now in this hole is little cooler (from pENTIUM).  I had to add
bigger  legs  for  my Amiga, but it WORKS.  Now I can leave my "girl" for a
week (or two) until she calculate "little" rendered animation.  If You have
fast  CPU inside Your A1200 it’s good time to do hole like mine inside Your
computer.   Your  system  will be more stable, and (probably) Your CPU will
live longer.

                    HEY ! Somebody read that doc ?!?

(ps: SORRY FOR MY "funny" ENGLISH! (again!))
```