```
**********************************************************
    3690 Tue Nov  4 18:59:18 2014
new/usr/src/lib/libm/amd64/src/__swapFLAGS.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "__swapFLAGS.s"

#include "libm.h"
#include "libm_synonyms.h"

/*
 * swap exception masks
 *
 * Put the complement of bits 5-0 of the argument into FPCW bits 5-0
 * and MXCSR bits 12-7, return the complement of the previous FPCW
 * bits 5-0.
 */
        ENTRY(__swapTE)         / di <-- NOT(desired xcptn_masks)
        subq    $8,%rsp
        fstcw   (%rsp)          / push current_cw on '86 stack
        movq    (%rsp),%rcx     / cx <-- current_cw
        movw    %cx,%ax         / ax <-- current_cw
        orw     $0x3f,%cx       / cx <-- current_cw, but masking all xcptns
        andw    $0x3f,%di       / make sure bits > B5 are all zero
        xorw    %di,%cx         / cx <-- present_cw, with new xcptn_masks
        movw    %cx,(%rsp)
        fldcw   (%rsp)          / load new cw
        stmxcsr (%rsp)
        movq    (%rsp),%rcx
        orw     $0x1f80,%cx     / cx <-- current mxcsr, but masking all xcptns
        shlw    $7,%di
        xorw    %di,%cx         / cx <-- present mxcsr, with new xcptn_masks
        movq    %rcx,(%rsp)
        ldmxcsr (%rsp)
        andq    $0x3f,%rax      / al[5..0] <-- former xcptn_masks
        xorq    $0x3f,%rax      / al[5..0] <-- NOT(former xcptn_masks)
        addq    $8,%rsp
        ret
```



```
        .align  16
        SET_SIZE(__swapTE)
______unchanged_portion_omitted_
```

```
**********************************************************
    1692 Tue Nov  4 18:59:19 2014
new/usr/src/lib/libm/amd64/src/acosl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "acosl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(acosl,function)
#include "libm_synonyms.h"

#undef fabs

        ENTRY(acosl)
        fldt    8(%rsp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomip %st(1),%st
        ja      9f
        fadd    %st(1),%st              / 1+x,x
        fldz
        fucomip %st(1),%st
        jp      .L1
        jne     .L1
        / x is -1
        fstp    %st(0)                  / -1
        fstp    %st(0)                  / empty NPX stack
        fldpi
        ret
.L1:
        fxch    %st(1)                  / x,1+x
        fld1                            / 1,x,1+x
        fsubp   %st,%st(1)              / 1-x,1+x
        fdivp   %st,%st(1)              / (1-x)/(1+x)
        fsqrt
        fld1                            / 1,sqrt((1-x)/(1+x))
        fpatan
```



```
        fadd    %st(0),%st
        ret
9:
        / |x| > 1
        fstp    %st(0)                  / x
        fsub    %st,%st(0)              / +/-0 or NaN+invalid
        fdiv    %st,%st(0)              / NaN+invalid or NaN
        ret
        .align  16
        SET_SIZE(acosl)
```

*unchanged_portion_omitted_*

****
1548 Tue Nov 4 18:59:19 2014
new/usr/src/lib/libm/amd64/src/asinl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "asinl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(asinl,function)
#include "libm_synonyms.h"

#undef fabs

        ENTRY(asinl)
        fldt    8(%rsp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomip %st(1),%st
        ja      9f
        fadd    %st(1),%st              / 1+x,x
        fld1                            / 1,1+x,x
        fsub    %st(2),%st              / 1-x,1+x,x
        fmulp   %st,%st(1)              / (1-x)*(1+x),x
        fsqrt                           / sqrt((1-x)*(1+x)),x
        fpatan                          / atan(x/sqrt((1-x)*(1+x)))
        ret
9:
        / |x| > 1
        fstp    %st(0)                  / x
        fsub    %st,%st(0)              / +/-0 or NaN+invalid
        fdiv    %st,%st(0)              / NaN+invalid or NaN
        ret
        .align  16
        SET_SIZE(asinl)
```

*_____unchanged_portion_omitted_*

****
1203 Tue Nov 4 18:59:20 2014
new/usr/src/lib/libm/amd64/src/atan2l.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "atan2l.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(atan2l,function)
33 #include "libm_synonyms.h"

34         ENTRY(atan2l)
35         fldt    8(%rsp)                 / push y
36         fldt    24(%rsp)                / push x
37         fpatan                          / return atan2(y,x)
38         ret
39         .align  16
40         SET_SIZE(atan2l)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1192 Tue Nov  4 18:59:20 2014
new/usr/src/lib/libm/amd64/src/atanl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "atanl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(atanl,function)
#include "libm_synonyms.h"

        ENTRY(atanl)
        fldt    8(%rsp)                 / push arg
        fld1                            / push 1.0
        fpatan                          / atan(arg/1.0)
        ret
        .align  16
        SET_SIZE(atanl)
______unchanged_portion_omitted_
```

```
**********************************************************
    1261 Tue Nov  4 18:59:21 2014
new/usr/src/lib/libm/amd64/src/copysignl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "copysignl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(copysignl,function)
#include "libm_synonyms.h"

        ENTRY(copysignl)
        movl    16(%rsp),%eax
        movl    32(%rsp),%ecx
        andl    $0x7fff,%eax
        andl    $0x8000,%ecx
        orl     %ecx,%eax
        movl    %eax,16(%rsp)
        fldt    8(%rsp)
        ret
        .align  16
        SET_SIZE(copysignl)
______unchanged_portion_omitted_
```

```
**********************************************************
    3519 Tue Nov  4 18:59:21 2014
new/usr/src/lib/libm/amd64/src/exp10l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp10l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp10l,function)
#include "libm_synonyms.h"

        .data
        .align  16
lt2_hi: .4byte  0xfbd00000, 0x9a209a84, 0x3ffd, 0x0
lt2_lo: .4byte  0x653f4837, 0x8677076a, 0xbfc9, 0x0

        ENTRY(exp10l)
        movl    16(%rsp),%ecx           / cx <--sign&bexp(x)
        andl    $0x7fff,%ecx            / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x3ffd,%ecx            / Is |x| < log10(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / maybe |x| only slightly < log10(2)
.general_case:                          / Here, |x| > log10(2) or x is NaN
        cmpl    $0x7fff,%ecx            / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        cmpl    $0x400e,%ecx            / |x| < 32768 = 2^15?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    8(%rsp)                 / x
        fld1                            / 1, x
        jmp     1f
.finite_non_special:                    / Here, log10(2) < |x| < 2^15
        fldt    8(%rsp)                 / x
        fld     %st(0)                  / x, x
        fldl2t                          / log2(10), x, x
        fmulp                           / z := x*log2(10), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
```



```
        fldt    PIC_L(lt2_hi)           / lt2_hi, [z], x, [z]
        fmulp                           / [z]*lt2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*lt2_hi, [z]
        fldt    PIC_L(lt2_lo)           / lt2_lo, x-[z]*lt2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*lt2_lo, x-[z]*lt2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*log10(2), [z]
        fldl2t                          / log2(10), r, [z]
        fmulp                           / f := r*log2(10), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
1:
        fscale                          / 10^x, [z]
        fstp    %st(1)
        ret

.check_tail:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x9a209a84,%ecx        / Is |x| < log10(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    8(%rsp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0xfbcff798,%edx        / Is |x| slightly > log10(2)?
        ja      .finite_non_special     / branch if |x| slightly > log10(2)
.shortcut:
        / Here, |x| < log10(2), so |z| = |x/log10(2)| < 1
        / whence z is in f2xm1's domain.
        fldt    8(%rsp)                 / x
        fldl2t                          / log2(10), x
        fmulp                           / z := x*log2(10)
        f2xm1                           / 2^z-1
        fld1                            / 1, 2^z-1
        faddp   %st,%st(1)              / 10^x
        ret

.not_finite:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(sgnfcnd(x)) = hi_32(sgnfcnd(INF)
        jne     .NaN_or_pinf            / if not, x is NaN or unsupp.
        movl    8(%rsp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0,%edx                 / lo_32(sgnfcnd(x)) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    16(%rsp),%eax           / ax <-- sign&bexp((x))
        andl    $0x8000,%eax            / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    8(%rsp)
        ret
        .align  16
        SET_SIZE(exp10l)
```

*unchanged_portion_omitted_*

```
**********************************************************
    2950 Tue Nov  4 18:59:21 2014
new/usr/src/lib/libm/amd64/src/exp2l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp2l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp2l,function)
#include "libm_synonyms.h"

        ENTRY(exp2l)
        movl    16(%rsp),%ecx           / cx <--sign&bexp(x)
        andl    $0x7fff,%ecx            / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x3fff,%ecx            / Is |x| <= 1?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be slightly > 1
.general_case:                          / Here, |x| > 1 or x is NaN
        cmpl    $0x7fff,%ecx            / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, 1 < |x| < INF
        fldt    8(%rsp)                 / push arg
        fld     %st(0)                  / duplicate stack top
        frndint                         / [x],x
        fucomi  %st(1),%st              / x integral?
        je      .x_integral             / branch if x integral
        fxch                            / x, [x]
        fsub    %st(1),%st              / x-[x], [x]
        f2xm1                           / 2**(x-[x])-1, [x]
        fld1                            / 1,2**(x-[x])-1, [x]
        faddp   %st,%st(1)              / 2**(x-[x]), [x]
        fscale                          / 2**x = 2**(arg), [x]
        fstp    %st(1)
        ret

.x_integral:
        fstp    %st(0)                  / ,x
        fld1                            / 1 = 2**0, x
```



```
        fscale                          / 2**(0 + x) = 2**x, x
        fstp    %st(1)                  / 2**x
        ret

.check_tail:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / Is |x| <= 1?
        ja      .finite_non_special
        movl    8(%rsp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0x0,%edx               / Is |x| slightly > 1?
        ja      .finite_non_special     / branch if |x| slightly > 1
.shortcut:
        / Here, |x| < 1,
        / whence x is in f2xm1's domain.
        fldt    8(%rsp)                 / push x
        f2xm1                           / 2**x - 1
        fld1                            / 1,2**x - 1
        faddp   %st,%st(1)              / 2**x
        ret

.not_finite:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%rsp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    16(%rsp),%eax           / ax <-- sign&bexp((x))
        andl    $0x8000,%eax            / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    8(%rsp)
        ret
        .align  16
        SET_SIZE(exp2l)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    3708 Tue Nov  4 18:59:22 2014
new/usr/src/lib/libm/amd64/src/expl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expl,function)
#include "libm_synonyms.h"

        .data
        .align  16
ln2_hi: .4byte  0xd1d00000, 0xb17217f7, 0x3ffe, 0x0
ln2_lo: .4byte  0x4c67fc0d, 0x8654361c, 0xbfce, 0x0

        ENTRY(expl)
        movl    16(%rsp),%ecx           / cx <--sign&bexp(x)
        andl    $0x7fff,%ecx            / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x3ffe,%ecx            / Is |x| < 0.5?
        jb      2f                      / If so, see which shortcut to take
        je      .check_tail             / More checking if 0.5 <= |x| < 1
.general_case:                          / Here, |x| >= 1 or x is NaN
        cmpl    $0x7fff,%ecx            / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        cmpl    $0x400e,%ecx            / |x| < 32768 = 2^15?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    8(%rsp)                 / x
        fld1                            / 1, x
        jmp     1f
.finite_non_special:                    / Here, ln(2) < |x| < 2^15
        fldt    8(%rsp)                 / x
        fld     %st(0)                  / x, x
        fldl2e                          / log2(e), x, x
        fmulp                           / z := x*log2(e), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
```



```
        fldt    PIC_L(ln2_hi)           / ln2_hi, [z], x, [z]
        fmulp                           / [z]*ln2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*ln2_hi, [z]
        fldt    PIC_L(ln2_lo)           / ln2_lo, x-[z]*ln2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*ln2_lo, x-[z]*ln2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*ln(2), [z]
        fldl2e                          / log2(e), r, [z]
        fmulp                           / f := r*log2(e), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
1:
        fscale                          / e^x, [z]
        fstp    %st(1)
        ret

2:                                      / Here, |x| < 0.5
        cmpl    $0x3fbe,%ecx            / Is |x| >= 2^-65?
        jae     .shortcut               / If so, take a shortcut
        fldt    8(%rsp)                 / x
        fld1                            / 1, x
        faddp   %st,%st(1)              / 1+x (for inexact & directed rounding)
        ret

.check_tail:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0xb17217f7,%ecx        / Is |x| < ln(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    8(%rsp),%edx            / edx <-- lo_32(x)
        cmpl    $0xd1cf79ab,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x/ln(2)| < 1,
        / whence z is in f2xm1's domain.
        fldt    8(%rsp)                 / x
        fldl2e                          / log2(e), x
        fmulp                           / x*log2(e)
        f2xm1                           / 2^(x*log2(e))-1 = e^x-1
        fld1                            / 1, e^x-1
        faddp   %st,%st(1)              / e^x
        ret

.not_finite:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%rsp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    16(%rsp),%eax           / ax <-- sign&bexp((x))
        andl    $0x8000,%eax            / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    8(%rsp)
        fadd    %st(0),%st              / quiet SNaN
        ret
        .align  16
        SET_SIZE(expl)
_____unchanged_portion_omitted_
```

```
**********************************************************
    3719 Tue Nov  4 18:59:22 2014
new/usr/src/lib/libm/amd64/src/expm1l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expm1l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expm1l,function)
#include "libm_synonyms.h"

        .data
        .align  16
ln2_hi: .4byte  0xd1d00000, 0xb17217f7, 0x3ffe, 0x0
ln2_lo: .4byte  0x4c67fc0d, 0x8654361c, 0xbfce, 0x0

        ENTRY(expm1l)
        movl    16(%rsp),%ecx           / cx <--sign&bexp(x)
        movl    %ecx,%eax               / ax <--sign&bexp(x)
        andl    $0x7fff,%ecx            / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x3ffe,%ecx            / Is |x| < ln(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be only slightly < ln(2)
.general_case:                          / Here, |x| > ln(2) or x is NaN
        cmpl    $0x7fff,%ecx            / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        andl    $0xffff,%eax            / eax <-- sign&bexp(x)
        cmpl    $0xc006,%eax            / x <= -128?
        jae     1f                      / if so, simply return -1
        cmpl    $0x400d,%ecx            / |x| < 16384 = 2^14?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    8(%rsp)                 / x >= 16384; x
        fld1                            / 1, x
        fscale                          / +Inf, x
        fstp    %st(1)                  / +Inf
        ret

.finite_non_special:                    / -128 < x < -ln(2) || ln(2) < x < 2^14
```



```
        fldt    8(%rsp)                 / x
        fld     %st(0)                  / x, x
        fldl2e                          / log2(e), x, x
        fmulp                           / z := x*log2(e), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
        fldt    PIC_L(ln2_hi)           / ln2_hi, [z], x, [z]
        fmulp                           / [z]*ln2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*ln2_hi, [z]
        fldt    PIC_L(ln2_lo)           / ln2_lo, x-[z]*ln2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*ln2_lo, x-[z]*ln2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*ln(2), [z]
        fldl2e                          / log2(e), r, [z]
        fmulp                           / f := r*log2(e), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
        fscale                          / e^x, [z]
        fstp    %st(1)                  / e^x
        fld1                            / 1, e^x
        fsubrp  %st,%st(1)              / e^x-1
        ret

.check_tail:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0xb17217f7,%ecx        / Is |x| < ln(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    8(%rsp),%edx            / edx <-- lo_32(x)
        cmpl    $0xd1cf79ab,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x/ln(2)| < 1,
        / whence z is in f2xm1's domain.
        fldt    8(%rsp)                 / x
        fldl2e                          / log2(e), x
        fmulp                           / z := x*log2(e)
        f2xm1                           / 2^(x*log2(e))-1 = e^x-1
        ret

.not_finite:
        movl    12(%rsp),%ecx           / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%rsp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    16(%rsp),%eax           / ax <-- sign&bexp((x))
        andl    $0x8000,%eax            / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
1:
        fld1                            / Here, x = -inf, so return -1
        fchs
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    8(%rsp)
        ret
        .align  16
        SET_SIZE(expm1l)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1153 Tue Nov  4 18:59:23 2014
new/usr/src/lib/libm/amd64/src/fabsl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "fabsl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fabsl,function)
#include "libm_synonyms.h"

        ENTRY(fabsl)
        fldt    8(%rsp)
#undef  fabs
        fabs
        ret
        .align  16
        SET_SIZE(fabsl)
______unchanged_portion_omitted_
```

****
1869 Tue Nov 4 18:59:23 2014
new/usr/src/lib/libm/amd64/src/floorl.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "floorl.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(ceill,function)
33 LIBM_ANSI_PRAGMA_WEAK(floorl,function)
34 #include "libm_synonyms.h"

35         ENTRY(ceill)
36         subq    $16,%rsp
37         fstcw   (%rsp)
38         fldt    24(%rsp)
39         movw    (%rsp),%cx
40         orw     $0x0c00,%cx
41         xorw    $0x0400,%cx
42         movw    %cx,4(%rsp)
43         fldcw   4(%rsp)                 / set RD = up
44         frndint
45         fstcw   4(%rsp)                 / restore RD
46         movw    4(%rsp),%dx
47         andw    $0xf3ff,%dx
48         movw    (%rsp),%cx
49         andw    $0x0c00,%cx
50         orw     %dx,%cx
51         movw    %cx,(%rsp)
52         fldcw   (%rsp)                  / restore RD
53         addq    $16,%rsp
54         ret
55         .align  16
56         SET_SIZE(ceill)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1754 Tue Nov  4 18:59:24 2014
new/usr/src/lib/libm/amd64/src/fmod.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmod.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmod,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(fmod)
        push    %rbp
        movq    %rsp,%rbp
        subq    $16,%rsp
        movlpd  %xmm1,-16(%rbp)
        movlpd  %xmm0,-8(%rbp)

        movl    -12(%rbp),%eax          / eax <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|y|)
        orl     -16(%rbp),%eax          / eax <-- lo_32(y)|hi_32(|y|)
        je      .yzero

        fldl    -16(%rbp)               / y
        fldl    -8(%rbp)                / x
.loop:
        fprem                           / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .loop                   / loop while reduction incomplete
        fstpl   -8(%rbp)
        movsd   -8(%rbp),%xmm0
        fstp    %st(0)
        leave
        ret

.yzero:
```



```
        PIC_SETUP(1)
        movl    $27,%edi
        movl    $2,%eax
        call    PIC_F(_SVID_libm_err)
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(fmod)
______unchanged_portion_omitted_
```

```
**********************************************************
    1496 Tue Nov  4 18:59:25 2014
new/usr/src/lib/libm/amd64/src/fmodf.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmodf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmodf,function)
#include "libm_synonyms.h"

        ENTRY(fmodf)
        push    %rbp
        movq    %rsp,%rbp
        subq    $16,%rsp
        movss   %xmm1,-8(%rbp)
        movss   %xmm0,-4(%rbp)
        flds    -8(%rbp)                / load arg y
        flds    -4(%rbp)                / load arg x
.loop:
        fprem                           / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .loop                   / loop while reduction incomplete
        fstps   -4(%rbp)
        movss   -4(%rbp),%xmm0
        fstp    %st(0)
        leave
        ret
        .align  4
        SET_SIZE(fmodf)
______unchanged_portion_omitted_
```

****
1367 Tue Nov 4 18:59:25 2014
new/usr/src/lib/libm/amd64/src/fmodl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmodl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmodl,function)
#include "libm_synonyms.h"

        ENTRY(fmodl)
        fldt    24(%rsp)                / load arg y
        fldt    8(%rsp)                 / load arg x
.mod_loop:
        fprem                           / partial fmod
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .mod_loop               / while incomplete, do fprem again
        fstp    %st(1)
        ret
        .align  16
        SET_SIZE(fmodl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    3181 Tue Nov  4 18:59:26 2014
new/usr/src/lib/libm/amd64/src/ieee_funcl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "ieee_funcl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(isinfl,function)
LIBM_ANSI_PRAGMA_WEAK(isnormall,function)
LIBM_ANSI_PRAGMA_WEAK(issubnormall,function)
LIBM_ANSI_PRAGMA_WEAK(iszerol,function)
LIBM_ANSI_PRAGMA_WEAK(signbitl,function)
#include "libm_synonyms.h"

        ENTRY(isinfl)
        movl    16(%rsp),%eax           / ax <-- sign and bexp of x
        notl    %eax
        andq    $0x7fff,%rax
        jz      .L6
        movq    $0,%rax
.not_inf:
        ret

.L6:                                    / here, (eax) = 0.0
        movl    12(%rsp),%ecx
        xorl    $0x80000000,%ecx        / handle unsupported implicitly
        orl     8(%rsp), %ecx
        jnz     .not_inf
        movq    $1,%rax
        ret
        .align  16
        SET_SIZE(isinfl)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2378 Tue Nov  4 18:59:26 2014
new/usr/src/lib/libm/amd64/src/ilogbl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file "ilogbl.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(ilogbl,function)
  33 #include "libm_synonyms.h"
  33 #include "xpg6.h"

  35         .data
  36         .align  16
  37 two63:  .4byte  0x0,0x43d00000          / 2**63

  39         ENTRY(ilogbl)
  40         movq    16(%rsp),%rax           / eax <-- sign and bexp of x
  41         andq    $0x7fff,%rax            / eax <-- bexp(x)
  42         jz      .bexp_0                 / jump iff x is 0 or subnormal
  43                                         / here, biased exponent is non-zero
  44         testl   $0x80000000,12(%rsp)    / test msb of hi_32(sgnfcnd(x))
  45         jz      .ilogbl_not_finite      / jump if unsupported format
  46         cmpq    $0x7fff,%rax
  47         je      .ilogbl_not_finite
  48         subq    $16383,%rax             / unbias exponent by 16383 = 0x3fff
  49         ret

  51 .ilogbl_not_finite:
  52         movq    $0x7fffffff,%rax        / x is NaN/inf/unsup
  53         jmp     0f

  55 .bexp_0:
  56         movq    8(%rsp),%rax            / rax <-- sgnfcnd(x)
  57         orq     %rax,%rax
  58         jnz     .ilogbl_subnorm         / jump iff x is subnormal
  59         movq    $-2147483647,%rax       / x is +/-0, so return 1-2^31
  60 0:
```



```
  61         PIC_SETUP(0)
  62         PIC_G_LOAD(movzwq,__xpg6,rcx)
  63         PIC_WRAPUP
  64         andl    $_C99SUSv3_ilogb_0InfNaN_raises_invalid,%ecx
  65         cmpl    $0,%ecx
  66         je      1f
  67         fldz
  68         fdivp   %st,%st(0)              / raise invalid as per SUSv3
  69 1:
  70         ret


  73 .ilogbl_subnorm:                        / subnormal or pseudo-denormal input
  74         fldt    8(%rsp)                 / push x, setting D-flag
  75         PIC_SETUP(1)
  76         fmull   PIC_L(two63)            / x*2**63
  77         PIC_WRAPUP
  78         subq    $16,%rsp
  79         fstpt   (%rsp)
  80         movq    $0x7fff,%rax
  81         andq    8(%rsp),%rax            / eax <-- sign and bexp of x*2**63
  82         subq    $16445,%rax             / unbias it by (16,383 + 63)
  83         addq    $16,%rsp
  84         ret
  85         .align  16
  86         SET_SIZE(ilogbl)
______unchanged_portion_omitted_
```

```
**********************************************************
    1225 Tue Nov  4 18:59:26 2014
new/usr/src/lib/libm/amd64/src/log10l.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "log10l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log10l,function)
#include "libm_synonyms.h"

        ENTRY(log10l)
        fldlg2
        fldt    8(%rsp)                 / st = arg, st(1) = log10(2)
        fyl2x                           / st = log10(arg) = log10(2)*log2(arg)
        ret
        .align  16
        SET_SIZE(log10l)
______unchanged_portion_omitted_
```

****
1191 Tue Nov 4 18:59:27 2014
new/usr/src/lib/libm/amd64/src/log2l.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "log2l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log2l,function)
#include "libm_synonyms.h"

        ENTRY(log2l)
        fld1                    / push 1.0
        fldt    8(%rsp)         / push x
        fyl2x                   / st = 1.0*log2(arg)
        ret
        .align  16
        SET_SIZE(log2l)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1212 Tue Nov  4 18:59:27 2014
new/usr/src/lib/libm/amd64/src/logl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "logl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(logl,function)
#include "libm_synonyms.h"

        ENTRY(logl)
        fldln2
        fldt    8(%rsp)                 / st = arg, st(1) = loge(2)
        fyl2x                           / st = ln(arg) = loge(2)*log2(arg)
        ret
        .align  16
        SET_SIZE(logl)
______unchanged_portion_omitted_
```

```
**********************************************************
    9506 Tue Nov  4 18:59:28 2014
new/usr/src/lib/libm/amd64/src/powl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "powl.s"

/ Special cases:
/
/ x ** 0 is 1
/ 1 ** y is 1                           (C99)
/ x ** NaN is NaN
/ NaN ** y (except 0) is NaN
/ x ** 1 is x
/ +-(|x| > 1) **  +inf is +inf
/ +-(|x| > 1) **  -inf is +0
/ +-(|x| < 1) **  +inf is +0
/ +-(|x| < 1) **  -inf is +inf
/ (-1) ** +-inf is +1                   (C99)
/ +0 ** +y (except 0, NaN)              is +0
/ -0 ** +y (except 0, NaN, odd int)     is +0
/ +0 ** -y (except 0, NaN)              is +inf (z flag)
/ -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
/ -0 ** y (odd int)                     is - (+0 ** x)
/ +inf ** +y (except 0, NaN)            is +inf
/ +inf ** -y (except 0, NaN)            is +0
/ -inf ** +-y (except 0, NaN)           is -0 ** -+y (NO z flag)
/ x ** -1 is 1/x
/ x ** 2 is x*x
/ -x ** y (an integer) is (-1)**(y) * (+x)**(y)
/ x ** y (x negative & y not integer) is NaN (i flag)

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(powl,function)
#include "libm_synonyms.h"
#include "xpg6.h"

#undef fabs
```



```
        .data
        .align  16
negzero:
        .float  -0.0
half:
        .float  0.5
one:
        .float  1.0
negone:
        .float  -1.0
two:
        .float  2.0
Snan:
        .4byte  0x7f800001
pinfinity:
        .4byte  0x7f800000
ninfinity:
        .4byte  0xff800000


        ENTRY(powl)
        pushq   %rbp
        movq    %rsp,%rbp
        PIC_SETUP(1)

        fldt    16(%rbp)                / x
        fxam                            / determine class of x
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dh                 / %dh <- condition code of x

        fldt    32(%rbp)                / y , x
        fxam                            / determine class of y
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dl                 / %dl <- condition code of y

        call    .pow_main               /// LOCAL
        PIC_WRAPUP
        leave
        ret

.pow_main:
        / x ** 0 is 1
        movb    %dl,%cl
        andb    $0x45,%cl
        cmpb    $0x40,%cl               / C3=1 C2=0 C1=? C0=0 when +-0
        jne     1f
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fld1                            / 1
        ret

1:      / y is not zero
        PIC_G_LOAD(movzwq,__xpg6,rax)
        andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
        cmpl    $0,%eax
        je      1f

        / C99: 1 ** anything is 1
        fld1                            / 1, y, x
        fucomip %st(2),%st              / y, x
        jp      1f                      / so that pow(NaN1,NaN2) returns NaN2
        jne     1f
        fstp    %st(0)                  / x
        ret
```

```
 127 1:
 128         / x ** NaN is NaN
 129         movb    %dl,%cl
 130         andb    $0x45,%cl
 131         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
 132         jne     1f
 133         fstp    %st(1)                  / y
 134         ret

 136 1:      / y is not NaN
 137         / NaN ** y (except 0) is NaN
 138         movb    %dh,%cl
 139         andb    $0x45,%cl
 140         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
 141         jne     1f
 142         fstp    %st(0)                  / x
 143         ret

 145 1:      / x is not NaN
 146         / x ** 1 is x
 147         fld1                            / 1, y, x
 148         fcomip  %st(1),%st              / y, x
 149         jne     1f
 150         fstp    %st(0)                  / x
 151         ret

 153 1:      / y is not 1
 154         / +-(|x| > 1) **  +inf is +inf
 155         / +-(|x| > 1) **  -inf is +0
 156         / +-(|x| < 1) **  +inf is +0
 157         / +-(|x| < 1) **  -inf is +inf
 158         / +-(|x| = 1) ** +-inf is NaN
 159         movb    %dl,%cl
 160         andb    $0x47,%cl
 161         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
 162         je      .yispinf
 163         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
 164         je      .yisninf

 166         / +0 ** +y (except 0, NaN)              is +0
 167         / -0 ** +y (except 0, NaN, odd int)     is +0
 168         / +0 ** -y (except 0, NaN)              is +inf (z flag)
 169         / -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
 170         / -0 ** y (odd int)                     is - (+0 ** x)
 171         movb    %dh,%cl
 172         andb    $0x47,%cl
 173         cmpb    $0x40,%cl               / C3=1 C2=0 C1=0 C0=0 when +0
 174         je      .xispzero
 175         cmpb    $0x42,%cl               / C3=1 C2=0 C1=1 C0=0 when -0
 176         je      .xisnzero

 178         / +inf ** +y (except 0, NaN)    is +inf
 179         / +inf ** -y (except 0, NaN)    is +0
 180         / -inf ** +-y (except 0, NaN)   is -0 ** -+y (NO z flag)
 181         movb    %dh,%cl
 182         andb    $0x47,%cl
 183         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
 184         je      .xispinf
 185         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
 186         je      .xisninf

 188         / x ** -1 is 1/x
 189         flds    PIC_L(negone)           / -1, y, x
 190         fcomip  %st(1),%st              / y, x
 191         jne     1f
 192         fld     %st(1)                  / x , y , x
```



```
 193         fdivrs  PIC_L(one)              / 1/x , y , x
 194         jmp     .signok                 / check for over/underflow

 196 1:      / y is not -1
 197         / x ** 2 is x*x
 198         flds    PIC_L(two)              / 2, y , x
 199         fcomip  %st(1),%st              / y, x
 200         jne     1f
 201         fld     %st(1)                  / x , y , x
 202         fld     %st(0)                  / x , x , y , x
 203         fmulp                           / x^2 , y , x
 204         jmp     .signok                 / check for over/underflow

 206 1:      / y is not 2
 207         / x ** 1/2 is sqrt(x)
 208         flds    PIC_L(half)             / 1/2, y , x
 209         fcomip  %st(1),%st              / y, x
 210         jne     1f
 211         fld     %st(1)                  / x , y , x
 212         fsqrt                           / sqrt(x) , y , x
 213         jmp     .signok                 / check for over/underflow

 215 1:      / y is not 1/2
 216         / make copies of x & y
 217         fld     %st(1)                  / x , y , x
 218         fld     %st(1)                  / y , x , y , x

 220         / -x ** y (an integer) is (-1)**(y) * (+x)**(y)
 221         / x ** y (x negative & y not integer) is  NaN
 222         movl    $0,%ecx                 / track whether to flip sign of result
 223         fldz                            / 0 , y , x , y , x
 224         fcomip  %st(2),%st              / compare 0 with %st(2)
 225         jb      .merge                  / 0 < x
 226         / x < 0
 227         call    .y_is_int
 228         cmpl    $0,%ecx
 229         jne     1f
 230         / x < 0 & y != int so x**y = NaN (i flag)
 231         fstp    %st(0)                  / x , y , x
 232         fstp    %st(0)                  / y , x
 233         fstp    %st(0)                  / x
 234         fstp    %st(0)                  / stack empty
 235         fldz
 236         fdiv    %st,%st(0)              / 0/0
 237         ret

 239 1:      / x < 0 & y = int
 240         fxch                            / x , y , y , x
 241         fchs                            / px = -x , y , y , x
 242         fxch                            / y , px , y , x
 243 .merge:
 244         / px > 0
 245         fxch                            / px , y , y , x

 247         / x**y   =   exp(y*ln(x))
 248         fyl2x                           / t=y*log2(px) , y , x
 249         fld     %st(0)                  / t , t , y , x
 250         frndint                         / [t] , t , y , x
 251         fxch                            / t , [t] , y , x
 252         fucomi  %st(1),%st
 253         je      1f                      / t is integral
 254         fsub    %st(1),%st              / t-[t] , [t] , y , x
 255         f2xm1                           / 2**(t-[t])-1 , [t] , y , x
 256         fadds   PIC_L(one)              / 2**(t-[t]) , [t] , y , x
 257         fscale                          / 2**t = px**y , [t] , y , x
 258         jmp     2f
```

```
 1:
        fstp    %st(0)                  / t=[t] , y , x
        fld1                            / 1 , t , y , x
        fscale                          / 1*2**t = x**y , t , y , x
 2:
        fstp    %st(1)                  / x**y , y , x
        cmpl    $1,%ecx
        jne     .signok
        fchs                            / change sign since x<0 & y=-int
.signok:
        fstp    %st(2)                  / y , x**y
        fstp    %st(0)                  / x**y
        ret

/ ------------------------------------------------------------------------

.xispinf:
        fldz
        fcomip  %st(1),%st              / compare 0 with %st(1)
        jb      .retpinf                / 0 < y
        jmp     .retpzero               / y < 0

.xisninf:
        / -inf ** +-y is -0 ** -+y
        fchs                            / -y , x
        flds    PIC_L(negzero)          / -0 , -y , x
        fstp    %st(2)                  / -y , -0
        jmp     .xisnzero

.yispinf:
        fld     %st(1)                  / x , y , x
        fabs                            / |x| , y , x
        flds    PIC_L(one)              / 1 , |x| , y , x
        fcomip  %st(1),%st              / |x| , y , x
        fstp    %st(0)                  / y , x
        je      .retponeorinvalid       / x == -1       C99
        jb      .retpinf                / 1 < |x|
        jmp     .retpzero               / |x| < 1

.yisninf:
        fld     %st(1)                  / x , y , x
        fabs                            / |x| , y , x
        flds    PIC_L(one)              / 1 , |x| , y , x
        fcomip  %st(1),%st              / |x| , y , x
        fstp    %st(0)                  / y , x
        je      .retponeorinvalid       / x == -1       C99
        jb      .retpzero               / 1 < |x|
        jmp     .retpinf                / |x| < 1

.xispzero:
        / y cannot be 0 or NaN ; stack has      y , x
        fldz                            / 0 , y , x
        fcomip  %st(1),%st              / compare 0 with %st(1)
        jb      .retpzero               / 0 < y
        / x = +0 & y < 0 so x**y = +inf
        jmp     .retpinfzflag           / ret +inf & z flag

.xisnzero:
        / y cannot be 0 or NaN ; stack has      y , x
        call    .y_is_int
        cmpl    $1,%ecx
        jne     1f                      / y is not an odd integer
        / y is an odd integer
        fldz
        fcomip  %st(1),%st              / compare 0 with %st(1)
        jb      .retnzero               / 0 < y
```



```
        / x = -0 & y < 0 (odd int)      return -inf (z flag)
        / x = -inf & y != 0 or NaN      return -inf (NO z flag)
        movb    %dh,%cl
        andb    $0x45,%cl
        cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
        je      2f
        fdiv    %st,%st(1)              / y / x, x (raise z flag)
 2:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        flds    PIC_L(ninfinity)        / -inf
        ret

 1:     / y is not an odd integer
        fldz
        fcomip  %st(1),%st              / compare 0 with %st(1)
        jb      .retpzero               / 0 < y
        / x = -0 & y < 0 (not odd int)  return +inf (z flag)
        / x = -inf & y not 0 or NaN     return +inf (NO z flag)
        movb    %dh,%cl
        andb    $0x45,%cl
        cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
        jne     .retpinfzflag           / ret +inf & divide-by-0 flag
        jmp     .retpinf                / return +inf (NO z flag)

.retpzero:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fldz                            / +0
        ret

.retnzero:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        flds    PIC_L(negzero)          / -0
        ret

.retponeorinvalid:
        PIC_G_LOAD(movzwq,__xpg6,rax)
        andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
        cmpl    $0,%eax
        je      1f
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fld1                            / 1
        ret

 1:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        flds    PIC_L(Snan)             / Q NaN (i flag)
        fwait
        ret

.retpinf:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        flds    PIC_L(pinfinity)        / +inf
        ret

.retpinfzflag:
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fldz
        fdivrs  PIC_L(one)              / 1/0
        ret
```

```
392 / Set %ecx to 2 if y is an even integer, 1 if y is an odd integer,
393 / 0 otherwise.  Assume y is not zero.  Do not raise inexact or modify
394 / %edx.
395 .y_is_int:
396         movl    40(%rbp),%eax
397         andl    $0x7fff,%eax            / exponent of y
398         cmpl    $0x403f,%eax
399         jae     1f                      / |y| >= 2^64, an even int
400         cmpl    $0x3fff,%eax
401         jb      2f                      / |y| < 1, can't be an int
402         movl    %eax,%ecx
403         subl    $0x403e,%ecx
404         negl    %ecx                    / 63 - unbiased exponent of y
405         movq    32(%rbp),%rax
406         bsfq    %rax,%rax               / index of least sig. 1 bit
407         cmpl    %ecx,%eax
408         jb      2f
409         ja      1f
410         movl    $1,%ecx
411         ret
412 1:
413         movl    $2,%ecx
414         ret
415 2:
416         xorl    %ecx,%ecx
417         ret
418         .align  16
419         SET_SIZE(powl)
```

*unchanged_portion_omitted_*

```
**********************************************************
    2016 Tue Nov  4 18:59:28 2014
new/usr/src/lib/libm/amd64/src/remainder.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainder.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainder,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(remainder)
        push    %rbp
        movq    %rsp,%rbp
        subq    $16,%rsp
        movlpd  %xmm1,-16(%rbp)
        movlpd  %xmm0,-8(%rbp)

        ucomisd %xmm0,%xmm1             / if x or y is NaN, use fprem1
        jp      1f

        movl    -12(%rbp),%eax          / eax <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|y|)
        orl     -16(%rbp),%eax          / eax <-- lo_32(y)|hi_32(|y|)
        je      .yzero_or_xinf

        movl    -4(%rbp),%eax           / eax <-- hi_32(x)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|x|)
        cmpl    $0x7ff00000,%eax
        jne     1f
        cmpl    $0,-8(%rbp)
        je      .yzero_or_xinf
1:
        fldl    -16(%rbp)               / y
        fldl    -8(%rbp)                / x
.rem_loop:
        fprem1                          / partial remainder
```



```
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .rem_loop               / while incomplete, do fprem1 again
        fstpl   -8(%rbp)
        movsd   -8(%rbp),%xmm0
        fstp    %st(0)
        leave
        ret

.yzero_or_xinf:
        PIC_SETUP(1)
        movl    $28,%edi
        movl    $2,%eax
        call    PIC_F(_SVID_libm_err)
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(remainder)
______unchanged_portion_omitted_
```

****
1530 Tue Nov 4 18:59:29 2014
new/usr/src/lib/libm/amd64/src/remainderf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainderf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainderf,function)
#include "libm_synonyms.h"

        ENTRY(remainderf)
        push    %rbp
        movq    %rsp,%rbp
        subq    $16,%rsp
        movss   %xmm1,-8(%rbp)
        movss   %xmm0,-4(%rbp)
        flds    -8(%rbp)                / load arg y
        flds    -4(%rbp)                / load arg x
.rem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .rem_loop               / while reduction incomplete, do fprem1
        fstps   -4(%rbp)
        movss   -4(%rbp),%xmm0
        fstp    %st(0)
        leave
        ret
        .align  4
        SET_SIZE(remainderf)
```

unchanged_portion_omitted_

****
1400 Tue Nov 4 18:59:29 2014
new/usr/src/lib/libm/amd64/src/remainderl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainderl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainderl,function)
#include "libm_synonyms.h"

        ENTRY(remainderl)
        fldt    24(%rsp)                / load arg y
        fldt    8(%rsp)                 / load arg x
.rem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .rem_loop               / while reduction incomplete, do fprem1
        fstp    %st(1)
        ret
        .align  16
        SET_SIZE(remainderl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1879 Tue Nov  4 18:59:29 2014
new/usr/src/lib/libm/amd64/src/remquol.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "remquol.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remquol,function)
#include "libm_synonyms.h"
#include "libm_protos.h"
        ENTRY(remquol)
        fldt    24(%rsp)                / load arg y
        fldt    8(%rsp)                 / load arg x
.Lreml_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .Lreml_loop             / while reduction incomplete, do fprem1
        fstsw   %ax
        fwait
        fstp    %st(1)
        movw    %ax,%dx
        andw    $0x4000,%dx             / get C3
        sarw    $13,%dx
        movw    %ax,%cx
        andw    $0x100,%cx              / get C0
        sarw    $6,%cx
        addw    %cx,%dx
        andw    $0x200,%ax              / get C1
        sarw    $9,%ax
        addw    %dx,%ax
        cwtl
        movl    16(%rsp),%edx           / sign and bexp of x
        movl    32(%rsp),%ecx           / sign and bexp of y
        andl    $0x8000,%edx            / edx <- sign(x)
        andl    $0x8000,%ecx            / ecx <- sign(y)
        cmpl    %edx,%ecx
```



```
        je      1f
        negl    %eax                    / negative n
1:
        movl    %eax,(%rdi)             / last 3 significant bits of quotient
        ret
        .align  16
        SET_SIZE(remquol)
______unchanged_portion_omitted_
```

****
1187 Tue Nov 4 18:59:30 2014
new/usr/src/lib/libm/amd64/src/rintl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "rintl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(rintl,function)
#include "libm_synonyms.h"

        ENTRY(rintl)
        fldt    8(%rsp)                 / load x
        frndint                         / [x], per rounding mode
        fwait
        ret
        .align  16
        SET_SIZE(rintl)
```

*unchanged_portion_omitted_*

```
**********************************************************
    2928 Tue Nov  4 18:59:30 2014
new/usr/src/lib/libm/amd64/src/rndintl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "rndintl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(aintl,function)
LIBM_ANSI_PRAGMA_WEAK(irintl,function)
LIBM_ANSI_PRAGMA_WEAK(anintl,function)
LIBM_ANSI_PRAGMA_WEAK(nintl,function)
#include "libm_synonyms.h"

#undef fabs

        ENTRY(aintl)
        movq    %rsp,%rax
        subq    $16,%rsp
        fstcw   -8(%rax)
        fldt    8(%rax)
        movw    -8(%rax),%cx
        orw     $0x0c00,%cx
        movw    %cx,-4(%rax)
        fldcw   -4(%rax)                / set RD = to_zero
        frndint
        fstcw   -4(%rax)
        movw    -4(%rax),%dx
        andw    $0xf3ff,%dx
        movw    -8(%rax),%cx
        andw    $0x0c00,%cx
        orw     %dx,%cx
        movw    %cx,-8(%rax)
        fldcw   -8(%rax)                / restore RD
        addq    $16,%rsp
        ret
        .align  16
        SET_SIZE(aintl)
______unchanged_portion_omitted_
```

****
1224 Tue Nov 4 18:59:31 2014
new/usr/src/lib/libm/amd64/src/scalbnl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "scalbnl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(scalbnl,function)
#include "libm_synonyms.h"

        ENTRY(scalbnl)
        subq    $16,%rsp
        movl    %edi,(%rsp)
        fildl   (%rsp)
        fldt    24(%rsp)
        addq    $16,%rsp
        fscale
        fstp    %st(1)
        ret
        .align  16
        SET_SIZE(scalbnl)
```

unchanged_portion_omitted_

****
1139 Tue Nov 4 18:59:31 2014
new/usr/src/lib/libm/amd64/src/sqrtl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "sqrtl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(sqrtl,function)
#include "libm_synonyms.h"

        ENTRY(sqrtl)
        fldt    8(%rsp)
        fsqrt
        ret
        .align  16
        SET_SIZE(sqrtl)
```

_____unchanged_portion_omitted_

```
**********************************************************
    4619 Tue Nov  4 18:59:31 2014
new/usr/src/lib/libm/common/C/acos.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __acos = acos
#pragma weak acos = __acos

/* INDENT OFF */
/*
 * acos(x)
 * Method :
 *      acos(x)  = pi/2 - asin(x)
 *      acos(-x) = pi/2 + asin(x)
 * For |x|<=0.5
 *      acos(x) = pi/2 - (x + x*x^2*R(x^2))     (see asin.c)
 * For x>0.5
 *      acos(x) = pi/2 - (pi/2 - 2asin(sqrt((1-x)/2)))
 *              = 2asin(sqrt((1-x)/2))
 *              = 2s + 2s*z*R(z)        ...z=(1-x)/2, s=sqrt(z)
 *              = 2f + (2c + 2s*z*R(z))
 *     where f=hi part of s, and c = (z-f*f)/(s+f) is the correction term
 *     for f so that f+c ~ sqrt(z).
 * For x<-0.5
 *      acos(x) = pi - 2asin(sqrt((1-|x|)/2))
 *              = pi - 0.5*(s+s*z*R(z)), where z=(1-|x|)/2,s=sqrt(z)
 *
 * Special cases:
 *      if x is NaN, return x itself;
 *      if |x|>1, return NaN with invalid signal.
 *
 * Function needed: sqrt
 */
/* INDENT ON */

#include "libm_synonyms.h"      /* __acos, __sqrt, __isnan */
#include "libm_protos.h"        /* _SVID_libm_error */
```



```
#include "libm_macros.h"
#include <math.h>

/* INDENT OFF */
static const double xxx[] = {
/* one */        1.00000000000000000000e+00,    /* 3FF00000, 00000000 */
/* pi */         3.14159265358979311600e+00,    /* 400921FB, 54442D18 */
/* pio2_hi */    1.57079632679489655800e+00,    /* 3FF921FB, 54442D18 */
/* pio2_lo */    6.12323399573676603587e-17,    /* 3C91A626, 33145C07 */
/* pS0 */        1.66666666666666657415e-01,    /* 3FC55555, 55555555 */
/* pS1 */       -3.25565818622400915405e-01,    /* BFD4D612, 03EB6F7D */
/* pS2 */        2.01212532134862925881e-01,    /* 3FC9C155, 0E884455 */
/* pS3 */       -4.00555345006794114027e-02,    /* BFA48228, B5688F3B */
/* pS4 */        7.91534994289814532176e-04,    /* 3F49EFE0, 7501B288 */
/* pS5 */        3.47933107596021167570e-05,    /* 3F023DE1, 0DFDF709 */
/* qS1 */       -2.40339491173441421878e+00,    /* C0033A27, 1C8A2D4B */
/* qS2 */        2.02094576023350569471e+00,    /* 40002AE5, 9C598AC8 */
/* qS3 */       -6.88283971605453293030e-01,    /* BFE6066C, 1B8D0159 */
/* qS4 */        7.70381505559019352791e-02     /* 3FB3B8C5, B12E9282 */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    2562 Tue Nov  4 18:59:32 2014
new/usr/src/lib/libm/common/C/acosh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __acosh = acosh
#pragma weak acosh = __acosh

/* INDENT OFF */
/*
 * acosh(x)
 * Method :
 *      Based on
 *              acosh(x) = log [ x + sqrt(x*x-1) ]
 *      we have
 *              acosh(x) := log(x)+ln2, if x is large; else
 *              acosh(x) := log(2x-1/(sqrt(x*x-1)+x)) if x > 2; else
 *              acosh(x) := log1p(t+sqrt(2.0*t+t*t)); where t = x-1.
 *
 * Special cases:
 *      acosh(x) is NaN with signal if x < 1.
 *      acosh(NaN) is NaN without signal.
 */
/* INDENT ON */

#include "libm_synonyms.h"      /* __acosh, __log, __log1p */
#include "libm_protos.h"        /* _SVID_libm_error */
#include "libm_macros.h"
#include <math.h>

static const double
        one = 1.0,
        ln2 = 6.93147180559945286227e-01;       /* 3FE62E42, FEFA39EF */

double
acosh(double x) {
        double t;
```



```
        int hx;

        hx = ((int *) &x)[HIWORD];
        if (hx < 0x3ff00000) {  /* x < 1 */
                if (isnan(x))
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                        return (hx >= 0xfff80000 ? x : (x - x) / (x - x));
                        /* assumes sparc-like QNaN */
#else
                        return (x - x) / (x - x);
#endif
                else
                        return (_SVID_libm_err(x, x, 29));
        } else if (hx >= 0x41b00000) {
                /* x > 2**28 */
                if (hx >= 0x7ff00000) { /* x is inf of NaN */
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                        return (hx >= 0x7ff80000 ? x : x + x);
                        /* assumes sparc-like QNaN */
#else
                        return (x + x);
#endif
                } else  /* acosh(huge)=log(2x) */
                        return (log(x) + ln2);
        } else if (((hx - 0x3ff00000) | ((int *) &x)[LOWORD]) == 0) {
                return (0.0);   /* acosh(1) = 0 */
        } else if (hx > 0x40000000) {
                /* 2**28 > x > 2 */
                t = x * x;
                return (log(2.0 * x - one / (x + sqrt(t - one))));
        } else {
                /* 1 < x < 2 */
                t = x - one;
                return (log1p(t + sqrt(2.0 * t + t * t)));
        }
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    4821 Tue Nov  4 18:59:32 2014
new/usr/src/lib/libm/common/C/asin.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asin = asin
#pragma weak asin = __asin

/* INDENT OFF */
/*
 * asin(x)
 * Method :
 *      Since  asin(x) = x + x^3/6 + x^5*3/40 + x^7*15/336 + ...
 *      we approximate asin(x) on [0,0.5] by
 *              asin(x) = x + x*x^2*R(x^2)
 *      where
 *              R(x^2) is a rational approximation of (asin(x)-x)/x^3
 *      and its remez error is bounded by
 *              |(asin(x)-x)/x^3 - R(x^2)| < 2^(-58.75)
 *
 *      For x in [0.5,1]
 *              asin(x) = pi/2-2*asin(sqrt((1-x)/2))
 *      Let y = (1-x), z = y/2, s := sqrt(z), and pio2_hi+pio2_lo=pi/2;
 *      then for x>0.98
 *              asin(x) = pi/2 - 2*(s+s*z*R(z))
 *                      = pio2_hi - (2*(s+s*z*R(z)) - pio2_lo)
 *      For x<=0.98, let pio4_hi = pio2_hi/2, then
 *              f = hi part of s;
 *              c = sqrt(z) - f = (z-f*f)/(s+f)          ...f+c=sqrt(z)
 *      and
 *              asin(x) = pi/2 - 2*(s+s*z*R(z))
 *                      = pio4_hi+(pio4-2s)-(2s*z*R(z)-pio2_lo)
 *                      = pio4_hi+(pio4-2f)-(2s*z*R(z)-(pio2_lo+2c))
 *
 * Special cases:
 *      if x is NaN, return x itself;
 *      if |x|>1, return NaN with invalid signal.
```



```
 *
 */
/* INDENT ON */

#include "libm_synonyms.h"      /* __asin, __sqrt, __isnan */
#include "libm_protos.h"        /* _SVID_libm_error */
#include "libm_macros.h"
#include <math.h>

/* INDENT OFF */
static const double xxx[] = {
/* one */       1.00000000000000000000e+00,    /* 3FF00000, 00000000 */
/* huge */      1.000e+300,
/* pio2_hi */   1.57079632679489655800e+00,    /* 3FF921FB, 54442D18 */
/* pio2_lo */   6.12323399573676603587e-17,    /* 3C91A626, 33145C07 */
/* pio4_hi */   7.85398163397448278999e-01,    /* 3FE921FB, 54442D18 */
/* coefficient for R(x^2) */
/* pS0 */       1.66666666666666657415e-01,    /* 3FC55555, 55555555 */
/* pS1 */      -3.25565818622400915405e-01,    /* BFD4D612, 03EB6F7D */
/* pS2 */       2.01212532134862925881e-01,    /* 3FC9C155, 0E884455 */
/* pS3 */      -4.00555345006794114027e-02,    /* BFA48228, B5688F3B */
/* pS4 */       7.91534994289814532176e-04,    /* 3F49EFE0, 7501B288 */
/* pS5 */       3.47933107596021167570e-05,    /* 3F023DE1, 0DFDF709 */
/* qS1 */      -2.40339491173441421878e+00,    /* C0033A27, 1C8A2D4B */
/* qS2 */       2.02094576023350569471e+00,    /* 40002AE5, 9C598AC8 */
/* qS3 */      -6.88283971605453293030e-01,    /* BFE6066C, 1B8D0159 */
/* qS4 */       7.70381505559019352791e-02     /* 3FB3B8C5, B12E9282 */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    2393 Tue Nov  4 18:59:33 2014
new/usr/src/lib/libm/common/C/asinh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinh = asinh
#pragma weak asinh = __asinh

/* INDENT OFF */
/*
 * asinh(x)
 * Method :
 *      Based on
 *              asinh(x) = sign(x) * log [ |x| + sqrt(x*x+1) ]
 *      we have
 *      asinh(x) := x  if  1+x*x == 1,
 *               := sign(x)*(log(x)+ln2)) for large |x|, else
 *               := sign(x)*log(2|x|+1/(|x|+sqrt(x*x+1))) if|x| > 2, else
 *               := sign(x)*log1p(|x|+x^2/(1+sqrt(1+x^2)))
 */
/* INDENT ON */

#include "libm_synonyms.h"      /* __asinh */
#include "libm_macros.h"
#include <math.h>

static const double xxx[] = {
/* one */       1.00000000000000000000e+00,     /* 3FF00000, 00000000 */
/* ln2 */       6.93147180559945286227e-01,     /* 3FE62E42, FEFA39EF */
/* huge */      1.00000000000000000000e+300
};
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    5990 Tue Nov  4 18:59:34 2014
new/usr/src/lib/libm/common/C/atan.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atan = atan
#pragma weak atan = __atan

/* INDENT OFF */
/*
 * atan(x)
 * Accurate Table look-up algorithm with polynomial approximation in
 * partially product form.
 *
 * -- K.C. Ng, October 17, 2004
 *
 * Algorithm
 *
 * (1). Purge off Inf and NaN and 0
 * (2). Reduce x to positive by atan(x) = -atan(-x).
 * (3). For x <= 1/8 and let z = x*x, return
 *      (2.1) if x < 2^(-prec/2), atan(x) = x  with inexact flag raised
 *      (2.2) if x < 2^(-prec/4-1), atan(x) = x+(x/3)(x*x)
 *      (2.3) if x < 2^(-prec/6-2), atan(x) = x+(z-5/3)(z*x/5)
 *      (2.4) Otherwise
 *              atan(x) = poly1(x) = x + A * B,
 *      where
 *              A = (p1*x*z) * (p2+z(p3+z))
 *              B = (p4+z)+z*z) * (p5+z(p6+z))
 *      Note: (i) domain of poly1 is [0, 1/8], (ii) remez relative
 *      approximation error of poly1 is bounded by
 *              |(atan(x)-poly1(x))/x| <= 2^-57.61
 * (4). For x >= 8 then
 *      (3.1) if x >= 2^prec,     atan(x) = atan(inf) - pio2lo
 *      (3.2) if x >= 2^(prec/3), atan(x) = atan(inf) - 1/x
 *      (3.3) if x <= 65,         atan(x) = atan(inf) - poly1(1/x)
 *      (3.4) otherwise           atan(x) = atan(inf) - poly2(1/x)
```



```
 *      where
 *              poly2(r) = (q1*r) * (q2+z(q3+z)) * (q4+z),
 *      its domain is [0, 0.0154]; and its remez absolute
 *      approximation error is bounded by
 *              |atan(x)-poly2(x)|<= 2^-59.45
 *
 * (5). Now x is in (0.125, 8).
 *      Recall identity
 *              atan(x) = atan(y) + atan((x-y)/(1+x*y)).
 *      Let j = (ix - 0x3fc00000) >> 16, 0 <= j < 96, where ix is the high
 *      part of x in IEEE double format. Then
 *              atan(x) = atan(y[j]) + poly2((x-y[j])/(1+x*y[j]))
 *      where y[j] are carefully chosen so that it matches x to around 4.5
 *      bits and at the same time atan(y[j]) is very close to an IEEE double
 *      floating point number. Calculation indicates that
 *              max|(x-y[j])/(1+x*y[j])| < 0.0154
 *              j,x
 *
 * Accuracy: Maximum error observed is bounded by 0.6 ulp after testing
 * more than 10 million random arguments
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"

extern const double _TBL_atan[];
static const double g[] = {
/* one  = */  1.0,
/* p1   = */  8.02176624254765935351230154992663301527500152588e-0002,
/* p2   = */  1.27223421700559402580665846471674740314483642578e+0000,
/* p3   = */ -1.20606901800503640842521235754247754812240600586e+0000,
/* p4   = */ -2.36088967922325565496066701598465442657470703125e+0000,
/* p5   = */  1.38345799501389166152875986881554126739501953125e+0000,
/* p6   = */  1.06742368078953453469637224770849570631980895996e+0000,
/* q1   = */ -1.42796626333911796935538518482644576579332351685e-0001,
/* q2   = */  3.51427110447873227059810477159863497078605962912e+0000,
/* q3   = */  5.92129112708164262457444237952586263418197631836e-0001,
/* q4   = */ -1.99272234785683144409063061175402253866195678711e+0000,
/* pio2hi */  1.57079632794896558e+00,
/* pio2lo */  6.12323399573676588 6e-17,
/* t1   = */ -0.33333333333333333333333333333333,
/* t2   = */  0.2,
/* t3   = */ -1.66666666666666666666666666666666,
};
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   18614 Tue Nov  4 18:59:34 2014
new/usr/src/lib/libm/common/C/atan2.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atan2 = atan2
#pragma weak atan2 = __atan2

#include "libm.h"

/*
 * Let t(0) = 1 and for i = 1, ..., 160, let t(i) be the slope of
 * the line bisecting the conical hull of the set of points (x,y)
 * where x and y are positive normal floating point numbers and
 * the high order words hx and hy of their binary representations
 * satisfy |hx - hy - i * 0x8000| <= 0x4000.  Then:
 *
 * TBL[4*i+2] is t(i) rounded to 21 significant bits (i.e., the
 *   low order word is zero), and
 *
 * TBL[4*i] + TBL[4*i+1] form a doubled-double approximation to
 *   atan(TBL[4*i+2]).
 *
 * Finally, TBL[4*161] = TBL[4*161+1] = TBL[4*161+2] = 0.
 *
 * Now for any (x,y) with 0 < y <= x and any 0 < t <= 1,  we have
 * atan(y/x) = atan(t) + atan((y-t*x)/(x+t*y)).  By choosing t =
 * TBL[4*i+2], where i is the multiple of 0x8000 nearest hx - hy,
 * if this multiple is less than 161, and i = 161 otherwise, we
 * find that |(y-t*x)/(x+t*y)| <~ 2^-5.
 */
static const double TBL[] = {
        7.85398163397448279000e-01, +3.06161699786838301790e-17,
        1.00000000000000000000e+00, +0,
        7.71989050126506112140e-01, +2.69899569600831536520e-16,
        9.73535060882568359380e-01, +0,
        7.60681439544613091640e-01, -3.51788105189419149720e-16,
        9.51749324798583984380e-01, +0,
```



```
        7.49536618763536388600e-01, -3.25481000045243374760e-16,
        9.30731296539306640620e-01, +0,
        7.38546149847283395220e-01, -2.07755710239104066680e-16,
        9.10427093505859375000e-01, +0,
        7.27701469620413377020e-01, +3.88832494031683488020e-16,
        8.90786647796630859380e-01, +0,
        7.16994924880937745120e-01, -4.04688415115472240710e-16,
        8.71765136718750000000e-01, +0,
        7.06418134886531490220e-01, +5.69024243539814840310e-17,
        8.53319644927978515620e-01, +0,
        6.95963511010356583600e-01, +2.82455133210750213030e-16,
        8.35411548614501953120e-01, +0,
        6.85623636805349434550e-01, -4.23169707216588540640e-16,
        8.18005084991455078120e-01, +0,
        6.75390556664382302190e-01, +4.35359172813000472330e-16,
        8.01065921783447265620e-01, +0,
        6.65257633469318321320e-01, +1.18304316024047279770e-17,
        7.84564018249511718750e-01, +0,
        6.55217675743101857220e-01, -1.74359231006510442080e-16,
        7.68470287322998046880e-01, +0,
        6.45263909994818973810e-01, -1.47419274030939839470e-16,
        7.52758026123046875000e-01, +0,
        6.35389798942048500410e-01, +1.57345350699956608530e-16,
        7.37402439117431640620e-01, +0,
        6.25589143469427177990e-01, -2.81755888563169109600e-16,
        7.22380638122558593750e-01, +0,
        6.15855864761579496760e-01, -4.30561673577252264490e-16,
        7.07671165466303593750e-01, +0,
        6.06184080275760983620e-01, +1.50180139184293202890e-16,
        6.93253993988037109380e-01, +0,
        5.96568178274867300100e-01, +5.52719420335576441570e-17,
        6.79110527038574218750e-01, +0,
        5.87002890834265045330e-01, -8.24113692826763832930e-17,
        6.65223598480224609380e-01, +0,
        5.77483030536276586990e-01, +4.94003837757091595580e-17,
        6.51576995846009375000e-01, +0,
        5.68003539683032521170e-01, +2.99244311033111095430e-16,
        6.38155460357666015620e-01, +0,
        5.58559538634938235190e-01, -2.03060034038687774030e-16,
        6.24944686889648437500e-01, +0,
        5.49147067083296747110e-01, +2.82553786137796674610e-17,
        6.11932277679443359380e-01, +0,
        5.39761766606180692920e-01, +1.63702487810787479950e-16,
        5.99105834960937500000e-01, +0,
        5.30398886014123327470e-01, -7.61960973600936801340e-17,
        5.86452960968017578120e-01, +0,
        5.21055439243188089900e-01, -2.24008156681547395610e-16,
        5.73964118957519531250e-01, +0,
        5.11727788739670508280e-01, -3.68881360198996811850e-16,
        5.61629295349121093750e-01, +0,
        5.02411996664521964820e-01, -2.54128914743970112810e-16,
        5.49438476562500000000e-01, +0,
        4.93104939542937437120e-01, +4.41321861282511522290e-16,
        5.37383079528808593750e-01, +0,
        4.83804368447509958170e-01, -2.78443879077766564880e-16,
        5.25455951690673828120e-01, +0,
        4.74506703614637537210e-01, -2.04943551973682860280e-16,
        5.13648509979248046880e-01, +0,
        4.63676600279763206910e-01, +3.17098786079547606680e-16,
        5.00036239624023437500e-01, +0,
        4.53047531040039253010e-01, +3.35934361224205748650e-16,
        4.86819267272949218750e-01, +0,
        4.44236580374070655350e-01, +2.19871831920080820150e-17,
        4.75966453552246093750e-01, +0,
        4.35670169725002942580e-01, +3.01184228053695526500e-16,
        4.65501785278320312500e-01, +0,
```

```
 127         4.2733152672544871820e-01, -3.2667693224866479909e-16,
 128         4.5539522171020507812e-01, +0,
 129         4.1920540176693954493e-01, -2.2454273841113897647e-16,
 130         4.4561982154846191406e-01, +0,
 131         4.1127722812701872357e-01, -3.1620568973494653391e-16,
 132         4.3615055084228515625e-01, +0,
 133         4.0353384063084263289e-01, -3.5932009901481421723e-16,
 134         4.2696499824523925781e-01, +0,
 135         3.9596319345246833166e-01, -4.0281533417458698585e-16,
 136         4.1804289817810058594e-01, +0,
 137         3.8855405220339722661e-01, +1.6132231486045176674e-16,
 138         4.0936565399169921875e-01, +0,
 139         3.8129566313738116889e-01, +1.7684657060650804570e-16,
 140         4.0091586112976074219e-01, +0,
 141         3.7417884791401867517e-01, +2.6897604227426977619e-16,
 142         3.9267849922180175781e-01, +0,
 143         3.6719421967585041955e-01, -4.5886151448673745001e-17,
 144         3.8463878631591796875e-01, +0,
 145         3.6033388248727771241e-01, +1.5804115573136074946e-16,
 146         3.7678408622741699219e-01, +0,
 147         3.5358982224579182940e-01, +1.2624619863035782939e-16,
 148         3.6910200119018554688e-01, +0,
 149         3.4695498404186952968e-01, +9.3221684607372865177e-17,
 150         3.6158156394958496094e-01, +0,
 151         3.4042268308109679964e-01, +2.7697913559445449137e-16,
 152         3.5421252250671386719e-01, +0,
 153         3.3398684598563566084e-01, +3.6085337449716011085e-16,
 154         3.4698557853698730469e-01, +0,
 155         3.2764182824591436827e-01, +2.0581506352606456186e-16,
 156         3.3989214897155761719e-01, +0,
 157         3.2138200938788497041e-01, -1.9015787485430693661e-16,
 158         3.3292388916015625000e-01, +0,
 159         3.1520245348069497737e-01, +2.6961839659264087022e-16,
 160         3.2607340812683105469e-01, +0,
 161         3.0909871873117023000e-01, -1.5641891686756272625e-16,
 162         3.1933403015136718750e-01, +0,
 163         3.0306644308947827682e-01, +2.8801634211591956223e-16,
 164         3.1269931793212890625e-01, +0,
 165         2.9710135482774191473e-01, -4.3148994478973365819e-16,
 166         3.0616307258605957031e-01, +0,
 167         2.9120015759141004708e-01, -6.8539854790808585159e-17,
 168         2.9972028732299804688e-01, +0,
 169         2.8535879880370362827e-01, -1.2231638445300492682e-16,
 170         2.9336524009704589844e-01, +0,
 171         2.7957422506893880865e-01, -4.6707752931043135528e-17,
 172         2.8709340095520019531e-01, +0,
 173         2.7384352102802367313e-01, -4.1215636366229625876e-16,
 174         2.8090047836303710938e-01, +0,
 175         2.6816369484161040049e-01, -2.3700583122400495333e-16,
 176         2.7478218078613281250e-01, +0,
 177         2.6253212627627764419e-01, +2.3123213692190889610e-16,
 178         2.6873469352722167969e-01, +0,
 179         2.5694635355759309903e-01, -4.0638513814701264145e-16,
 180         2.6275444030761718750e-01, +0,
 181         2.5140385572454615470e-01, -3.4795333793554943723e-16,
 182         2.5683784484863281250e-01, +0,
 183         2.4500357070096612233e-01, +6.6542334848010259289e-17,
 184         2.5002646446228027344e-01, +0,
 185         2.3877666609573036760e-01, -2.7756633678549343650e-16,
 186         2.4342155456542968750e-01, +0,
 187         2.3365669377188336142e-01, +3.2700803838522067998e-16,
 188         2.3800384998321533203e-01, +0,
 189         2.2870810463931334766e-01, -4.4279127662219799521e-16,
 190         2.3278105258941650391e-01, +0,
 191         2.2391820542294382790e-01, +3.7558889374284208052e-16,
 192         2.2773718833923339844e-01, +0,
```



```
 193         2.1927501815429550902e-01, -1.4829838176513811186e-16,
 194         2.2285830974578857422e-01, +0,
 195         2.1476740847367459253e-01, -2.0535381496063397578e-17,
 196         2.1813154220581054688e-01, +0,
 197         2.1038568111737454558e-01, -4.2826767738736168650e-16,
 198         2.1354568004608154297e-01, +0,
 199         2.0612057974373865221e-01, +4.2108051749502232359e-16,
 200         2.0909011363983154297e-01, +0,
 201         2.0196410359405447821e-01, +3.5157118083511092869e-16,
 202         2.0475566387176513672e-01, +0,
 203         1.9790861144712756925e-01, +3.7894950972257700994e-16,
 204         2.0053362846374511719e-01, +0,
 205         1.9394752160084305359e-01, +2.8270367403478935534e-16,
 206         1.9641649723052978516e-01, +0,
 207         1.9007440763641536563e-01, -2.0842758095683676397e-16,
 208         1.9239699840545654297e-01, +0,
 209         1.8628369629742813629e-01, +3.4710917040399448932e-16,
 210         1.8846881389617919922e-01, +0,
 211         1.8256998712939509488e-01, +1.1053834120570125251e-16,
 212         1.8462586402893066406e-01, +0,
 213         1.7892875067284830237e-01, +3.0486232913366680305e-16,
 214         1.8086302280426025391e-01, +0,
 215         1.7535529778449010507e-01, -2.3810135019970148624e-16,
 216         1.7717504501342773438e-01, +0,
 217         1.7184559192514736736e-01, +5.1432582846210893916e-17,
 218         1.7355740070343017578e-01, +0,
 219         1.6839590847744290159e-01, +3.1605623296041433586e-18,
 220         1.7000591754913330078e-01, +0,
 221         1.6500283902547518977e-01, +1.5405422268770998251e-16,
 222         1.6651678085327148438e-01, +0,
 223         1.6166306303174859949e-01, +4.0042241517254928672e-16,
 224         1.6308629512786865234e-01, +0,
 225         1.5837358268281231943e-01, -2.2786616251622967291e-16,
 226         1.5971112251281738281e-01, +0,
 227         1.5513160990288810126e-01, -3.7547723514797166336e-16,
 228         1.5638816356658935547e-01, +0,
 229         1.5193468535499299321e-01, +4.3497510505554267446e-16,
 230         1.5311467647552490234e-01, +0,
 231         1.4878033155427861089e-01, -2.3102860235324261895e-16,
 232         1.4988791942596435547e-01, +0,
 233         1.4566628729590647140e-01, +9.9227592950040279415e-17,
 234         1.4670538902282714844e-01, +0,
 235         1.4259050967286590605e-01, -3.3869909683813096906e-18,
 236         1.4356482028961181641e-01, +0,
 237         1.3955105903633846509e-01, +1.5500435650773331566e-17,
 238         1.4046406745910644531e-01, +0,
 239         1.3654610022831903393e-01, +3.3965918616682805753e-16,
 240         1.3740110397338867188e-01, +0,
 241         1.3357402082462854764e-01, +2.7572431581527535421e-16,
 242         1.3437414169311523438e-01, +0,
 243         1.3063319828908959153e-01, -3.4667213797076707331e-16,
 244         1.3138139247894287109e-01, +0,
 245         1.2772200049776749609e-01, +3.1089261947725651968e-16,
 246         1.2842106819152832031e-01, +0,
 247         1.2436931430778752627e-01, -4.0654251891464630059e-16,
 248         1.2501454353332519531e-01, +0,
 249         1.2111683701666819957e-01, -3.9381654342464836012e-16,
 250         1.2171256542205810547e-01, +0,
 251         1.1844801833536511282e-01, -3.6673155595150283444e-16,
 252         1.1900508403778076172e-01, +0,
 253         1.1587365536613614125e-01, -1.5026628801318421951e-16,
 254         1.1639505624771118164e-01, +0,
 255         1.1338607085741525538e-01, +1.2886806274050538880e-16,
 256         1.1387449502944946289e-01, +0,
 257         1.1097844020819369604e-01, +2.3848343623577768044e-16,
 258         1.1143630743026733398e-01, +0,
```

```
259         1.0864456107308662069e-01, +4.2065430313285469408e-16,
260         1.0907405614852905273e-01, +0,
261         1.0637891628473727934e-01, -4.6883543790348472687e-18,
262         1.0678201913833618164e-01, +0,
263         1.0417650062205296990e-01, +1.4774925414624453292e-16,
264         1.0455501079559326172e-01, +0,
265         1.0203276464730581807e-01, -1.5677032794816452332e-16,
266         1.0238832235336303711e-01, +0,
267         9.9943617083734892503e-02, +3.4511310907979792828e-16,
268         1.0027772188186645508e-01, +0,
269         9.7905249824711049200e-02, +3.4489485563461708496e-16,
270         9.8219275474548339844e-02, +0,
271         9.5914316649349906641e-02, -1.3214510886789011569e-17,
272         9.6209526062011718750e-02, +0,
273         9.3967698614664918466e-02, +1.1048427091217964090e-16,
274         9.4245254993438720703e-02, +0,
275         9.2062564267554769515e-02, -3.7297463814697759309e-16,
276         9.2323541641235351562e-02, +0,
277         9.0196252506350660383e-02, -3.5280143043576718079e-16,
278         9.0441644191741943359e-02, +0,
279         8.8366391663268650802e-02, -6.1140673227541621183e-17,
280         8.8597118854522705078e-02, +0,
281         8.6570782100201526532e-02, -2.0998844594957629702e-16,
282         8.6787700653076171875e-02, +0,
283         8.4807337678923566671e-02, +3.9530981588194673068e-16,
284         8.5011243820190429688e-02, +0,
285         8.3074323040850828193e-02, -4.3022503210464894539e-17,
286         8.3265960216522216797e-02, +0,
287         8.1369880712663267275e-02, -6.3063867569127169744e-18,
288         8.1549942493438720703e-02, +0,
289         7.9692445771216036121e-02, -5.0787623072962671502e-17,
290         7.9861581325531005859e-02, +0,
291         7.8040568735575632786e-02, -3.8810063021216721741e-16,
292         7.8199386596679687500e-02, +0,
293         7.6412797391314235540e-02, +4.1246529500495762995e-16,
294         7.6561868190765380859e-02, +0,
295         7.4807854772808823896e-02, -3.7025599052186724156e-16,
296         7.4947714805603027344e-02, +0,
297         7.3224639528778112663e-02, +4.2209138483206712401e-17,
298         7.3355793952941894531e-02, +0,
299         7.1661929761571485642e-02, -3.2074473649855177622e-16,
300         7.1784853935241699219e-02, +0,
301         7.0118738881148168218e-02, -2.5371257235753296804e-16,
302         7.0233881473541259766e-02, +0,
303         6.8594137996416115755e-02, +3.3796987842548399135e-16,
304         6.8701922893524169922e-02, +0,
305         6.7087137393172291411e-02, +5.5061492696328852397e-17,
306         6.7187964916229248047e-02, +0,
307         6.5596983299946565182e-02, -2.1580863111502565280e-16,
308         6.5691232681274414062e-02, +0,
309         6.4122802037412718335e-02, -3.1315661827469233434e-16,
310         6.4210832118988037109e-02, +0,
311         6.2426231582525915087e-02, -2.5758980071296622188e-16,
312         6.2507450580596923828e-02, +0,
313         6.0781559928021700046e-02, +1.3736899336217710591e-16,
314         6.0856521129608154297e-02, +0,
315         5.9432882624005145544e-02, +2.2246097394328856474e-16,
316         5.9502959251403808594e-02, +0,
317         5.8132551274581167888e-02, -6.2525053236379489390e-18,
318         5.8198124170303344727e-02, +0,
319         5.6876611930681164608e-02, -2.6589930995607417149e-16,
320         5.6938022375106811523e-02, +0,
321         5.5661522654748551986e-02, -4.2736362859832186197e-16,
322         5.5719077587127685547e-02, +0,
323         5.4484124463757943602e-02, -1.6708067365310384253e-16,
324         5.4538100957870483398e-02, +0,
```



```
325         5.3341582449436764080e-02, +3.3271673004611311850e-17,
326         5.3392231464385986328e-02, +0,
327         5.2231267345892007370e-02, -3.5593396674200571616e-16,
328         5.2278816699981689453e-02, +0,
329         5.1150874758829623090e-02, +1.4432815841187114832e-16,
330         5.1195532083511352539e-02, +0,
331         5.0098306612679444072e-02, +9.4680943793589404083e-17,
332         5.0140261650085449219e-02, +0,
333         4.9071641675614507960e-02, +2.1131168520301896817e-16,
334         4.9111068248748779297e-02, +0,
335         4.8069135772851545596e-02, +1.6035336741307516296e-16,
336         4.8106193542480468750e-02, +0,
337         4.7089192241088539959e-02, -2.2491738698796901479e-16,
338         4.7124028205871582031e-02, +0,
339         4.6130362086062248750e-02, -1.5111423469578965206e-16,
340         4.6163111925125122070e-02, +0,
341         4.5191314382707403752e-02, +4.1989325207399786612e-16,
342         4.5222103595733642578e-02, +0,
343         4.4270836390474244126e-02, -4.1432635292331004454e-16,
344         4.4299781322479248047e-02, +0,
345         4.3367774164955186222e-02, -3.0615383054587355892e-16,
346         4.3394982814788818359e-02, +0,
347         4.2481121875321825598e-02, -3.6730166956273555173e-16,
348         4.2506694793701171875e-02, +0,
349         4.1609902899457651415e-02, -4.4226425958068821782e-16,
350         4.1633933782577514648e-02, +0,
351         4.0753259129372665370e-02, +1.9801161516527046872e-16,
352         4.0775835514068603516e-02, +0,
353         3.9910361780060910064e-02, +8.2560620036613164573e-18,
354         3.9931565523147583008e-02, +0,
355         3.9080441183869218946e-02, +3.9908991939242971628e-17,
356         3.9100348949432373047e-02, +0,
357         3.8262816593271686827e-02, +9.5182237812195590276e-17,
358         3.8281500339508056641e-02, +0,
359         3.7456806948784837630e-02, +1.5213508760679563439e-16,
360         3.7474334239959716797e-02, +0,
361         3.6661849947035918262e-02, +7.3335516005184616486e-17,
362         3.6678284406661987305e-02, +0,
363         3.5877353272533163420e-02, -1.3007348019891714540e-16,
364         3.5892754793167114258e-02, +0,
365         3.5102754135096780885e-02, -2.9903662298950558656e-16,
366         3.5117179155349731445e-02, +0,
367         3.4337638360670830195e-02, +2.9656295131966114331e-16,
368         3.4351140260696411133e-02, +0,
369         3.3581472523789734907e-02, +3.4810947205572817820e-16,
370         3.3594101667404174805e-02, +0,
371         3.2833871859357266487e-02, -3.8885440174405159838e-16,
372         3.2845675945281982422e-02, +0,
373         3.2094421679560447558e-02, +5.8805134853032009978e-17,
374         3.2105445861816406250e-02, +0,
375         3.1243584858944295490e-02, +2.8737383773884313066e-17,
376         3.1253755092620849609e-02, +0,
377         0, 0, 0, 0
378 };
```

_____***unchanged_portion_omitted_***

****
1345 Tue Nov 4 18:59:35 2014
new/usr/src/lib/libm/common/C/atan2pi.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak atan2pi = __atan2pi

/*
 * atan2pi(x) = atan2(x)/pi
 */

#include "libm.h"

static const double invpi = 0.31830988618379067153776752;

double
atan2pi(double y, double x) {
        int     ix, iy;

        if (x == 0.0 && y == 0.0) {
                ix = ((int *)&x)[HIWORD];
                iy = ((int *)&y)[HIWORD];
                if (ix >= 0)
                        return (y);
                return ((iy >= 0)? 1.0 : -1.0);
        }
        return (atan2(y, x) * invpi);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    2061 Tue Nov  4 18:59:35 2014
new/usr/src/lib/libm/common/C/atanh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanh = atanh
#pragma weak atanh = __atanh

/* INDENT OFF */
/*
 * atanh(x)
 * Code originated from 4.3bsd.
 * Modified by K.C. Ng for SUN 4.0 libm.
 * Method :
 *                  1                2x                          x
 *      atanh(x) = --- * log(1 + -------) = 0.5 * log1p(2 * --------)
 *                  2             1 - x                       1 - x
 * Note: to guarantee atanh(-x) = -atanh(x), we use
 *                 sign(x)              |x|
 *      atanh(x) = ------- * log1p(2*-------).
 *                    2               1 - |x|
 *
 * Special cases:
 *      atanh(x) is NaN if |x| > 1 with signal;
 *      atanh(NaN) is that NaN with no signal;
 *      atanh(+-1) is +-INF with signal.
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>

double
atanh(double x) {
        double t;
```



```
        if (isnan(x))
                return (x * x);         /* switched from x + x for Cheetah */
        t = fabs(x);
        if (t > 1.0)
                return (_SVID_libm_err(x, x, 30));      /* sNaN */
        if (t == 1.0)
                return (_SVID_libm_err(x, x, 31));      /* x/0; */
        t = t / (1.0 - t);
        return (copysign(0.5, x) * log1p(t + t));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1725 Tue Nov  4 18:59:35 2014
new/usr/src/lib/libm/common/C/ceil.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ceil = ceil
#pragma weak ceil = __ceil

/*
 * ceil(x) returns the least integral value bigger than or equal to x.
 * NOTE: ceil(x) returns result with the same sign as x's, including 0.
 *
 * Modified 8/4/04 for performance.
 */

#include "libm.h"

static const double
        zero = 0.0,
        one = 1.0,
        two52 = 4503599627370496.0;

double
ceil(double x) {
        double  t, w;
        int     hx, lx, ix;

        hx = ((int *)&x)[HIWORD];
        lx = ((int *)&x)[LOWORD];
        ix = hx & ~0x80000000;
        if (ix >= 0x43300000)   /* return x if |x| >= 2^52, or x is NaN */
                return (x * one);
        t = (hx >= 0)? two52 : -two52;
        w = x + t;
        t = w - t;
        if (ix < 0x3ff00000) {
                if ((ix | lx) == 0)
                        return (x);
```



```
                else
                        return ((hx < 0)? -zero : one);
        }
        return ((t >= x)? t : t + one);
}
______unchanged_portion_omitted_
```

****
1200 Tue Nov 4 18:59:36 2014
new/usr/src/lib/libm/common/C/copysign.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __copysign = copysign
#pragma weak copysign = __copysign

#include "libm.h"

double
copysign(double x, double y) {
        int hx, hy;

        hx = ((int *) &x)[HIWORD];
        hy = ((int *) &y)[HIWORD];
        return (hx ^ hy) >= 0 ?  (x) : (-x);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    6131 Tue Nov  4 18:59:36 2014
new/usr/src/lib/libm/common/C/cos.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cos = cos
#pragma weak cos = __cos

/* INDENT OFF */
/*
 * cos(x)
 * Accurate Table look-up algorithm by K.C. Ng, May, 1995.
 *
 * Algorithm: see sincos.c
 */

#include "libm.h"

static const double sc[] = {
/* ONE  = */  1.0,
/* NONE = */ -1.0,
/*
 * |sin(x) - (x+pp1*x^3+pp2*x^5)| <= 2^-58.79 for |x| < 0.008
 */
/* PP1  = */ -0.166666666666631655886725205237888952148062785868305556 7,
/* PP2  = */   .008333315652997472323564894248466758248475374977974017927,
/*
 * |(sin(x) - (x+p1*x^3+...+p4*x^9)|
 * |---------------------------- | <= 2^-57.63 for |x| < 0.1953125
 * |                x             |
 */
/* P1   = */ -1.6666666666666296698052151389203015896 56e-0001,
/* P2   = */  8.3333333323909512956839934552803363766 63e-0003,
/* P3   = */ -1.9841262379979766927915517782300984039 60e-0004,
/* P4   = */  2.7534036248542772376499876228483303511 10e-0006,
/*
 * |cos(x) - (1+qq1*x^2+qq2*x^4)| <= 2^-55.99 for |x| <= 0.008 (0x3f80624d)
 */
```



```
/* QQ1  = */ -0.4999999999975492381842911981948418542742729,
/* QQ2  = */  0.0416665429043520592945452091583576403987717 40,
/* Q1   = */ -0.5,
/* Q2   = */  4.1666666665003507036809455208607486174 45e-0002,
/* Q3   = */ -1.3888885964369722106942662905778486960 06e-0003,
/* Q4   = */  2.4785630788585894736795195178929534921 92e-0005,
/* PIO2_H    = */  1.570796326794896557999,
/* PIO2_L    = */  6.123233995736765886130e-17,
/* PIO2_L0   = */  6.123233995727922165564e-17,
/* PIO2_L1   = */  8.843720566135701120255e-29,
/* PI3O2_H   = */  4.712388980384689673997,
/* PI3O2_L   = */  1.836970198721029765839e-16,
/* PI3O2_L0  = */  1.836970198720396133587e-16,
/* PI3O2_L1  = */  6.336322524749201142226e-29,
/* PI5O2_H   = */  7.853981633974482789995,
/* PI5O2_L   = */  3.061616997868382943065e-16,
/* PI5O2_L0  = */  3.061616997861941598865e-16,
/* PI5O2_L1  = */  6.441344200433640781982e-28,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2452 Tue Nov  4 18:59:37 2014
new/usr/src/lib/libm/common/C/cosh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cosh = cosh
#pragma weak cosh = __cosh

/* INDENT OFF */
/*
 * cosh(x)
 * Code originated from 4.3bsd.
 * Modified by K.C. Ng for SUN 4.0 libm.
 * Method :
 *      1. Replace x by |x| (cosh(x) = cosh(-x)).
 *      2.
 *                                              [ exp(x) - 1 ]^2
 *  0        <= x <= 0.3465  :  cosh(x) := 1 + -------------------
 *                                                                 2*exp(x)
 *
 *                                                 exp(x) +  1/exp(x)
 *  0.3465   <= x <= 22      :  cosh(x) := -------------------
 *                                                                      2
 *  22       <= x <= lnovft  :  cosh(x) := exp(x)/2
 *  lnovft   <= x <  INF     :  cosh(x) := scalbn(exp(x-1024*ln2),1023)
 *
 *      Note: .3465 is a number near one half of ln2.
 *
 * Special cases:
 *      cosh(x) is |x| if x is +INF, -INF, or NaN.
 *      only cosh(0)=1 is exact for finite x.
 */
/* INDENT ON */

#include "libm.h"

static const double
        ln2 = 6.93147180559945286227e-01,
        ln2hi = 6.93147180369123816490e-01,
        ln2lo = 1.90821492927058770002e-10,
        lnovft = 7.09782712893383973096e+02;

double
cosh(double x) {
        double t, w;

        w = fabs(x);
        if (!finite(w))
                return (w * w);
        if (w < 0.3465) {
                t = expm1(w);
                w = 1.0 + t;
                if (w != 1.0)
                        w = 1.0 + (t * t) / (w + w);
                return (w);
        } else if (w < 22.0) {
                t = exp(w);
                return (0.5 * (t + 1.0 / t));
        } else if (w <= lnovft) {
                return (0.5 * exp(w));
        } else {
                w = (w - 1024 * ln2hi) - 1024 * ln2lo;
                if (w >= ln2)
                        return (_SVID_libm_err(x, x, 5));
                else
                        return (scalbn(exp(w), 1023));
        }
}
```


*_____unchanged_portion_omitted_*

```
**********************************************************
   13821 Tue Nov  4 18:59:37 2014
new/usr/src/lib/libm/common/C/erf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __erf = erf
#pragma weak __erfc = erfc
#pragma weak erf = __erf
#pragma weak erfc = __erfc

/* INDENT OFF */
/*
 * double erf(double x)
 * double erfc(double x)
 *                           x
 *                    2      |\
 *     erf(x)  =  ---------  | exp(-t*t)dt
 *                 sqrt(pi) \|
 *                           0
 *
 *     erfc(x) =  1-erf(x)
 *  Note that
 *              erf(-x) = -erf(x)
 *              erfc(-x) = 2 - erfc(x)
 *
 * Method:
 *      1. For |x| in [0, 0.84375]
 *          erf(x)  = x + x*R(x^2)
 *          erfc(x) = 1 - erf(x)           if x in [-.84375,0.25]
 *                  = 0.5 + ((0.5-x)-x*R)  if x in [0.25,0.84375]
 *         where R = P/Q where P is an odd poly of degree 8 and
 *         Q is an odd poly of degree 10.
 *                                                -57.90
 *                      | R - (erf(x)-x)/x | <= 2
 *
 *
 *         Remark. The formula is derived by noting
```



```
 *          erf(x) = (2/sqrt(pi))*(x - x^3/3 + x^5/10 - x^7/42 + ....)
 *         and that
 *          2/sqrt(pi) = 1.128379167095512573896158903121545171688
 *         is close to one. The interval is chosen because the fix
 *         point of erf(x) is near 0.6174 (i.e., erf(x)=x when x is
 *         near 0.6174), and by some experiment, 0.84375 is chosen to
 *         guarantee the error is less than one ulp for erf.
 *
 *      2. For |x| in [0.84375,1.25], let s = |x| - 1, and
 *         c = 0.84506291151 rounded to single (24 bits)
 *              erf(x)  = sign(x) * (c  + P1(s)/Q1(s))
 *              erfc(x) = (1-c)  - P1(s)/Q1(s) if x > 0
 *                        1+(c+P1(s)/Q1(s))    if x < 0
 *              |P1/Q1 - (erf(|x|)-c)| <= 2**-59.06
 *         Remark: here we use the taylor series expansion at x=1.
 *              erf(1+s) = erf(1) + s*Poly(s)
 *                       = 0.845.. + P1(s)/Q1(s)
 *         That is, we use rational approximation to approximate
 *                      erf(1+s) - (c = (single)0.84506291151)
 *         Note that |P1/Q1|< 0.078 for x in [0.84375,1.25]
 *         where
 *              P1(s) = degree 6 poly in s
 *              Q1(s) = degree 6 poly in s
 *
 *      3. For x in [1.25,1/0.35(~2.857143)],
 *              erfc(x) = (1/x)*exp(-x*x-0.5625+R1/S1)
 *              erf(x)  = 1 - erfc(x)
 *         where
 *              R1(z) = degree 7 poly in z, (z=1/x^2)
 *              S1(z) = degree 8 poly in z
 *
 *      4. For x in [1/0.35,28]
 *              erfc(x) = (1/x)*exp(-x*x-0.5625+R2/S2) if x > 0
 *                      = 2.0 - (1/x)*exp(-x*x-0.5625+R2/S2) if -6<x<0
 *                      = 2.0 - tiny            (if x <= -6)
 *              erf(x)  = sign(x)*(1.0 - erfc(x)) if x < 6, else
 *              erf(x)  = sign(x)*(1.0 - tiny)
 *         where
 *              R2(z) = degree 6 poly in z, (z=1/x^2)
 *              S2(z) = degree 7 poly in z
 *
 *      Note1:
 *         To compute exp(-x*x-0.5625+R/S), let s be a single
 *         precision number and s := x; then
 *              -x*x = -s*s + (s-x)*(s+x)
 *              exp(-x*x-0.5626+R/S) =
 *                      exp(-s*s-0.5625)*exp((s-x)*(s+x)+R/S);
 *      Note2:
 *         Here 4 and 5 make use of the asymptotic series
 *                        exp(-x*x)
 *              erfc(x) ~ ---------- * ( 1 + Poly(1/x^2) )
 *                        x*sqrt(pi)
 *         We use rational approximation to approximate
 *              g(s)=f(1/x^2) = log(erfc(x)*x) - x*x + 0.5625
 *         Here is the error bound for R1/S1 and R2/S2
 *              |R1/S1 - f(x)|  < 2**(-62.57)
 *              |R2/S2 - f(x)|  < 2**(-61.52)
 *
 *      5. For inf > x >= 28
 *              erf(x)  = sign(x) *(1 - tiny)  (raise inexact)
 *              erfc(x) = tiny*tiny (raise underflow) if x > 0
 *                      = 2 - tiny if x<0
 *
 *      7. Special case:
 *              erf(0)  = 0, erf(inf)  = 1, erf(-inf) = -1,
 *              erfc(0) = 1, erfc(inf) = 0, erfc(-inf) = 2,
```

```
 126  *      erfc/erf(NaN) is NaN
 127  */
 128 /* INDENT ON */

 130 #include "libm_synonyms.h"      /* __erf, __erfc, __exp */
 130 #include "libm_macros.h"
 131 #include <math.h>

 133 static const double xxx[] = {
 134 /* tiny */      1e-300,
 135 /* half */      5.00000000000000000000e-01,     /* 3FE00000, 00000000 */
 136 /* one */       1.00000000000000000000e+00,     /* 3FF00000, 00000000 */
 137 /* two */       2.00000000000000000000e+00,     /* 40000000, 00000000 */
 138 /* erx */       8.45062911510467529297e-01,     /* 3FEB0AC1, 60000000 */
 139 /*
 140  * Coefficients for approximation to  erf on [0,0.84375]
 141  */
 142 /* efx */        1.28379167095512586316e-01,    /* 3FC06EBA, 8214DB69 */
 143 /* efx8 */       1.02703333676410069053e+00,    /* 3FF06EBA, 8214DB69 */
 144 /* pp0 */        1.28379167095512558561e-01,    /* 3FC06EBA, 8214DB68 */
 145 /* pp1 */       -3.25042107247001499370e-01,    /* BFD4CD7D, 691CB913 */
 146 /* pp2 */       -2.84817495755985104766e-02,    /* BF9D2A51, DBD7194F */
 147 /* pp3 */       -5.77027029648944159157e-03,    /* BF77A291, 236668E4 */
 148 /* pp4 */       -2.37630166566501626084e-05,    /* BEF8EAD6, 120016AC */
 149 /* qq1 */        3.97917223959155352819e-01,    /* 3FD97779, CDDADC09 */
 150 /* qq2 */        6.50222499887672944485e-02,    /* 3FB0A54C, 5536CEBA */
 151 /* qq3 */        5.08130628187576562776e-03,    /* 3F74D022, C4D36B0F */
 152 /* qq4 */        1.32494738004321644526e-04,    /* 3F215DC9, 221C1A10 */
 153 /* qq5 */       -3.96022827877536812320e-06,    /* BED09C43, 42A26120 */
 154 /*
 155  * Coefficients for approximation to  erf  in [0.84375,1.25]
 156  */
 157 /* pa0 */       -2.36211856075265944077e-03,    /* BF6359B8, BEF77538 */
 158 /* pa1 */        4.14856118683748331666e-01,    /* 3FDA8D00, AD92B34D */
 159 /* pa2 */       -3.72207876035701323847e-01,    /* BFD7D240, FBB8C3F1 */
 160 /* pa3 */        3.18346619901161753674e-01,    /* 3FD45FCA, 805120E4 */
 161 /* pa4 */       -1.10894694282396677476e-01,    /* BFBC6398, 3D3E28EC */
 162 /* pa5 */        3.54783043256182359371e-02,    /* 3FA22A36, 599795EB */
 163 /* pa6 */       -2.16637559486879084300e-03,    /* BF61BF38, 0A96073F */
 164 /* qa1 */        1.06420880400844228286e-01,    /* 3FBB3E66, 18EEE323 */
 165 /* qa2 */        5.40397917702171048937e-01,    /* 3FE14AF0, 92EB6F33 */
 166 /* qa3 */        7.18286544141962662868e-02,    /* 3FB2635C, D99FE9A7 */
 167 /* qa4 */        1.26171219808761642112e-01,    /* 3FC02660, E763351F */
 168 /* qa5 */        1.36370839120290507362e-02,    /* 3F8BEDC2, 6B51DD1C */
 169 /* qa6 */        1.19844998467991074170e-02,    /* 3F888B54, 5735151D */
 170 /*
 171  * Coefficients for approximation to  erfc in [1.25,1/0.35]
 172  */
 173 /* ra0 */       -9.86494403484714822705e-03,    /* BF843412, 600D6435 */
 174 /* ra1 */       -6.93858572707181764372e-01,    /* BFE63416, E4BA7360 */
 175 /* ra2 */       -1.05586262253232909814e+01,    /* C0251E04, 41B0E726 */
 176 /* ra3 */       -6.23753324503260060396e+01,    /* C04F300A, E4CBA38D */
 177 /* ra4 */       -1.62396669462573470355e+02,    /* C0644CB1, 84282266 */
 178 /* ra5 */       -1.84605092906711035994e+02,    /* C067135C, EBCCABB2 */
 179 /* ra6 */       -8.12874355063065934246e+01,    /* C0545265, 57E4D2F2 */
 180 /* ra7 */       -9.81432934416914548592e+00,    /* C023A0EF, C69AC25C */
 181 /* sa1 */        1.96512716674392571292e+01,    /* 4033A6B9, BD707687 */
 182 /* sa2 */        1.37657754143519042600e+02,    /* 4061350C, 526AE721 */
 183 /* sa3 */        4.34565877475229228821e+02,    /* 407B290D, D58A1A71 */
 184 /* sa4 */        6.45387271733267880336e+02,    /* 40842B19, 21EC2868 */
 185 /* sa5 */        4.29008140027567833386e+02,    /* 407AD021, 57700314 */
 186 /* sa6 */        1.08635005541779435134e+02,    /* 405B28A3, EE48AE2C */
 187 /* sa7 */        6.57024977031928170135e+00,    /* 401A47EF, 8E484A93 */
 188 /* sa8 */       -6.04244152148580987438e-02,    /* BFAEEFF2, EE749A62 */
 189 /*
 190  * Coefficients for approximation to  erfc in [1/.35,28]
```



```
 191  */
 192 /* rb0 */       -9.86494292470009928597e-03,    /* BF843412, 39E86F4A */
 193 /* rb1 */       -7.99283237680523006574e-01,    /* BFE993BA, 70C285DE */
 194 /* rb2 */       -1.77579549177547519889e+01,    /* C031C209, 555F995A */
 195 /* rb3 */       -1.60636384855821916062e+02,    /* C064145D, 43C5ED98 */
 196 /* rb4 */       -6.37566443368389627722e+02,    /* C083EC88, 1375F228 */
 197 /* rb5 */       -1.02509513161107724954e+03,    /* C0900461, 6A2E5992 */
 198 /* rb6 */       -4.83519191608651397019e+02,    /* C07E384E, 9BDC383F */
 199 /* sb1 */        3.03380607434824582924e+01,    /* 403E568B, 261D5190 */
 200 /* sb2 */        3.25792512996573918826e+02,    /* 40745CAE, 221B9F0A */
 201 /* sb3 */        1.53672958608443695994e+03,    /* 409802EB, 189D5118 */
 202 /* sb4 */        3.19985821950859553908e+03,    /* 40A8FFB7, 688C246A */
 203 /* sb5 */        2.55305040643316442583e+03,    /* 40A3F219, CEDF3BE6 */
 204 /* sb6 */        4.74528541206955367215e+02,    /* 407DA874, E79FE763 */
 205 /* sb7 */       -2.24409524465858183362e+01     /* C03670E2, 42712D62 */
 206 };
_____unchanged_portion_omitted_
```

```
**********************************************************
   14865 Tue Nov  4 18:59:38 2014
new/usr/src/lib/libm/common/C/exp.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __exp = exp
#pragma weak exp = __exp

/*
 * exp(x)
 * Hybrid algorithm of Peter Tang's Table driven method (for large
 * arguments) and an accurate table (for small arguments).
 * Written by K.C. Ng, November 1988.
 * Method (large arguments):
 *      1. Argument Reduction: given the input x, find r and integer k
 *         and j such that
 *                    x = (k+j/32)*(ln2) + r,  |r| <= (1/64)*ln2
 *
 *      2. exp(x) = 2^k * (2^(j/32) + 2^(j/32)*expm1(r))
 *         a. expm1(r) is approximated by a polynomial:
 *            expm1(r) ~ r + t1*r^2 + t2*r^3 + ... + t5*r^6
 *            Here t1 = 1/2 exactly.
 *         b. 2^(j/32) is represented to twice double precision
 *            as TBL[2j]+TBL[2j+1].
 *
 * Note: If divide were fast enough, we could use another approximation
 *       in 2.a:
 *            expm1(r) ~ (2r)/(2-R), R = r - r^2*(t1 + t2*r^2)
 *            (for the same t1 and t2 as above)
 *
 * Special cases:
 *      exp(INF) is INF, exp(NaN) is NaN;
 *      exp(-INF)=  0;
 *      for finite argument, only exp(0)=1 is exact.
 *
 * Accuracy:
 *      According to an error analysis, the error is always less than
 *      an ulp (unit in the last place).  The largest errors observed
```



```
 *      are less than 0.55 ulp for normal results and less than 0.75 ulp
 *      for subnormal results.
 *
 * Misc. info.
 *      For IEEE double
 *              if x >  7.09782712893383973096e+02 then exp(x) overflow
 *              if x < -7.45133219101941108420e+02 then exp(x) underflow
 */

#include "libm.h"

static const double TBL[] = {
        1.00000000000000000000e+00,  0.00000000000000000000e+00,
        1.02189714865411662714e+00,  5.10922502897344389359e-17,
        1.04427378242741375480e+00,  8.55188970553796365958e-17,
        1.06714040067682369717e+00, -7.89985396684158212226e-17,
        1.09050773266525768967e+00, -3.04678207981247114697e-17,
        1.11438674259589243221e+00,  1.04102784568455709549e-16,
        1.13878863475669156458e+00,  8.91281267602540777782e-17,
        1.16372485877757747552e+00,  3.82920483692409349872e-17,
        1.18920711500272102690e+00,  3.98201523146564611098e-17,
        1.21524735998046895524e+00, -7.71263069268148813091e-17,
        1.24185781207348400201e+00,  4.65802759183693679123e-17,
        1.26905095719173321989e+00,  2.66793213134218609523e-18,
        1.29683955465100964055e+00,  2.53825027948883149593e-17,
        1.32523664315974132322e+00, -2.85873121003886075697e-17,
        1.35425554693689265129e+00,  7.70094837980298946162e-17,
        1.38390988196383202258e+00, -6.77051165879478628716e-17,
        1.41421356237309514547e+00, -9.66729331345291345105e-17,
        1.44518080697704665027e+00, -3.02375813499398731940e-17,
        1.47682614593949934623e+00, -3.48399455689279579579e-17,
        1.50916442759342284141e+00, -1.01645532775429503911e-16,
        1.54221082540794074411e+00,  7.94983480969762085616e-17,
        1.57598084510788649659e+00, -1.01369164712783039808e-17,
        1.61049033194925428347e+00,  2.47071925697978878522e-17,
        1.64575547815396494578e+00, -1.01256799136747726038e-16,
        1.68179283050742900407e+00,  8.19901002058149652013e-17,
        1.71861929812247793414e+00, -1.85138041826311098821e-17,
        1.75625216037329945351e+00,  2.96014069544887330703e-17,
        1.79470907500310716820e+00,  1.82274584279120867698e-17,
        1.83400808640934243066e+00,  3.28310722424562658722e-17,
        1.87416763411029996256e+00, -6.12276341300414256164e-17,
        1.91520656139714740007e+00, -1.06199460561959626376e-16,
        1.95714412417540017941e+00,  8.96076779103666776760e-17,
};
______unchanged_portion_omitted_
```

****
2581 Tue Nov 4 18:59:38 2014
new/usr/src/lib/libm/common/C/exp10.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak exp10 = __exp10

/* INDENT OFF */
/*
 * exp10(x)
 * Code by K.C. Ng for SUN 4.0 libm.
 * Method :
 *      n = nint(x*(log10/log2));
 *      exp10(x) = 10**x = exp(x*ln(10)) = exp(n*ln2+(x*ln10-n*ln2))
 *               = 2**n*exp(ln10*(x-n*log2/log10)))
 *      If x is an integer < 23 then use repeat multiplication. For
 *      10**22 is the largest representable integer.
 */
/* INDENT ON */

#include "libm.h"

static const double C[] = {
        3.3219280948736234787,  /* log(10)/log(2) */
        2.3025850929940456840,  /* log(10) */
        3.0102999565860955045E-1,       /* log(2)/log(10) high */
        5.3716447674669983622E-12,      /* log(2)/log(10) low */
        0.0,
        0.5,
        1.0,
        10.0,
        1.0e300,
        1.0e-300,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2067 Tue Nov  4 18:59:39 2014
new/usr/src/lib/libm/common/C/exp2.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29 #pragma weak __exp2 = exp2
  29 #pragma weak exp2 = __exp2

  31 /* INDENT OFF */
  32 /*
  33  * exp2(x)
  34  * Code by K.C. Ng for SUN 4.0 libm.
  35  * Method :
  36  *      exp2(x) = 2**x = 2**((x-anint(x))+anint(x))
  37  *              = 2**anint(x)*2**(x-anint(x))
  38  *              = 2**anint(x)*exp((x-anint(x))*ln2)
  39  */
  40 /* INDENT ON */

  42 #include "libm.h"

  44 static const double C[] = {
  45         0.0,
  46         1.0,
  47         0.5,
  48         6.931471805599452862227e-01,
  49         1.0e300,
  50         1.0e-300,
  51 };
_____unchanged_portion_omitted_
```

```
**********************************************************
    8478 Tue Nov  4 18:59:39 2014
new/usr/src/lib/libm/common/C/expm1.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __expm1 = expm1
#pragma weak expm1 = __expm1

/* INDENT OFF */
/*
 * expm1(x)
 * Returns exp(x)-1, the exponential of x minus 1.
 *
 * Method
 *   1. Arugment reduction:
 *      Given x, find r and integer k such that
 *
 *               x = k*ln2 + r,  |r| <= 0.5*ln2 ~ 0.34658
 *
 *      Here a correction term c will be computed to compensate
 *      the error in r when rounded to a floating-point number.
 *
 *   2. Approximating expm1(r) by a special rational function on
 *      the interval [0,0.34658]:
 *      Since
 *          r*(exp(r)+1)/(exp(r)-1) = 2+ r^2/6 - r^4/360 + ...
 *      we define R1(r*r) by
 *          r*(exp(r)+1)/(exp(r)-1) = 2+ r^2/6 * R1(r*r)
 *      That is,
 *          R1(r**2) = 6/r *((exp(r)+1)/(exp(r)-1) - 2/r)
 *                   = 6/r * ( 1 + 2.0*(1/(exp(r)-1) - 1/r))
 *                   = 1 - r^2/60 + r^4/2520 - r^6/100800 + ...
 *      We use a special Reme algorithm on [0,0.347] to generate
 *      a polynomial of degree 5 in r*r to approximate R1. The
 *      maximum error of this polynomial approximation is bounded
 *      by 2**-61. In other words,
 *          R1(z) ~ 1.0 + Q1*z + Q2*z**2 + Q3*z**3 + Q4*z**4 + Q5*z**5
```



```
 *      where   Q1  =  -1.6666666666666567384E-2,
 *              Q2  =   3.9682539681370365873E-4,
 *              Q3  =  -9.9206344733435987357E-6,
 *              Q4  =   2.5051361420808517002E-7,
 *              Q5  =  -6.2843505682382617102E-9;
 *      (where z=r*r, and the values of Q1 to Q5 are listed below)
 *      with error bounded by
 *          |                    5           |     -61
 *          | 1.0+Q1*z+...+Q5*z   -  R1(z) | <= 2
 *          |                                |
 *
 *      expm1(r) = exp(r)-1 is then computed by the following
 *      specific way which minimize the accumulation rounding error:
 *                             2     3
 *                            r     r    [ 3 - (R1 + R1*r/2)  ]
 *            expm1(r) = r + --- + --- * [--------------------]
 *                            2     2    [ 6 - r*(3 - R1*r/2) ]
 *
 *      To compensate the error in the argument reduction, we use
 *              expm1(r+c) = expm1(r) + c + expm1(r)*c
 *                         ~ expm1(r) + c + r*c
 *      Thus c+r*c will be added in as the correction terms for
 *      expm1(r+c). Now rearrange the term to avoid optimization
 *      screw up:
 *                         (      2                                    2 )
 *                         ({  ( r    [ R1 -  (3 - R1*r/2) ]  )  }    r  )
 *       expm1(r+c)~r - ({r*(--- * [--------------------]-c)-c} - --- )
 *                         ({  ( 2    [ 6 - r*(3 - R1*r/2) ]  )  }    2  )
 *                         (                                          )
 *
 *                   = r - E
 *   3. Scale back to obtain expm1(x):
 *      From step 1, we have
 *         expm1(x) = either 2^k*[expm1(r)+1] - 1
 *                  = or     2^k*[expm1(r) + (1-2^-k)]
 *   4. Implementation notes:
 *      (A). To save one multiplication, we scale the coefficient Qi
 *           to Qi*2^i, and replace z by (x^2)/2.
 *      (B). To achieve maximum accuracy, we compute expm1(x) by
 *        (i)   if x < -56*ln2, return -1.0, (raise inexact if x != inf)
 *        (ii)  if k=0, return r-E
 *        (iii) if k=-1, return 0.5*(r-E)-0.5
 *        (iv)  if k=1 if r < -0.25, return 2*((r+0.5)- E)
 *                                       else         return  1.0+2.0*(r-E);
 *        (v)   if (k<-2||k>56) return 2^k(1-(E-r)) - 1 (or exp(x)-1)
 *        (vi)  if k <= 20, return 2^k((1-2^-k)-(E-r)), else
 *        (vii) return 2^k(1-((E+2^-k)-r))
 *
 * Special cases:
 *      expm1(INF) is INF, expm1(NaN) is NaN;
 *      expm1(-INF) is -1, and
 *      for finite argument, only expm1(0)=0 is exact.
 *
 * Accuracy:
 *      according to an error analysis, the error is always less than
 *      1 ulp (unit in the last place).
 *
 * Misc. info.
 *      For IEEE double
 *          if x >  7.09782712893383973096e+02 then expm1(x) overflow
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following
 * constants. The decimal values may be used, provided that the
 * compiler will convert from decimal to binary accurately enough
 * to produce the hexadecimal values shown.
```

```
 127  */
 128 /* INDENT ON */

 130 #include "libm_synonyms.h"      /* __expm1 */
 130 #include "libm_macros.h"
 131 #include <math.h>

 133 static const double xxx[] = {
 134 /* one */               1.0,
 135 /* huge */              1.0e+300,
 136 /* tiny */              1.0e-300,
 137 /* o_threshold */       7.09782712893383973096e+02,    /* 40862E42 FEFA39EF */
 138 /* ln2_hi */            6.93147180369123816490e-01,    /* 3FE62E42 FEE00000 */
 139 /* ln2_lo */            1.90821492927058770002e-10,    /* 3DEA39EF 35793C76 */
 140 /* invln2 */            1.44269504088896338700e+00,    /* 3FF71547 652B82FE */
 141 /* scaled coefficients related to expm1 */
 142 /* Q1 */               -3.33333333333331316428e-02,    /* BFA11111 111110F4 */
 143 /* Q2 */                1.58730158725481460165e-03,    /* 3F5A01A0 19FE5585 */
 144 /* Q3 */               -7.93650757867487942473e-05,    /* BF14CE19 9EAADBB7 */
 145 /* Q4 */                4.00821782732936239552e-06,    /* 3ED0CFCA 86E65239 */
 146 /* Q5 */               -2.01099218183624371326e-07     /* BE8AFDB7 6E09C32D */
 147 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1178 Tue Nov  4 18:59:40 2014
new/usr/src/lib/libm/common/C/fabs.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fabs = fabs
#pragma weak fabs = __fabs

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_macros.h"
#include <math.h>

double
fabs(double x) {
        int *px = (int *) &x;

        px[HIWORD] &= ~0x80000000;
        return (x);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    1730 Tue Nov  4 18:59:40 2014
new/usr/src/lib/libm/common/C/floor.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __floor = floor
#pragma weak floor = __floor

/*
 * floor(x) returns the biggest integral value less than or equal to x.
 * NOTE: floor(x) returns result with the same sign as x's, including 0.
 *
 * Modified 8/4/04 for performance.
 */

#include "libm.h"

static const double
        zero = 0.0,
        one = 1.0,
        two52 = 4503599627370496.0;

double
floor(double x) {
        double  t, w;
        int     hx, lx, ix;

        hx = ((int *)&x)[HIWORD];
        lx = ((int *)&x)[LOWORD];
        ix = hx & ~0x80000000;
        if (ix >= 0x43300000)   /* return x if |x| >= 2^52, or x is NaN */
                return (x * one);
        t = (hx >= 0)? two52 : -two52;
        w = x + t;
        t = w - t;
        if (ix < 0x3ff00000) {
                if ((ix | lx) == 0)
                        return (x);
```



```
                else
                        return ((hx < 0)? -one : zero);
        }
        return ((t <= x)? t : t - one);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    3207 Tue Nov  4 18:59:41 2014
new/usr/src/lib/libm/common/C/fmod.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fmod = fmod
#pragma weak fmod = __fmod

#include "libm.h"

static const double zero = 0.0;

/*
 * The following implementation assumes fast 64-bit integer arith-
 * metic.  This is fine for sparc because we build libm in v8plus
 * mode.  It's also fine for sparcv9 and amd64, although we have
 * assembly code on amd64.  For x86, it would be better to use
 * 32-bit code, but we have assembly for x86, too.
 */
double
fmod(double x, double y) {
        double          w;
        long long       hx, ix, iy, iz;
        int             nd, k, ny;

        hx = *(long long *)&x;
        ix = hx & ~0x8000000000000000ull;
        iy = *(long long *)&y & ~0x8000000000000000ull;

        /* handle special cases */
        if (iy == 0ll)
                return (_SVID_libm_err(x, y, 27));

        if (ix >= 0x7ff0000000000000ll || iy > 0x7ff0000000000000ll)
                return ((x * y) * zero);

        if (ix <= iy)
                return ((ix < iy)? x : x * zero);
```



```
        /*
         * Set:
         *      ny = true exponent of y
         *      nd = true exponent of x minus true exponent of y
         *      ix = normalized significand of x
         *      iy = normalized significand of y
         */
        ny = iy >> 52;
        k = ix >> 52;
        if (ny == 0) {
                /* y is subnormal, x could be normal or subnormal */
                ny = 1;
                while (iy < 0x0010000000000000ll) {
                        ny -= 1;
                        iy += iy;
                }
                nd = k - ny;
                if (k == 0) {
                        nd += 1;
                        while (ix < 0x0010000000000000ll) {
                                nd -= 1;
                                ix += ix;
                        }
                } else {
                        ix = 0x0010000000000000ll | (ix & 0x000fffffffffffll);
                }
        } else {
                /* both x and y are normal */
                nd = k - ny;
                ix = 0x0010000000000000ll | (ix & 0x000fffffffffffll);
                iy = 0x0010000000000000ll | (iy & 0x000fffffffffffll);
        }

        /* perform fixed point mod */
        while (nd--) {
                iz = ix - iy;
                if (iz >= 0)
                        ix = iz;
                ix += ix;
        }
        iz = ix - iy;
        if (iz >= 0)
                ix = iz;

        /* convert back to floating point and restore the sign */
        if (ix == 0ll)
                return (x * zero);
        while (ix < 0x0010000000000000ll) {
                ix += ix;
                ny -= 1;
        }
        while (ix > 0x0020000000000000ll) {     /* XXX can this ever happen? */
                ny += 1;
                ix >>= 1;
        }
        if (ny <= 0) {
                /* result is subnormal */
                k = -ny + 1;
                ix >>= k;
                *(long long *)&w = (hx & 0x8000000000000000ull) | ix;
                return (w);
        }
        *(long long *)&w = (hx & 0x8000000000000000ull) |
            ((long long)ny << 52) | (ix & 0x000fffffffffffll);
        return (w);
```

```
 127 }
______unchanged_portion_omitted_
```

****
1333 Tue Nov 4 18:59:41 2014
new/usr/src/lib/libm/common/C/gamma.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29 #pragma weak __gamma = gamma
29 #pragma weak gamma = __gamma

31 #include "libm.h"

33 extern int signgam;

35 double
36 gamma(double x) {
37         double  g;

39         if (!finite(x))
40                 return (x * x);

42         g = rint(x);
43         if (x == g && x <= 0.0) {
44                 signgam = 1;
45                 return (_SVID_libm_err(x, x, 41));
46         }

48         g = __k_lgamma(x, &signgam);
49         if (!finite(g))
50             g = _SVID_libm_err(x, x, 40);
51         return (g);
52 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1126 Tue Nov  4 18:59:41 2014
new/usr/src/lib/libm/common/C/gamma_r.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __gamma_r = gamma_r
#pragma weak gamma_r = __gamma_r

#include "libm.h"

double
gamma_r(double x, int *signgamp) {
        return (lgamma_r(x, signgamp));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5483 Tue Nov  4 18:59:42 2014
new/usr/src/lib/libm/common/C/hypot.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __hypot = hypot
#pragma weak hypot = __hypot

/* INDENT OFF */
/*
 * Hypot(x, y)
 * by K.C. Ng for SUN 4.0 libm, updated 3/11/2003.
 * Method :
 * A. When rounding is rounded-to-nearest:
 *      If z = x * x + y * y has error less than sqrt(2) / 2 ulp than
 *      sqrt(z) has error less than 1 ulp.
 *      So, compute sqrt(x*x+y*y) with some care as follows:
 *      Assume x > y > 0;
 *      1. Check whether save and set rounding to round-to-nearest
 *      2. if x > 2y  use
 *              xh*xh+(y*y+((x-xh)*(x+xh))) for x*x+y*y
 *      where xh = x with lower 32 bits cleared;  else
 *      3. if x <= 2y use
 *              x2h*yh+((x-y)*(x-y)+(x2h*(y-yh)+(x2-x2h)*y))
 *      where x2 = 2*x, x2h = 2x with lower 32 bits cleared, yh = y with
 *      lower 32 bits chopped.
 *
 * B. When rounding is not rounded-to-nearest:
 *      The following (magic) formula will yield an error less than 1 ulp.
 *      z = sqrt(x * x + y * y)
 *              hypot(x, y) = x + (y / ((x + z) / y))
 *
 * NOTE: DO NOT remove parenthsis!
 *
 * Special cases:
 *      hypot(x, y) is INF if x or y is +INF or -INF; else
 *      hypot(x, y) is NAN if x or y is NAN.
 *
 * Accuracy:
 *      hypot(x, y) returns sqrt(x^2+y^2) with error less than 1 ulps
 *      (units in the last place)
 */

#include "libm.h"

static const double
        zero = 0.0,
        onep1u = 1.00000000000000022204e+00,    /* 0x3ff00000 1 = 1+2**-52 */
        twom53 = 1.11022302462515654042e-16,    /* 0x3ca00000 0 = 2**-53 */
        twom768 = 6.44114876959713308e-232,     /* 2^-768 */
        two768  = 1.55251809230070893e+231;     /* 2^768 */

/* INDENT ON */

double
hypot(double x, double y) {
        double xh, yh, w, ax, ay;
        int i, j, nx, ny, ix, iy, iscale = 0;
        unsigned lx, ly;

        ix = ((int *) &x)[HIWORD] & ~0x80000000;
        lx = ((int *) &x)[LOWORD];
        iy = ((int *) &y)[HIWORD] & ~0x80000000;
        ly = ((int *) &y)[LOWORD];
/*
 * Force ax = |x| ~>~ ay = |y|
 */
        if (iy > ix) {
                ax = fabs(y);
                ay = fabs(x);
                i = ix;
                ix = iy;
                iy = i;
                i = lx;
                lx = ly;
                ly = i;
        } else {
                ax = fabs(x);
                ay = fabs(y);
        }
        nx = ix >> 20;
        ny = iy >> 20;
        j  = nx - ny;
/*
 * x >= 2^500 (x*x or y*y may overflow)
 */
        if (nx >= 0x5f3) {
                if (nx == 0x7ff) {      /* inf or NaN, signal of sNaN */
                        if (((ix - 0x7ff00000) | lx) == 0)
                                return (ax == ay ? ay : ax);
                        else if (((iy - 0x7ff00000) | ly) == 0)
                                return (ay == ax ? ax : ay);
                        else
                                return (ax * ay);       /* + -> * for Cheetah */
                } else if (j > 32) {    /* x >> y */
                        if (j <= 53)
                                ay *= twom53;
                        ax += ay;
                        if (((int *) &ax)[HIWORD] == 0x7ff00000)
                                ax = _SVID_libm_err(x, y, 4);
                        return (ax);
                }
                ax *= twom768;
```

```
127                 ay *= twom768;
128                 iscale = 2;
129                 ix -= 768 << 20;
130                 iy -= 768 << 20;
131         }
132 /*
133  * y < 2^-450 (x*x or y*y may underflow)
134  */
135         else if (ny < 0x23d) {
136                 if ((ix | lx) == 0)
137                         return (ay);
138                 if ((iy | ly) == 0)
139                         return (ax);
140                 if (j > 53)             /* x >> y */
141                         return (ax + ay);
142                 iscale = 1;
143                 ax *= two768;
144                 ay *= two768;
145                 if (nx == 0) {
146                         if (ax == zero) /* guard subnormal flush to zero */
147                                 return (ax);
148                         ix = ((int *) &ax)[HIWORD];
149                 } else
150                         ix += 768 << 20;
151                 if (ny == 0) {
152                         if (ay == zero) /* guard subnormal flush to zero */
153                                 return (ax * twom768);
154                         iy = ((int *) &ay)[HIWORD];
155                 } else
156                         iy += 768 << 20;
157                 j = (ix >> 20) - (iy >> 20);
158                 if (j > 32) {           /* x >> y */
159                         if (j <= 53)
160                                 ay *= twom53;
161                         return ((ax + ay) * twom768);
162                 }
163         } else if (j > 32) {            /* x >> y */
164                 if (j <= 53)
165                         ay *= twom53;
166                 return (ax + ay);
167         }
168 /*
169  * Medium range ax and ay with max{|ax/ay|,|ay/ax|} bounded by 2^32
170  * First check rounding mode by comparing onep1u*onep1u with onep1u+twom53.
171  * Make sure the computation is done at run-time.
172  */
173         if (((lx | ly) << 5) == 0) {
174                 ay = ay * ay;
175                 ax += ay / (ax + sqrt(ax * ax + ay));
176         } else
177         if (onep1u * onep1u != onep1u + twom53) {
178         /* round-to-zero, positive, negative mode */
179         /* magic formula with less than an ulp error */
180                 w = sqrt(ax * ax + ay * ay);
181                 ax += ay / ((ax + w) / ay);
182         } else {
183         /* round-to-nearest mode */
184                 w = ax - ay;
185                 if (w > ay) {
186                         ((int *) &xh)[HIWORD] = ix;
187                         ((int *) &xh)[LOWORD] = 0;
188                         ay = ay * ay + (ax - xh) * (ax + xh);
189                         ax = sqrt(xh * xh + ay);
190                 } else {
191                         ax = ax + ax;
192                         ((int *) &xh)[HIWORD] = ix + 0x00100000;
```



```
193                         ((int *) &xh)[LOWORD] = 0;
194                         ((int *) &yh)[HIWORD] = iy;
195                         ((int *) &yh)[LOWORD] = 0;
196                         ay = w * w + ((ax - xh) * yh + (ay - yh) * ax);
197                         ax = sqrt(xh * yh + ay);
198                 }
199         }
200         if (iscale > 0) {
201                 if (iscale == 1)
202                         ax *= twom768;
203                 else {
204                         ax *= two768;   /* must generate side effect here */
205                         if (((int *) &ax)[HIWORD] == 0x7ff00000)
206                                 ax = _SVID_libm_err(x, y, 4);
207                 }
208         }
209         return (ax);
210 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2226 Tue Nov  4 18:59:42 2014
new/usr/src/lib/libm/common/C/ilogb.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ilogb = ilogb
#pragma weak ilogb = __ilogb

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */

#if defined(__x86)
static const double two52 = 4503599627370496.0;
#else
/*
 * v: high part of a non-zero subnormal |x|; w: low part of |x|
 */
static int
ilogb_subnormal(unsigned v, unsigned w) {
        int r = -1022 - 52;

        if (v)
                r += 32;
        else
                v = w;
        if (v & 0xffff0000)
                r += 16, v >>= 16;
        if (v & 0xff00)
                r += 8, v >>= 8;
        if (v & 0xf0)
                r += 4, v >>= 4;
        v <<= 1;
        return (r + ((0xffffaa50 >> v) & 0x3));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    8797 Tue Nov  4 18:59:43 2014
new/usr/src/lib/libm/common/C/j0.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Floating point Bessel's function of the first and second kinds
 * of order zero: j0(x),y0(x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 */

#pragma weak __j0 = j0
#pragma weak __y0 = y0
#pragma weak j0 = __j0
#pragma weak y0 = __y0

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>
#include <values.h>

#define GENERIC double
static const GENERIC
zero    = 0.0,
small   = 1.0e-5,
tiny    = 1.0e-18,
one     = 1.0,
eight   = 8.0,
invsqrtpi = 5.64189583547756286948079451560772585844le-0001,
tpi     = 0.63661977236758134307553505349005744 8;

static GENERIC pzero(GENERIC), qzero(GENERIC);
static const GENERIC r0[4] = {  /* [1.e-5, 1.28] */
```



```
        -2.50000000000000362213188089483047675553 7e-0001,
        1.09559754733483026323443385593237535330 3e-0002,
        -1.81973475046332092179918725898709808769 7e-0004,
        9.97700194680613165754421250106989393084 6e-0007,
};
______unchanged_portion_omitted_
```

```
**********************************************************
    9167 Tue Nov  4 18:59:43 2014
new/usr/src/lib/libm/common/C/j1.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * floating point Bessel's function of the first and second kinds
 * of order zero: j1(x),y1(x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 */

#pragma weak __j1 = j1
#pragma weak __y1 = y1
#pragma weak j1 = __j1
#pragma weak y1 = __y1

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>
#include <values.h>

#define GENERIC double
static const GENERIC
zero    = 0.0,
small   = 1.0e-5,
tiny    = 1.0e-20,
one     = 1.0,
invsqrtpi = 5.64189583547756286948079451560772585844 1e-0001,
tpi     = 0.63661977236758134307553505349005744 8;

static GENERIC pone(GENERIC), qone(GENERIC);
static const GENERIC r0[4] = {
        -6.25000000000000220305320098141321894954 8e-0002,
```



```
        1.60099845564007290132160510198150126376 2e-0003,
        -1.96388881594831375855251188439016286493 0e-0005,
        8.26391734109354975978133971341820162099 8e-0008,
};
______unchanged_portion_omitted_
```

```
**********************************************************
    7423 Tue Nov  4 18:59:45 2014
new/usr/src/lib/libm/common/C/jn.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __jn = jn
#pragma weak __yn = yn
#pragma weak jn = __jn
#pragma weak yn = __yn

/*
 * floating point Bessel's function of the 1st and 2nd kind
 * of order n: jn(n,x),yn(n,x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 * Note 2. About jn(n,x), yn(n,x)
 *      For n=0, j0(x) is called,
 *      for n=1, j1(x) is called,
 *      for n<x, forward recursion us used starting
 *      from values of j0(x) and j1(x).
 *      for n>x, a continued fraction approximation to
 *      j(n,x)/j(n-1,x) is evaluated and then backward
 *      recursion is used starting from a supposed value
 *      for j(n,x). The resulting value of j(0,x) is
 *      compared with the actual value to correct the
 *      supposed value of j(n,x).
 *
 *      yn(n,x) is similar in all respects, except
 *      that forward recursion is used for all
 *      values of n>1.
 *
 */

#include "libm.h"
#include <float.h>      /* DBL_MIN */
```



```
#include <values.h>     /* X_TLOSS */
#include "xpg6.h"       /* __xpg6 */

#define GENERIC double

static const GENERIC
        invsqrtpi = 5.64189583547756286948079451560772585844 1e-0001,
        two     = 2.0,
        zero    = 0.0,
        one     = 1.0;

GENERIC
jn(int n, GENERIC x) {
        int i, sgn;
        GENERIC a, b, temp = 0;
        GENERIC z, w, ox, on;

        /*
         * J(-n,x) = (-1)^n * J(n, x), J(n, -x) = (-1)^n * J(n, x)
         * Thus, J(-n,x) = J(n,-x)
         */
        ox = x; on = (GENERIC)n;
        if (n < 0) {
                n = -n;
                x = -x;
        }
        if (isnan(x))
                return (x*x);   /* + -> * for Cheetah */
        if (!((int) _lib_version == libm_ieee ||
              (__xpg6 & _C99SUSv3_math_errexcept) != 0)) {
            if (fabs(x) > X_TLOSS)
                        return (_SVID_libm_err(on, ox, 38));
        }
        if (n == 0)
                return (j0(x));
        if (n == 1)
                return (j1(x));
        if ((n&1) == 0)
                sgn = 0;                        /* even n */
        else
                sgn = signbit(x);       /* old n  */
        x = fabs(x);
        if (x == zero||!finite(x)) b = zero;
        else if ((GENERIC)n <= x) {
                                        /*
                                         * Safe to use
                                         *  J(n+1,x)=2n/x *J(n,x)-J(n-1,x)
                                         */
            if (x > 1.0e91) {
                                /*
                                 * x >> n**2
                                 *    Jn(x) = cos(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *   Yn(x) = sin(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *   Let s=sin(x), c=cos(x),
                                 *      xn=x-(2n+1)*pi/4, sqt2 = sqrt(2),then
                                 *
                                 *         n    sin(xn)*sqt2    cos(xn)*sqt2
                                 *      ----------------------------------
                                 *         0     s-c             c+s
                                 *         1    -s-c            -c+s
                                 *         2    -s+c            -c-s
                                 *         3     s+c             c-s
                                 */
                switch (n&3) {
                    case 0: temp =  cos(x)+sin(x); break;
                    case 1: temp = -cos(x)+sin(x); break;
```

```
126                     case 2: temp = -cos(x)-sin(x); break;
127                     case 3: temp =  cos(x)-sin(x); break;
128                 }
129                 b = invsqrtpi*temp/sqrt(x);
130             } else {
131                         a = j0(x);
132                         b = j1(x);
133                         for (i = 1; i < n; i++) {
134                     temp = b;
135                     b = b*((GENERIC)(i+i)/x) - a; /* avoid underflow */
136                     a = temp;
137                         }
138             }
139         } else {
140             if (x < 1e-9) {     /* use J(n,x) = 1/n!*(x/2)^n */
141                 b = pow(0.5*x, (GENERIC) n);
142                 if (b != zero) {
143                     for (a = one, i = 1; i <= n; i++) a *= (GENERIC)i;
144                     b = b/a;
145                 }
146             } else {
147                 /*
148                  * use backward recurrence
149                  *                      x         x^2     x^2
150                  *  J(n,x)/J(n-1,x) =  ----   ------   ------   .....
151                  *                      2n  - 2(n+1) - 2(n+2)
152                  *
153                  *                      1         1         1
154                  *  (for large x)   =  ----  ------   ------   .....
155                  *                      2n   2(n+1)   2(n+2)
156                  *                      -- - ------ - ------ -
157                  *                       x       x             x
158                  *
159                  * Let w = 2n/x and h = 2/x, then the above quotient
160                  * is equal to the continued fraction:
161                  *                  1
162                  *      = -----------------------
163                  *                     1
164                  *         w - -----------------
165                  *                    1
166                  *                   w+h - ---------
167                  *                      w+2h - ...
168                  *
169                  * To determine how many terms needed, let
170                  * Q(0) = w, Q(1) = w(w+h) - 1,
171                  * Q(k) = (w+k*h)*Q(k-1) - Q(k-2),
172                  * When Q(k) > 1e4      good for single
173                  * When Q(k) > 1e9      good for double
174                  * When Q(k) > 1e17     good for quaduple
175                  */
176             /* determin k */
177                 GENERIC t, v;
178                 double q0, q1, h, tmp; int k, m;
179                 w  = (n+n)/(double)x; h = 2.0/(double)x;
180                 q0 = w;  z = w + h; q1 = w*z - 1.0; k = 1;
181                 while (q1 < 1.0e9) {
182                         k += 1; z += h;
183                         tmp = z*q1 - q0;
184                         q0 = q1;
185                         q1 = tmp;
186                 }
187                 m = n+n;
188                 for (t = zero, i = 2*(n+k); i >= m; i -= 2) t = one/(i/x-t);
189                 a = t;
190                 b = one;
191                 /*
```



```
192                  * estimate log((2/x)^n*n!) = n*log(2/x)+n*ln(n)
193                  *  hence, if n*(log(2n/x)) > ...
194                  *  single 8.8722839355e+01
195                  *  double 7.09782712893383973096e+02
196                  *  long double 1.13565234062941439494919310779707650060170e+04
197                  *  then recurrent value may overflow and the result is
198                  *  likely underflow to zero
199                  */
200                 tmp = n;
201                 v = two/x;
202                 tmp = tmp*log(fabs(v*tmp));
203                 if (tmp < 7.09782712893383973096e+02) {
204                             for (i = n-1; i > 0; i--) {
205                                 temp = b;
206                                 b = ((i+i)/x)*b - a;
207                             a = temp;
208                                 }
209                 } else {
210                                 for (i = n-1; i > 0; i--) {
211                                     temp = b;
212                                     b = ((i+i)/x)*b - a;
213                                     a = temp;
214                                         if (b > 1e100) {
215                                                 a /= b;
216                                                 t /= b;
217                                                 b  = 1.0;
218                                         }
219                                 }
220                 }
221                         b = (t*j0(x)/b);
222             }
223         }
224         if (sgn == 1)
225                 return (-b);
226         else
227                 return (b);
228 }
```

*_____unchanged_portion_omitted_*

****
1336 Tue Nov 4 18:59:45 2014
new/usr/src/lib/libm/common/C/lgamma.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgamma = lgamma
#pragma weak lgamma = __lgamma

#include "libm.h"

extern int signgam;

double
lgamma(double x) {
        double  g;

        if (!finite(x))
                return (x * x);

        g = rint(x);
        if (x == g && x <= 0.0) {
                signgam = 1;
                return (_SVID_libm_err(x, x, 15));
        }

        g = __k_lgamma(x, &signgam);
        if (!finite(g))
            g = _SVID_libm_err(x, x, 14);
        return (g);
}
```

_____***unchanged_portion_omitted_***

****
1336 Tue Nov 4 18:59:46 2014
new/usr/src/lib/libm/common/C/lgamma_r.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgamma_r = lgamma_r
#pragma weak lgamma_r = __lgamma_r

#include "libm.h"

double
lgamma_r(double x, int *signgamp) {
        double  g;

        if (isnan(x))
                return (x * x);

        g = rint(x);
        if (x == g && x <= 0.0) {
                *signgamp = 1;
                return (_SVID_libm_err(x, x, 15));
        }

        g = __k_lgamma(x, signgamp);
        if (!finite(g))
            g = _SVID_libm_err(x, x, 14);
        return (g);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    5100 Tue Nov  4 18:59:46 2014
new/usr/src/lib/libm/common/C/libm.h
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifndef _LIBM_H
#define _LIBM_H

#include <sys/isa_defs.h>

#ifdef _ASM
/* BEGIN CSTYLED */

/*
 * Disable amd64 assembly code profiling for now.
 */
#if defined(__amd64)
#undef PROF
#endif

#include <sys/asm_linkage.h>

#define NAME(x) x
#define TEXT    .section        ".text"
#define DATA    .section        ".data"
#define RO_DATA .section        ".rodata"
#define IDENT(x)        .ident  x

#if defined(__sparc)

#define LIBM_ANSI_PRAGMA_WEAK(sym,stype) \
        .weak __/**/sym;                 \
        .type __/**/sym,#stype;          \
__/**/sym       = sym
        .weak sym; \
        .type sym,#stype; \
sym     = __/**/sym
```



```
#ifndef SET_FILE
#define SET_FILE(x) \
        .file   x
#endif  /* !defined(SET_FILE) */

#ifdef PIC
/*
 * One should *never* pass o7 to PIC_SETUP.
 */
#define PIC_SETUP(via) \
9:      call    8f; \
        sethi   %hi(NAME(_GLOBAL_OFFSET_TABLE_)-(9b-.)),%via; \
8:      or      %via,%lo(NAME(_GLOBAL_OFFSET_TABLE_)-(9b-.)),%via; \
        add     %via,%o7,%via
/*
 * Must save/restore %o7 in leaf routines; may *not* use jmpl!
 */
#define PIC_LEAF_SETUP(via) \
        or      %g0,%o7,%g1; \
9:      call    8f; \
        sethi   %hi(NAME(_GLOBAL_OFFSET_TABLE_)-(9b-.)),%via; \
8:      or      %via,%lo(NAME(_GLOBAL_OFFSET_TABLE_)-(9b-.)),%via; \
        add     %via,%o7,%via; \
        or      %g0,%g1,%o7
#ifdef __sparcv9
#define PIC_SET(via,sym,dst)    ldx     [%via+sym],%dst
#else   /* defined(__sparcv9) */
#define PIC_SET(via,sym,dst)    ld      [%via+sym],%dst
#endif  /* defined(__sparcv9) */
#else   /* defined(PIC) */
#define PIC_SETUP(via)
#define PIC_LEAF_SETUP(via)
#ifdef __sparcv9
/*
 * g1 is used as scratch register in V9 mode
 */
#define PIC_SET(via,sym,dst)    setx    sym,%g1,%dst
#else   /* defined(__sparcv9) */
#define PIC_SET(via,sym,dst)    set     sym,%dst
#endif  /* defined(__sparcv9) */
#endif  /* defined(PIC) */

/*
 * Workaround for 4337025: MCOUNT in asm_linkage.h does not support __sparcv9
 */
#if defined(PROF) && defined(__sparcv9)

#undef MCOUNT_SIZE
#undef MCOUNT

#if !defined(PIC)
#define MCOUNT_SIZE     (9*4)   /* 9 instructions */
#define MCOUNT(x) \
        save    %sp, -SA(MINFRAME), %sp; \
        sethi   %hh(.L_/**/x/**/1), %o0; \
        sethi   %lm(.L_/**/x/**/1), %o1; \
        or      %o0, %hm(.L_/**/x/**/1), %o0; \
        or      %o1, %lo(.L_/**/x/**/1), %o1; \
        sllx    %o0, 32, %o0; \
        call    _mcount; \
        or      %o0, %o1, %o0; \
        restore; \
        .common .L_/**/x/**/1, 8, 8
#elif defined(PIC32)
#define MCOUNT_SIZE     (10*4)  /* 10 instructions */
#define MCOUNT(x) \
```

```
125         save    %sp,-SA(MINFRAME),%sp; \
126 1:      call    .+8; \
127         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(1b-.)),%o0; \
128         sethi   %hi(.L_/**/x/**/1),%o1; \
129         add     %o0,%lo(_GLOBAL_OFFSET_TABLE_-(1b-.)),%o0; \
130         add     %o1,%lo(.L_/**/x/**/1),%o1; \
131         add     %o0,%o7,%o0; \
132         call    _mcount; \
133         ldx     [%o0+%o1],%o0; \
134         restore; \
135         .common .L_/**/x/**/1,8,8
136 #else   /* PIC13 */
137 #define MCOUNT_SIZE     (8*4)   /* 8 instructions */
138 #define MCOUNT(x) \
139         save    %sp,-SA(MINFRAME),%sp; \
140 1:      call    .+8; \
141         sethi   %hi(_GLOBAL_OFFSET_TABLE_-(1b-.)),%o0; \
142         add     %o0,%lo(_GLOBAL_OFFSET_TABLE_-(1b-.)),%o0; \
143         add     %o0,%o7,%o0; \
144         call    _mcount; \
145         ldx     [%o0+%lo(.L_/**/x/**/1)],%o0; \
146         restore; \
147         .common .L_/**/x/**/1,8,8
148 #endif  /* !defined(PIC) */
149 #endif /* defined(PROF) && defined(__sparcv9) */

151 #elif defined(__x86)

153 #define LIBM_ANSI_PRAGMA_WEAK(sym,stype) \
154         .weak __/**/sym;                \
155         .type __/**/sym,@stype;         \
156 __/**/sym       = sym
154         .weak sym; \
155         .type sym,@stype; \
156 sym     = __/**/sym

158 #ifdef PIC
159 #if defined(__amd64)
160 #define PIC_SETUP(x)
161 #define PIC_WRAPUP
162 #define PIC_F(x)        x@PLT
163 #define PIC_G(x)        x@GOTPCREL(%rip)
164 #define PIC_L(x)        x(%rip)
165 #define PIC_G_LOAD(insn,sym,dst) \
166         movq    PIC_G(sym),%dst; \
167         insn    (%dst),%dst
168 #else
169 #define PIC_SETUP(label) \
170         pushl   %ebx; \
171         call    .label; \
172 .label: popl    %ebx; \
173         addl    $_GLOBAL_OFFSET_TABLE_+[.-.label],%ebx
174 #define PIC_WRAPUP      popl    %ebx
175 #define PIC_F(x)        x@PLT
176 #define PIC_G(x)        x@GOT(%ebx)
177 #define PIC_L(x)        x@GOTOFF(%ebx)
178 #define PIC_G_LOAD(insn,sym,dst) \
179         mov     PIC_G(sym),%dst; \
180         insn    (%dst),%dst
181 #endif
182 #else   /* defined(PIC) */
183 #define PIC_SETUP(x)
184 #define PIC_WRAPUP
185 #define PIC_F(x)        x
186 #define PIC_G(x)        x
187 #define PIC_L(x)        x
```



```
188 #define PIC_G_LOAD(insn,sym,dst)        insn    sym,%dst
189 #endif  /* defined(PIC) */

191 #else
192 #error Unknown architecture
193 #endif

195 /* END CSTYLED */
196 #else   /* defined(_ASM) */

198 #include "libm_macros.h"
199 #include "libm_synonyms.h"
199 #include "libm_protos.h"
200 #include "libm_inlines.h"
201 #include <math.h>
202 #if defined(__SUNPRO_C)
203 #include <sunmath.h>
204 #endif

206 #endif  /* defined(_ASM) */

208 #endif  /* _LIBM_H */
```

```
**********************************************************
    7989 Tue Nov  4 18:59:47 2014
new/usr/src/lib/libm/common/C/log.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log = log
#pragma weak log = __log

/* INDENT OFF */
/*
 * log(x)
 * Table look-up algorithm with product polynomial approximation.
 * By K.C. Ng, Oct 23, 2004. Updated Oct 18, 2005.
 *
 * (a). For x in [1-0.125, 1+0.1328125], using a special approximation:
 *      Let f = x - 1 and z = f*f.
 *      return f + ((a1*z) *
 *                  ((a2 + (a3*f)*(a4+f)) + (f*z)*(a5+f))) *
 *                  (((a6 + f*(a7+f)) + (f*z)*(a8+f)) *
 *                  ((a9 + (a10*f)*(a11+f)) + (f*z)*(a12+f)))
 * a1   -6.88821452420390473170286327331268694251775741577e-0002,
 * a2    1.97493380704769294631262255279580131173133850098e+0000,
 * a3    2.24963218866067560242072431719861924648284912109e+0000,
 * a4   -9.02975906958474405783476868236903101205825805664e-0001,
 * a5   -1.47391630715542865104339398385491222143173217773e+0000,
 * a6    1.86846544648220058704168877738993614912033081055e+0000,
 * a7    1.82277370459347465292410106485476717352867126465e+0000,
 * a8    1.25295479915214102994980294170090928673744201660e+0000,
 * a9    1.96709676945198275177517643896862864494323730469e+0000,
 * a10  -4.00127989749189894030934055990655906498432159424e-0001,
 * a11   3.01675528558798333733648178167641162872314453125e+0000,
 * a12  -9.52325445049240770778453679668018594384193420410e-0001,
 *
 *      with remez error |(log(1+f) - P(f))/f| <= 2**-56.81 and
 *
 * (b). For 0.09375 <= x < 24
 *      Use an 8-bit table look-up (3-bit for exponent and 5 bit for
 *      significand):
```



```
 *      Let ix stands for the high part of x in IEEE double format.
 *      Since 0.09375 <= x < 24, we have
 *                      0x3fb80000 <= ix < 0x40380000.
 *      Let j = (ix - 0x3fb80000) >> 15. Then  0 <= j < 256. Choose
 *      a Y[j] such that HIWORD(Y[j]) ~ 0x3fb8400 + (j<<15) (the middle
 *      number between 0x3fb80000 + (j<<15) and 3fb80000 + ((j+1)<<15)),
 *      and at the same time 1/Y[j] as well as log(Y[j]) are very close
 *      to 53-bits floating point numbers.
 *      A table of Y[j], 1/Y[j], and log(Y[j]) are pre-computed and thus
 *              log(x)  = log(Y[j]) + log(1 + (x-Y[j])*(1/Y[j]))
 *                      = log(Y[j]) + log(1 + s)
 *      where
 *              s = (x-Y[j])*(1/Y[j])
 *      We compute max (x-Y[j])*(1/Y[j]) for the chosen Y[j] and obtain
 *      |s| < 0.0154. By applying remez algorithm with Product Polynomial
 *      Approximiation, we find the following approximated of log(1+s)
 *              (b1*s)*(b2+s*(b3+s))*((b4+s*b5)+(s*s)*(b6+s))*(b7+s*(b8+s))
 *      with remez error |log(1+s) - P(s)| <= 2**-63.5
 *
 * (c). Otherwise, get "n", the exponent of x, and then normalize x to
 *      z in [1,2). Then similar to (b) find a Y[i] that matches z to 5.5
 *      significant bits. Then
 *          log(x) = n*ln2 + log(Y[i]) + log(z/Y[i]).
 *
 * Special cases:
 *      log(x) is NaN with signal if x < 0 (including -INF) ;
 *      log(+INF) is +INF; log(0) is -INF with signal;
 *      log(NaN) is that NaN with no signal.
 *
 * Maximum error observed: less than 0.90 ulp
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */
/* INDENT ON */

#include "libm.h"

extern const double _TBL_log[];

static const double P[] = {
/* ONE    */  1.0,
/* TWO52 */  4503599627370496.0,
/* LN2HI */  6.93147180369123816490e-01,        /* 3fe62e42, fee00000 */
/* LN2LO */  1.90821492927058770002e-10,        /* 3dea39ef, 35793c76 */
/* A1    */ -6.88821452420390473170286327331268694251775741577e-0002,
/* A2    */  1.97493380704769294631262255279580131173133850098e+0000,
/* A3    */  2.24963218866067560242072431719861924648284912109e+0000,
/* A4    */ -9.02975906958474405783476868236903101205825805664e-0001,
/* A5    */ -1.47391630715542865104339398385491222143173217773e+0000,
/* A6    */  1.86846544648220058704168877738993614912033081055e+0000,
/* A7    */  1.82277370459347465292410106485476717352867126465e+0000,
/* A8    */  1.25295479915214102994980294170090928673744201660e+0000,
/* A9    */  1.96709676945198275177517643896862864494323730469e+0000,
/* A10   */ -4.00127989749189894030934055990655906498432159424e-0001,
/* A11   */  3.01675528558798333733648178167641162872314453125e+0000,
/* A12   */ -9.52325445049240770778453679668018594384193420410e-0001,
/* B1    */ -1.25041641589283658575482149899471551179885864258e-0001,
/* B2    */  1.87161713283355151891381127914642725337613123482e+0000,
/* B3    */ -1.89082956295731507978530316904652863740921020508e+0000,
/* B4    */ -2.50562891673640253387134180229622870683670043945e+0000,
/* B5    */  1.64822828085258366037635369139024987816810607910e+0000,
/* B6    */ -1.24409107065868340669112512841820716857910156250e+0000,
```

```
 127 /* B7    */  1.70534231658220414296067701798165217041969299316e+0000,
 128 /* B8    */  1.99196833784655646937267192697618156671524047852e+0000,
 129 };
______unchanged_portion_omitted_
```

```
**********************************************************
    7235 Tue Nov  4 18:59:47 2014
new/usr/src/lib/libm/common/C/log10.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log10 = log10
#pragma weak log10 = __log10

/* INDENT OFF */
/*
 * log10(x) = log(x)/log10
 *
 * Base on Table look-up algorithm with product polynomial
 * approximation for log(x).
 *
 * By K.C. Ng, Nov 29, 2004
 *
 * (a). For x in [1-0.125, 1+0.125], from log.c we have
 *      log(x) =  f + ((a1*f^2) *
 *                ((a2 + (a3*f)*(a4+f)) + (f^3)*(a5+f))) *
 *                (((a6 + f*(a7+f)) + (f^3)*(a8+f)) *
 *                ((a9 + (a10*f)*(a11+f)) + (f^3)*(a12+f)))
 *      where f = x - 1.
 *      (i) modify a1 <- a1 / log10
 *      (ii) 1/log10 = 0.4342944819...
 *                   = 0.4375 - 0.003205518... (7 bit shift)
 *           Let lgv = 0.4375 - 1/log10, then
 *           lgv = 0.00320551809674817234887108308339 5...,
 *      (iii) f*0.4375 is exact because f has 3 trailing zero.
 *      (iv) Thus, log10(x) = f*0.4375 - (lgv*f - PPoly)
 *
 * (b). For 0.09375 <= x < 24
 *      Let j = (ix - 0x3fb80000) >> 15. Look up Y[j], 1/Y[j], and log(Y[j])
 *      from _TBL_log.c. Then
 *              log10(x)  = log10(Y[j]) + log10(1 + (x-Y[j])*(1/Y[j]))
 *                        = log(Y[j])(1/log10) + log10(1 + s)
 *      where
 *              s = (x-Y[j])*(1/Y[j])
```



```
 *      From log.c, we have log(1+s) =
 *                                  2               2                   2
 *              (b s) (b + b s + s ) [b + b s + s (b + s)] (b + b s + s )
 *                1     2   3          4   5        6        7   8
 *
 *      By setting b1 <- b1/log10, we have
 *              log10(x) = 0.4375 * T - (lgv * T - POLY(s))
 *
 * (c). Otherwise, get "n", the exponent of x, and then normalize x to
 *      z in [1,2). Then similar to (b) find a Y[i] that matches z to 5.5
 *      significant bits. Then
 *          log(x) = n*ln2 + log(Y[i]) + log(z/Y[i]).
 *          log10(x) = n*(ln2/ln10) + log10(z).
 *
 * Special cases:
 *      log10(x) is NaN with signal if x < 0 (including -INF) ;
 *      log10(+INF) is +INF; log10(0) is -INF with signal;
 *      log10(NaN) is that NaN with no signal.
 *
 * Maximum error observed: less than 0.89 ulp
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */
/* INDENT ON */

#include "libm.h"

extern const double _TBL_log[];

static const double P[] = {
/* ONE   */  1.0,
/* TWO52 */  4503599627370496.0,
/* LNAHI */  3.01029995607677847147e-01,        /* 3FD34413 50900000 */
/* LNALO */  5.630334806675097698 41e-11,       /* 3DCEF3FD E623E256 */
/* A1    */ -2.91425219601365825073854807070445828021 84e-02,
/* A2    */  1.99628461483039965074226529395673424005508422852e+0000,
/* A3    */  2.26812367662950720159642514772713184356689453125e+0000,
/* A4    */ -9.05030639084976384900471657601883634924888610840e-0001,
/* A5    */ -1.48275767132434044270894446526654064655303955078e+0000,
/* A6    */  1.88158320939722756293122074566781520843505859375e+0000,
/* A7    */  1.83309386046986411145098827546462416648864746094e+0000,
/* A8    */  1.24847063988317086291601754055591300129890441895e+0000,
/* A9    */  1.98372421445537705508854742220137268304824829102e+0000,
/* A10   */ -3.94711735767898475035764249696512706577777862549e-0001,
/* A11   */  3.07890395362954372160402272129431366920471191406e+0000,
/* A12   */ -9.60099585275022149311041630426188930869102478027e-0001,
/* B1    */ -5.43048949503500529608380967524915402866 89e-02,
/* B2    */  1.87161713283355151891381127914642725337613123482e+0000,
/* B3    */ -1.89082956295731507978530316904652863740921020508e+0000,
/* B4    */ -2.50562891673640253387134180229622870683670043945e+0000,
/* B5    */  1.64822828085258366037635369139024987816810607910e+0000,
/* B6    */ -1.24409107065868340669112512841820716857910156250e+0000,
/* B7    */  1.70534231658220414296067701798165217041969299316e+0000,
/* B8    */  1.99196833784655646937267192697618156671524047852e+0000,
/* LGH   */  0.4375,
/* LGL   */  0.00320551809674817234887108308339 5,
/* LG10V */  0.43429448190325182765112891891660509576226,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6364 Tue Nov  4 18:59:48 2014
new/usr/src/lib/libm/common/C/log1p.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log1p = log1p
#pragma weak log1p = __log1p

/* INDENT OFF */
/*
 * Method :
 *   1. Argument Reduction: find k and f such that
 *                      1+x = 2^k * (1+f),
 *         where  sqrt(2)/2 < 1+f < sqrt(2) .
 *
 *      Note. If k=0, then f=x is exact. However, if k != 0, then f
 *      may not be representable exactly. In that case, a correction
 *      term is need. Let u=1+x rounded. Let c = (1+x)-u, then
 *      log(1+x) - log(u) ~ c/u. Thus, we proceed to compute log(u),
 *      and add back the correction term c/u.
 *      (Note: when x > 2**53, one can simply return log(x))
 *
 *   2. Approximation of log1p(f).
 *      Let s = f/(2+f) ; based on log(1+f) = log(1+s) - log(1-s)
 *               = 2s + 2/3 s**3 + 2/5 s**5 + .....,
 *               = 2s + s*R
 *      We use a special Reme algorithm on [0,0.1716] to generate
 *      a polynomial of degree 14 to approximate R The maximum error
 *      of this polynomial approximation is bounded by 2**-58.45. In
 *      other words,
 *                      2      4      6      8      10      12      14
 *          R(z) ~ Lp1*s +Lp2*s +Lp3*s +Lp4*s +Lp5*s  +Lp6*s  +Lp7*s
 *      (the values of Lp1 to Lp7 are listed in the program)
 *      and
 *          |      2          14          |     -58.45
 *          | Lp1*s +...+Lp7*s    -  R(z) | <= 2
 *          |                             |
 *      Note that 2s = f - s*f = f - hfsq + s*hfsq, where hfsq = f*f/2.
```



```
 *      In order to guarantee error in log below 1ulp, we compute log
 *      by
 *              log1p(f) = f - (hfsq - s*(hfsq+R)).
 *
 *      3. Finally, log1p(x) = k*ln2 + log1p(f).
 *                           = k*ln2_hi+(f-(hfsq-(s*(hfsq+R)+k*ln2_lo)))
 *         Here ln2 is splitted into two floating point number:
 *                      ln2_hi + ln2_lo,
 *         where n*ln2_hi is always exact for |n| < 2000.
 *
 * Special cases:
 *      log1p(x) is NaN with signal if x < -1 (including -INF) ;
 *      log1p(+INF) is +INF; log1p(-1) is -INF with signal;
 *      log1p(NaN) is that NaN with no signal.
 *
 * Accuracy:
 *      according to an error analysis, the error is always less than
 *      1 ulp (unit in the last place).
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following
 * constants. The decimal values may be used, provided that the
 * compiler will convert from decimal to binary accurately enough
 * to produce the hexadecimal values shown.
 *
 * Note: Assuming log() return accurate answer, the following
 *       algorithm can be used to compute log1p(x) to within a few ULP:
 *
 *              u = 1+x;
 *              if (u == 1.0) return x ; else
 *                         return log(u)*(x/(u-1.0));
 *
 *       See HP-15C Advanced Functions Handbook, p.193.
 */
/* INDENT ON */

#include "libm.h"

static const double xxx[] = {
/* ln2_hi */    6.93147180369123816490e-01,     /* 3fe62e42 fee00000 */
/* ln2_lo */    1.90821492927058770002e-10,     /* 3dea39ef 35793c76 */
/* two54 */     1.80143985094819840000e+16,     /* 43500000 00000000 */
/* Lp1 */       6.66666666666735130e-01,        /* 3FE55555 55555593 */
/* Lp2 */       3.99999999940941908e-01,        /* 3FD99999 9997FA04 */
/* Lp3 */       2.85714287436623914e-01,        /* 3FD24924 94229359 */
/* Lp4 */       2.22221984321497839e-01,        /* 3FCC71C5 1D8E78AF */
/* Lp5 */       1.81835721616180501e-01,        /* 3FC74664 96CB03DE */
/* Lp6 */       1.53183769920937332e-01,        /* 3FC39A09 D078C69F */
/* Lp7 */       1.47981986051165859le-01,       /* 3FC2F112 DF3E5244 */
/* zero */      0.0
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7366 Tue Nov  4 18:59:48 2014
new/usr/src/lib/libm/common/C/log2.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log2 = log2
#pragma weak log2 = __log2

/* INDENT OFF */
/*
 * log2(x) = log(x)/log2
 *
 * Base on Table look-up algorithm with product polynomial
 * approximation for log(x).
 *
 * By K.C. Ng, Nov 29, 2004
 *
 * (a). For x in [1-0.125, 1+0.125], from log.c we have
 *      log(x) =  f + ((a1*f^2) *
 *                ((a2 + (a3*f)*(a4+f)) + (f^3)*(a5+f))) *
 *                (((a6 + f*(a7+f)) + (f^3)*(a8+f)) *
 *                ((a9 + (a10*f)*(a11+f)) + (f^3)*(a12+f)))
 *      where f = x - 1.
 *      (i) modify a1 <- a1 / log2
 *      (ii) 1/log2 = 1.4426950408889634...
 *                  = 1.5 - 0.057304959... (4 bit shift)
 *           Let lv = 1.5 - 1/log2, then
 *           lv = 0.05730495911103659264007531899810795666532 5,
 *      (iii) f*1.5 is exact because f has 3 trailing zero.
 *      (iv) Thus, log2(x) = f*1.5 - (lv*f - PPoly)
 *
 * (b). For 0.09375 <= x < 24
 *      Let j = (ix - 0x3fb80000) >> 15. Look up Y[j], 1/Y[j], and log(Y[j])
 *      from _TBL_log.c. Then
 *              log2(x)  = log2(Y[j]) + log2(1 + (x-Y[j])*(1/Y[j]))
 *                       = log(Y[j])(1/log2) + log2(1 + s)
 *      where
```



```
 *              s = (x-Y[j])*(1/Y[j])
 *      From log.c, we have log(1+s) =
 *                                  2                   2                   2
 *              (b s) (b + b s + s ) [b + b s + s (b + s)] (b + b s + s )
 *                1     2   3         4   5       6       7   8
 *
 *      By setting b1 <- b1/log2, we have
 *              log2(x) = 1.5 * T - (lv * T - POLY(s))
 *
 * (c). Otherwise, get "n", the exponent of x, and then normalize x to
 *      z in [1,2). Then similar to (b) find a Y[i] that matches z to 5.5
 *      significant bits. Then
 *          log2(x) = n + log2(z).
 *
 * Special cases:
 *      log2(x) is NaN with signal if x < 0 (including -INF) ;
 *      log2(+INF) is +INF; log2(0) is -INF with signal;
 *      log2(NaN) is that NaN with no signal.
 *
 * Maximum error observed: less than 0.84 ulp
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"

extern const double _TBL_log[];

static const double P[] = {
/* ONE   */  1.0,
/* TWO52 */  4503599627370496.0,
/* LN10V */  1.44269504088896340735992468100189204333 47,   /* 1/log10 */
/* ZERO  */  0.0,
/* A1    */ -9.6809362455249638217841932228967194640116e-02,
/* A2    */  1.9962846148303996507422652939567342400550842 2852e+0000,
/* A3    */  2.2681236766295072015964251477271318435668945 3125e+0000,
/* A4    */ -9.0503063908497638490047165760188363492488861 0840e-0001,
/* A5    */ -1.4827576713243404427089444652665406465530395 5078e+0000,
/* A6    */  1.8815832093972275629312207456678152084350585 9375e+0000,
/* A7    */  1.8330938604698641114509882754646241664886486 4746094e+0000,
/* A8    */  1.2484706398831708629160175405559130012989044 1895e+0000,
/* A9    */  1.9837242144553770550885474222013726830482402 9102e+0000,
/* A10   */ -3.9471173576789847503576424969651270657777786 2549e-0001,
/* A11   */  3.0789039536295437216040227212943136692047119 1406e+0000,
/* A12   */ -9.6009958527502214931104163042618893086910247 8027e-0001,
/* B1    */ -1.8039695622547469514898963204616532885451e-01,
/* B2    */  1.8716171328335515189138112791464272533761312 3482e+0000,
/* B3    */ -1.8908295629573150797853031690465286374092102 0508e+0000,
/* B4    */ -2.5056289167364025338713418022962287068367004 3945e+0000,
/* B5    */  1.6482282808525836603763536913902498781681060 7910e+0000,
/* B6    */ -1.2440910706586834066911251284182071685791015 6250e+0000,
/* B7    */  1.7053423165822041429606770179816521704196929 9316e+0000,
/* B8    */  1.9919683378465564693726719269761815667152404 7852e+0000,
/* LGH   */  1.5,
/* LGL   */  0.05730495911103659264007531899810795666532 5,
};
______unchanged_portion_omitted_
```

```
**********************************************************
    2116 Tue Nov  4 18:59:49 2014
new/usr/src/lib/libm/common/C/logb.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __logb = logb
#pragma weak _logb = logb
#pragma weak logb = __logb
#pragma weak _logb = __logb

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */
#define _C99SUSv3_logb  _C99SUSv3_logb_subnormal_is_like_ilogb

#if defined(__x86)
static const double two52 = 4503599627370496.0;
#else
/*
 * v: high part of a non-zero subnormal |x|; w: low part of |x|
 */
static int
ilogb_subnormal(unsigned v, unsigned w) {
        int r = -1022 - 52;

        if (v)
                r += 32;
        else
                v = w;
        if (v & 0xffff0000)
                r += 16, v >>= 16;
        if (v & 0xff00)
                r += 8, v >>= 8;
        if (v & 0xf0)
                r += 4, v >>= 4;
        v <<= 1;
        return (r + ((0xffffaa50 >> v) & 0x3));
}
______unchanged_portion_omitted_
```

****
2272 Tue Nov 4 18:59:49 2014
new/usr/src/lib/libm/common/C/nextafter.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __nextafter = nextafter
#pragma weak _nextafter = nextafter
#pragma weak nextafter = __nextafter
#pragma weak _nextafter = __nextafter

#include "libm.h"
#include <float.h>              /* DBL_MIN */

double
nextafter(double x, double y) {
        int             hx, hy, k;
        double          ans;
        unsigned        lx;
        volatile double dummy;

        hx = ((int *)&x)[HIWORD];
        lx = ((int *)&x)[LOWORD];
        hy = ((int *)&y)[HIWORD];
        k = (hx & ~0x80000000) | lx;

        if (x == y)
                return (y);             /* C99 requirement */
        if (x != x || y != y)
                return (x * y);
        if (k == 0) {                   /* x = 0 */
                k = hy & 0x80000000;
                ((int *)&ans)[HIWORD] = k;
                ((int *)&ans)[LOWORD] = 1;
        } else if (hx >= 0) {
                if (x > y) {
                        ((int *)&ans)[LOWORD] = lx - 1;
                        k = (lx == 0)? hx - 1 : hx;
                        ((int *)&ans)[HIWORD] = k;
```



```
                } else {
                        ((int *)&ans)[LOWORD] = lx + 1;
                        k  = (lx == 0xffffffff)? hx + 1 : hx;
                        ((int *)&ans)[HIWORD] = k;
                }
        } else {
                if (x < y) {
                        ((int *)&ans)[LOWORD] = lx - 1;
                        k = (lx == 0)? hx - 1 : hx;
                        ((int *)&ans)[HIWORD] = k;
                } else {
                        ((int *)&ans)[LOWORD] = lx + 1;
                        k  = (lx == 0xffffffff)? hx + 1 : hx;
                        ((int *)&ans)[HIWORD] = k;
                }
        }
        k = (k >> 20) & 0x7ff;
        if (k == 0x7ff) {
                /* overflow */
                return (_SVID_libm_err(x, y, 46));
#if !defined(__lint)
        } else if (k == 0) {
                /* underflow */
                dummy = DBL_MIN * copysign(DBL_MIN, x);
#endif
        }
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   10175 Tue Nov  4 18:59:50 2014
new/usr/src/lib/libm/common/C/pow.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __pow = pow
#pragma weak pow = __pow

/*
 * pow(x,y) return x**y
 *                    n
 * Method:  Let x =  2   * (1+f)
 *      1. Compute and return log2(x) in two pieces:
 *              log2(x) = w1 + w2,
 *         where w1 has 24 bits trailing zero.
 *      2. Perform y*log2(x) by simulating muti-precision arithmetic
 *      3. Return x**y = exp2(y*log(x))
 *
 * Special cases:
 *      1.  (anything) ** +-0 is 1
 *      1'. 1 ** (anything)   is 1      (C99; 1 ** +-INF/NAN used to be NAN)
 *      2.  (anything) ** 1   is itself
 *      3.  (anything except 1) ** NAN is NAN ("except 1" is C99)
 *      4.  NAN ** (anything except 0) is NAN
 *      5.  +-(|x| > 1) **  +INF is +INF
 *      6.  +-(|x| > 1) **  -INF is +0
 *      7.  +-(|x| < 1) **  +INF is +0
 *      8.  +-(|x| < 1) **  -INF is +INF
 *      9.  -1          ** +-INF is 1   (C99; -1 ** +-INF used to be NAN)
 *      10. +0 ** (+anything except 0, NAN)              is +0
 *      11. -0 ** (+anything except 0, NAN, odd integer)  is +0
 *      12. +0 ** (-anything except 0, NAN)              is +INF
 *      13. -0 ** (-anything except 0, NAN, odd integer)  is +INF
 *      14. -0 ** (odd integer) = -( +0 ** (odd integer) )
 *      15. +INF ** (+anything except 0,NAN) is +INF
 *      16. +INF ** (-anything except 0,NAN) is +0
 *      17. -INF ** (anything)  = -0 ** (-anything)
```



```
 *      18. (-anything) ** (integer) is (-1)**(integer)*(+anything**integer)
 *      19. (-anything except 0 and inf) ** (non-integer) is NAN
 *
 * Accuracy:
 *      pow(x,y) returns x**y nearly rounded. In particular
 *                      pow(integer,integer)
 *      always returns the correct integer provided it is representable.
 */

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */
#define _C99SUSv3_pow   _C99SUSv3_pow_treats_Inf_as_an_even_int

static const double zero = 0.0, one = 1.0, two = 2.0;

extern const double _TBL_log2_hi[], _TBL_log2_lo[];
static const double
        two53 = 9007199254740992.0,
        A1_hi = 2.88539004325866699219875,
        A1_lo = 3.85192598250350419636060002e-8,
        A1 = 2.88539008177792681722254196360600202608 6e+0000,
        A2 = 9.61796693920727082838054397985228625586 2e-0001,
        A3 = 5.77080768088787596486885312487369620199 5e-0001,
        B0_hi = 2.88539004325866699219875,
        B0_lo = 3.85192598225327930563743205850e-8,
        B0 = 2.88539008177792681472029305637432058568 9e+0000,
        B1 = 9.61796693925975513894920235039620025763 2e-0001,
        B2 = 5.77078016358568700078211277644879795338 2e-0001,
        B3 = 4.12198548894877152329017451246177835495 3e-0001,
        B4 = 3.20759053481243297043364178902266685019 3e-0001;

static double
log2_x(double x, double *w) {
        double f, s, z, qn, h, t;
        int *px = (int *) &x;
        int *pz = (int *) &z;
        int i, j, ix, n;

        n = 0;
        ix = px[HIWORD];
        if (ix >= 0x3fef03f1 && ix < 0x3ff08208) {      /* 65/63 > x > 63/65 */
                double f1, v;
                f = x - one;
                if (((ix - 0x3ff00000) | px[LOWORD]) == 0) {
                        *w = zero;
                        return (zero);          /* log2(1)= +0 */
                }
                qn = one / (two + f);
                s = f * qn;                             /* |s|<2**-6 */
                v = s * s;
                h = (double) ((float) s);
                f1 = (double) ((float) f);
                t = qn * (((f - two * h) - h * f1) - h * (f - f1));
                                                        /* s = h+t */
                f1 = h * B0_lo + s * (v * (B1 + v * (B2 + v * (B3 + v * B4))));
                t = f1 + t * B0;
                h *= B0_hi;
                s = (double) ((float) (h + t));
                *w = t - (s - h);
                return (s);
        }
        if (ix < 0x00100000) {                          /* subnormal x */
                x *= two53;
                n = -53;
                ix = px[HIWORD];
        }
```

```
 127         /* LARGE N */
 128         n += ((ix + 0x1000) >> 20) - 0x3ff;
 129         ix = (ix & 0x000fffff) | 0x3ff00000;            /* scale x to [1,2] */
 130         px[HIWORD] = ix;
 131         i = ix + 0x1000;
 132         pz[HIWORD] = i & 0xffffe000;
 133         pz[LOWORD] = 0;
 134         qn = one / (x + z);
 135         f = x - z;
 136         s = f * qn;
 137         h = (double) ((float) s);
 138         t = qn * ((f - (h + h) * z) - h * f);
 139         j = (i >> 13) & 0x7f;
 140         f = s * s;
 141         t = t * A1 + h * A1_lo;
 142         t += (s * f) * (A2 + f * A3);
 143         qn = h * A1_hi;
 144         s = n + _TBL_log2_hi[j];
 145         h = qn + s;
 146         t += _TBL_log2_lo[j] - ((h - s) - qn);
 147         f = (double) ((float) (h + t));
 148         *w = t - (f - h);
 149         return (f);
 150 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2344 Tue Nov  4 18:59:50 2014
new/usr/src/lib/libm/common/C/remainder.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remainder = remainder
#pragma weak remainder = __remainder

/*
 * remainder(x,p)
 * Code originated from 4.3bsd.
 * Modified by K.C. Ng for SUN 4.0 libm.
 * Return :
 *      returns  x REM p  =  x - [x/p]*p as if in infinite precise arithmetic,
 *      where [x/p] is the (inifinite bit) integer nearest x/p (in half way
 *      case choose the even one).
 * Method :
 *      Based on fmod() return x-[x/p]chopped*p exactly.
 */

#include "libm.h"

static const double zero = 0.0, half = 0.5;

double
remainder(double x, double p) {
        double  halfp;
        int     ix, hx, hp;

        ix = ((int *)&x)[HIWORD];
        hx = ix & ~0x80000000;
        hp = ((int *)&p)[HIWORD] & ~0x80000000;

        if (hp > 0x7ff00000 || (hp == 0x7ff00000 && ((int *)&p)[LOWORD] != 0))
                return (x * p);
        if (hx > 0x7ff00000 || (hx == 0x7ff00000 && ((int *)&x)[LOWORD] != 0))
                return (x * p);
```



```
        if ((hp | ((int *)&p)[LOWORD]) == 0 || hx == 0x7ff00000)
                return (_SVID_libm_err(x, p, 28));

        p = fabs(p);
        if (hp < 0x7fe00000)
                x = fmod(x, p + p);
        x = fabs(x);
        if (hp < 0x00200000) {
                if (x + x > p) {
                        if (x == p)     /* avoid x-x=-0 in RM mode */
                                return ((ix < 0)? -zero : zero);
                        x -= p;
                        if (x + x >= p)
                                x -= p;
                }
        } else {
                halfp = half * p;
                if (x > halfp) {
                        if (x == p)     /* avoid x-x=-0 in RM mode */
                                return ((ix < 0)? -zero : zero);
                        x -= p;
                        if (x >= halfp)
                                x -= p;
                }
        }
        return ((ix < 0)? -x : x);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2042 Tue Nov  4 18:59:51 2014
new/usr/src/lib/libm/common/C/rint.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __rint = rint
#pragma weak rint = __rint

/*
 * rint(x) return x rounded to integral according to the rounding direction
 * rint(x) returns result with the same sign as x's,  including 0.0.
 */

#include "libm.h"

#if defined(__i386) && !defined(__amd64) && (!defined(__FLT_EVAL_METHOD__) || \
        __FLT_EVAL_METHOD__ != 0)
extern enum fp_precision_type __swapRP(enum fp_precision_type);
#define DECLRP(x)       enum fp_precision_type x;
#define SWAPRP(new, x)  x = __swapRP(new);
#define RESTRP(x)       (void) __swapRP(x);
#else
#define DECLRP(x)
#define SWAPRP(new, x)
#define RESTRP(x)
#endif

static const double
        two52 = 4503599627370496.0,
        zero = 0.0,
        one = 1.0;

double
rint(double x) {
        DECLRP(rp)
        double  t, w;
        int     ix, hx;
```



```
        ix = ((int *)&x)[HIWORD];
        hx = ix & ~0x80000000;

        if (hx >= 0x43300000)
                return (x * one);
        t = (ix < 0)? -two52 : two52;
        SWAPRP(fp_double, rp)           /* set precision mode to double */
        w = x + t;                      /* x+sign(x)*2**52 rounded */
        RESTRP(rp)                      /* restore precision mode */
        if (w == t)
                return ((ix < 0)? -zero : zero);
        return (w - t);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2082 Tue Nov  4 18:59:51 2014
new/usr/src/lib/libm/common/C/scalb.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalb = scalb
#pragma weak _scalb = scalb
#pragma weak scalb = __scalb
#pragma weak _scalb = __scalb

#include "libm.h"

double
scalb(double x, double fn) {
        int     hn, in, n;
        double  z;

        if (isnan(x) || isnan(fn))
                return (x * fn);

        in = ((int *)&fn)[HIWORD];
        hn = in & ~0x80000000;
        if (hn == 0x7ff00000)   /* fn is inf */
                return (_SVID_libm_err(x, fn, 47));

        /* see if fn is an integer without raising inexact */
        if (hn >= 0x43300000) {
                /* |fn| >= 2^52, so it must be an integer */
                n = (in < 0)? -65000 : 65000;
        } else if (hn < 0x3ff00000) {
                /* |fn| < 1, so it must be zero or non-integer */
                return ((fn == 0.0)? x : (x - x) / (x - x));
        } else if (hn < 0x41400000) {
                /* |fn| < 2^21 */
                if ((hn & ((1 << (0x413 - (hn >> 20))) - 1))
                    | ((int *)&fn)[LOWORD])
                        return ((x - x) / (x - x));
                n = (int)fn;
```



```
        } else {
                if (((int *)&fn)[LOWORD] & ((1 << (0x433 - (hn >> 20))) - 1))
                        return ((x - x) / (x - x));
                n = (in < 0)? -65000 : 65000;
        }
        z = scalbn(x, n);
        if (z != x) {
                if (z == 0.0)
                        return (_SVID_libm_err(x, fn, 33));
                if (!finite(z))
                        return (_SVID_libm_err(x, fn, 32));
        }
        return (z);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2763 Tue Nov  4 18:59:51 2014
new/usr/src/lib/libm/common/C/scalbn.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbn = scalbn
#pragma weak scalbn = __scalbn

#include "libm.h"

static const double
        one     = 1.0,
        huge    = 1.0e300,
        tiny    = 1.0e-300,
        twom54  = 5.55111512312578270211815834045410156250e-17;

#if defined(__x86)
static const double two52 = 4503599627370496.0;
#else
/*
 * Normalize non-zero subnormal x and return biased exponent of x in [-51,0]
 */
static int
ilogb_biased(unsigned *px) {
        int s = 52;
        unsigned v = px[HIWORD] & ~0x80000000, w = px[LOWORD], t = v;

        if (t)
                s -= 32;
        else
                t = w;
        if (t & 0xffff0000)
                s -= 16, t >>= 16;
        if (t & 0xff00)
                s -= 8, t >>= 8;
        if (t & 0xf0)
                s -= 4, t >>= 4;
        t <<= 1;
```



```
        s -= (0xffffaa50 >> t) & 0x3;
        if (s < 32) {
                v = (v << s) | w >> (32 - s);
                w <<= s;
        } else {
                v = w << (s - 32);
                w = 0;
        }
        px[HIWORD] = (px[HIWORD] & 0x80000000) | v;
        px[LOWORD] = w;
        return (1 - s);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1067 Tue Nov  4 18:59:52 2014
new/usr/src/lib/libm/common/C/signgam.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __signgam = signgam
#pragma weak signgam = __signgam

#include "libm_synonyms.h"
#include <math.h>

int signgam = 0;
```

```
**********************************************************
    1455 Tue Nov  4 18:59:52 2014
new/usr/src/lib/libm/common/C/significand.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __significand = significand
#pragma weak significand = __significand

#include "libm.h"

double
significand(double x) {
        int ix = ((int *) &x)[HIWORD] & ~0x80000000;

        /* weed out 0/+-Inf/NaN because C99 ilogb raises invalid on them */
        if ((ix | ((int *) &x)[LOWORD]) == 0 || ix >= 0x7ff00000)
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return ((ix & 0x80000) != 0 ? x : x + x);
                /* assumes sparc-like QNaN */
#else
                return (x + x);
#endif
        else
                return (scalbn(x, -ilogb(x)));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5187 Tue Nov  4 18:59:53 2014
new/usr/src/lib/libm/common/C/sin.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sin = sin
#pragma weak sin = __sin

/* INDENT OFF */
/*
 * sin(x)
 * Accurate Table look-up algorithm by K.C. Ng, May, 1995.
 *
 * Algorithm: see sincos.c
 */

#include "libm.h"

static const double sc[] = {
/* ONE  = */  1.0,
/* NONE = */ -1.0,
/*
 * |sin(x) - (x+pp1*x^3+pp2*x^5)| <= 2^-58.79 for |x| < 0.008
 */
/* PP1  = */ -0.166666666666631655886725205237888952148062785868305556 7,
/* PP2  = */   .008333315652997472323564894248466758248475374977974017927,
/*
 * |(sin(x) - (x+p1*x^3+...+p4*x^9)|
 * |---------------------------- | <= 2^-57.63 for |x| < 0.1953125
 * |                x             |
 */
/* P1   = */ -1.6666666666666296698052151389203015896 56e-0001,
/* P2   = */  8.3333333323909512956839934552803363766 63e-0003,
/* P3   = */ -1.9841262379979766927915517782300984039 60e-0004,
/* P4   = */  2.7534036248542772376499876228483303511 10e-0006,
/*
 * |cos(x) - (1+qq1*x^2+qq2*x^4)| <= 2^-55.99 for |x| <= 0.008 (0x3f80624d)
 */
```



```
/* QQ1  = */ -0.4999999999975492381842911981948418542742729,
/* QQ2  = */  0.0416665429043520592945452091583576403987717 40,
/* PI_H = */  3.1415926535897931159979634685,
/* PI_L    = */  1.2246467991473531772260659322 7425e-16,
/* PI_L0   = */  1.2246467991455844331128387920 5095e-16,
/* PI_L1   = */  1.7687441132271402233000052337355517376e-28,
/* PI2_H   = */  6.2831853071795862319959269370,
/* PI2_L   = */  2.4492935982947063544521318645 4850e-16,
/* PI2_L0  = */  2.4492935982911688662256775841 0190e-16,
/* PI2_L1  = */  3.5374882264542804466000104674 71034752e-28,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   10669 Tue Nov  4 18:59:53 2014
new/usr/src/lib/libm/common/C/sincos.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sincos = sincos
#pragma weak sincos = __sincos

/* INDENT OFF */
/*
 * sincos(x,s,c)
 * Accurate Table look-up algorithm by K.C. Ng, 2000.
 *
 * 1. Reduce x to x>0 by cos(-x)=cos(x), sin(-x)=-sin(x).
 * 2. For 0<= x < 8, let i = (64*x chopped)-10. Let d = x - a[i], where
 *    a[i] is a double that is close to (i+10.5)/64 (and hence |d|< 10.5/64)
 *    and such that sin(a[i]) and cos(a[i]) is close to a double (with error
 *    less than 2**-8 ulp). Then
 *
 *      cos(x) = cos(a[i]+d) = cos(a[i])cos(d) - sin(a[i])*sin(d)
 *             = TBL_cos_a[i]*(1+QQ1*d^2+QQ2*d^4) -
 *                       TBL_sin_a[i]*(d+PP1*d^3+PP2*d^5)
 *             = TBL_cos_a[i] + (TBL_cos_a[i]*d^2*(QQ1+QQ2*d^2) -
 *                       TBL_sin_a[i]*(d+PP1*d^3+PP2*d^5))
 *
 *      sin(x) = sin(a[i]+d) = sin(a[i])cos(d) + cos(a[i])*sin(d)
 *             = TBL_sin_a[i]*(1+QQ1*d^2+QQ2*d^4) +
 *                       TBL_cos_a[i]*(d+PP1*d^3+PP2*d^5)
 *             = TBL_sin_a[i] + (TBL_sin_a[i]*d^2*(QQ1+QQ2*d^2) +
 *                       TBL_cos_a[i]*(d+PP1*d^3+PP2*d^5))
 *
 *    Note: for x close to n*pi/2, special treatment is need for either
 *    sin or cos:
 *    i in [81, 100] (  pi/2 +-10.5/64 => tiny cos(x) = sin(pi/2-x)
 *    i in [181,200] (  pi   +-10.5/64 => tiny sin(x) = sin(pi-x)
 *    i in [282,301] (  3pi/2+-10.5/64 => tiny cos(x) = sin(x-3pi/2)
 *    i in [382,401] (  2pi  +-10.5/64 => tiny sin(x) = sin(x-2pi)
 *    i in [483,502] (  5pi/2+-10.5/64 => tiny cos(x) = sin(5pi/2-x)
```



```
 *
 * 3. For x >= 8.0, use kernel function __rem_pio2 to perform argument
 *    reduction and call __k_sincos_ to compute sin and cos.
 *
 * kernel function:
 *      __rem_pio2      ... argument reduction routine
 *      __k_sincos_     ... sine and cosine function on [-pi/4,pi/4]
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      TRIG(x) returns trig(x) nearly rounded (less than 1 ulp)
 */

#include "libm.h"

static const double sc[] = {
/* ONE  = */  1.0,
/* NONE = */ -1.0,
/*
 * |sin(x) - (x+pp1*x^3+pp2*x^5)| <= 2^-58.79 for |x| < 0.008
 */
/* PP1  = */ -0.16666666666631655886725205237888952148062785683055567,
/* PP2  = */   .00833331565299747232356489424846675824847537497797401792 7,
/*
 * |(sin(x) - (x+p1*x^3+...+p4*x^9)|
 * |------------------------------ | <= 2^-57.63 for |x| < 0.1953125
 * |                x              |
 */
/* P1   = */ -1.66666666666662966980521513892030158965 6e-0001,
/* P2   = */  8.33333333239095129568399345528033637666 3e-0003,
/* P3   = */ -1.98412623799797669279155177823009840396 0e-0004,
/* P4   = */  2.75340362485427723764998762284833051110 e-0006,
/*
 * |cos(x) - (1+qq1*x^2+qq2*x^4)| <= 2^-55.99 for |x| <= 0.008 (0x3f80624d)
 */
/* QQ1  = */ -0.49999999999754923818429119819484185427429,
/* QQ2  = */  0.04166654290435205929454520915835764039877 1740,
/* Q1   = */ -0.5,
/* Q2   = */  4.16666666650035070368094552086074861744 5e-0002,
/* Q3   = */ -1.38888859643697221069426629057784869600 6e-0003,
/* Q4   = */  2.47856307885858947367951951789295349219 2e-0005,
/* PIO2_H    = */  1.57079632679489655799 9,
/* PIO2_L    = */  6.12323399573676588613 0e-17,
/* PIO2_L0   = */  6.12323399572792216556 4e-17,
/* PIO2_L1   = */  8.84372056613570112025 5e-29,
/* PI_H      = */  3.14159265358979311599 7634685,
/* PI_L      = */  1.22464679914735317722 6065932274 25e-16,
/* PI_L0     = */  1.22464679914558443311 2838792050 95e-16,
```

```
 127 /* PI_L1     = */  1.76874411322714022330000523373551737 6e-28,
 128 /* PI3O2_H   = */  4.712388980384689673997,
 129 /* PI3O2_L   = */  1.836970198721029765839e-16,
 130 /* PI3O2_L0  = */  1.83697019872039613358 7e-16,
 131 /* PI3O2_L1  = */  6.336322524749201142226e-29,
 132 /* PI2_H     = */  6.283185307179586231995926937 0,
 133 /* PI2_L     = */  2.44929359829470635445213186454850e-16,
 134 /* PI2_L0    = */  2.44929359829116886622567758410190e-16,
 135 /* PI2_L1    = */  3.53748822645428044660001046747 1034752e-28,
 136 /* PI5O2_H   = */  7.853981633974482789995,
 137 /* PI5O2_L   = */  3.061616997868382943065e-16,
 138 /* PI5O2_L0  = */  3.06161699786194159886 5e-16,
 139 /* PI5O2_L1  = */  6.441344200433640781982e-28,
 140 };
______unchanged_portion_omitted_
```

```
**********************************************************
    5486 Tue Nov  4 18:59:54 2014
new/usr/src/lib/libm/common/C/sincospi.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sincospi = __sincospi

/* INDENT OFF */
/*
 * void sincospi(double x, double *s, double *c)
 * *s = sin(pi*x); *c = cos(pi*x);
 *
 * Algorithm, 10/17/2002, K.C. Ng
 * ----------------------------
 * Let y = |4x|, z = floor(y), and n = (int)(z mod 8.0) (displayed in binary).
 *      1. If y == z, then x is a multiple of pi/4. Return the following values:
 *              ------------------------------------------------------
 *                n  x mod 2    sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ------------------------------------------------------
 *               000  0.00       +0 ___       +1 ___       +0
 *               001  0.25       +\/0.5       +\/0.5       +1
 *               010  0.50       +1 ___       +0 ___       +inf
 *               011  0.75       +\/0.5       -\/0.5       -1
 *               100  1.00       -0 ___       -1 ___       +0
 *               101  1.25       -\/0.5       -\/0.5       +1
 *               110  1.50       -1 ___       -0 ___       +inf
 *               111  1.75       -\/0.5       +\/0.5       -1
 *              ------------------------------------------------------
 *      2. Otherwise,
 *              ------------------------------------------------------
 *                n     t        sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ------------------------------------------------------
 *               000  (y-z)/4     sinpi(t)     cospi(t)    tanpi(t)
 *               001  (z+1-y)/4   cospi(t)     sinpi(t)    1/tanpi(t)
 *               010  (y-z)/4     cospi(t)    -sinpi(t)   -1/tanpi(t)
 *               011  (z+1-y)/4   sinpi(t)    -cospi(t)   -tanpi(t)
 *               100  (y-z)/4    -sinpi(t)    -cospi(t)    tanpi(t)
```



```
 *               101  (z+1-y)/4  -cospi(t)    -sinpi(t)    1/tanpi(t)
 *               110  (y-z)/4    -cospi(t)     sinpi(t)   -1/tanpi(t)
 *               111  (z+1-y)/4  -sinpi(t)     cospi(t)   -tanpi(t)
 *              ------------------------------------------------------
 *
 * NOTE. This program compute sinpi/cospi(t<0.25) by __k_sin/cos(pi*t, 0.0).
 * This will return a result with error slightly more than one ulp (but less
 * than 2 ulp). If one wants accurate result,  one may break up pi*t in
 * high (tpi_h) and low (tpi_l) parts and call __k_sin/cos(tip_h, tip_lo)
 * instead.
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include "libm_macros.h"
#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif

static const double
        pi      = 3.14159265358979323846,       /* 400921FB,54442D18 */
        sqrth_h = 0.70710678118654757273731092936941422522068023681640625,
        sqrth_l = -4.83364665672645651859358442991279322134116601310004e-17;
/* INDENT ON */

void
sincospi(double x, double *s, double *c) {
        double y, z, t;
        int n, ix, k;
        int hx = ((int *) &x)[HIWORD];
        unsigned h, lx = ((unsigned *) &x)[LOWORD];

        ix = hx & ~0x80000000;
        n = (ix >> 20) - 0x3ff;
        if (n >= 51) {                  /* |x| >= 2**51 */
                if (n >= 1024)
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                        *s = *c = ix >= 0x7ff80000 ? x : x - x;
                        /* assumes sparc-like QNaN */
#else
                        *s = *c = x - x;
#endif
                else {
                        if (n >= 53)  {
                                *s = 0.0;
                                *c = 1.0;
                        }
                        else if (n == 52)  {
                                if ((lx & 1) == 0) {
                                        *s = 0.0;
                                        *c = 1.0;
                                }
                                else {
                                        *s = -0.0;
                                        *c = -1.0;
                                }
                        }
                        else {  /* n == 51 */
                                if ((lx & 1) == 0) {
                                        *s = 0.0;
                                        *c = 1.0;
                                }
                                else {
                                        *s = 1.0;
```

```
125                                     *c = 0.0;
126                             }
127                             if ((lx & 2) != 0) {
128                                     *s = -*s;
129                                     *c = -*c;
130                             }
131                     }
132             }
133         }
134         else if (n < -2)        /* |x| < 0.25 */
135                 *s = __k_sincos(pi * fabs(x), 0.0, c);
136         else {
137                 /* y = |4x|, z = floor(y), and n = (int)(z mod 8.0) */
138                 if (ix < 0x41C00000) {          /* |x| < 2**29 */
139                         y = 4.0 * fabs(x);
140                         n = (int) y;            /* exact */
141                         z = (double) n;
142                         k = z == y;
143                         t = (y - z) * 0.25;
144                 }
145                 else {                          /* 2**29 <= |x| < 2**51 */
146                         y = fabs(x);
147                         k = 50 - n;
148                         n = lx >> k;
149                         h = n << k;
150                         ((unsigned *) &z)[LOWORD] = h;
151                         ((int *) &z)[HIWORD] = ix;
152                         k = h == lx;
153                         t = y - z;
154                 }
155                 if (k) {                        /* x = N/4 */
156                         if ((n & 1) != 0)
157                                 *s = *c = sqrth_h + sqrth_l;
158                         else
159                                 if ((n & 2) == 0) {
160                                         *s = 0.0;
161                                         *c = 1.0;
162                                 }
163                                 else {
164                                         *s = 1.0;
165                                         *c = 0.0;
166                                 }
167                                 y = (n & 2) == 0 ? 0.0 : 1.0;
168                                 if ((n & 4) != 0)
169                                         *s = -*s;
170                                 if (((n + 1) & 4) != 0)
171                                         *c = -*c;
172                 }
173                 else {
174                         if ((n & 1) != 0)
175                                 t = 0.25 - t;
176                         if (((n + (n & 1)) & 2) == 0)
177                                 *s = __k_sincos(pi * t, 0.0, c);
178                         else
179                                 *c = __k_sincos(pi * t, 0.0, s);
180                                 if ((n & 4) != 0)
181                                         *s = -*s;
182                                 if (((n + 2) & 4) != 0)
183                                         *c = -*c;
184                 }
185         }
186         if (hx < 0)
187                 *s = -*s;
188 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2079 Tue Nov  4 18:59:54 2014
new/usr/src/lib/libm/common/C/sinh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinh = sinh
#pragma weak sinh = __sinh

/* INDENT OFF */
/*
 * sinh(x)
 * Code originated from 4.3bsd.
 * Modified by K.C. Ng for SUN 4.0 libm.
 * Method :
 *      1. reduce x to non-negative by sinh(-x) = - sinh(x).
 *      2.
 *                                      expm1(x) + expm1(x)/(expm1(x)+1)
 *   0 <= x <= lnovft   : sinh(x) := --------------------------------
 *                                                                    2
 *  lnovft <= x <  INF  : sinh(x) := exp(x-1024*ln2)*2**1023
 *
 *
 * Special cases:
 *      sinh(x) is x if x is +INF, -INF, or NaN.
 *      only sinh(0)=0 is exact for finite argument.
 *
 */
/* INDENT ON */

#include "libm.h"

static const double
        ln2hi = 6.93147180369123816490e-01,
        ln2lo = 1.90821492927058770002e-10,
        lnovft = 7.09782712893383973096e+02;

double
```



```
sinh(double x) {
        double  ox, r, t;

        ox = x;
        r = fabs(x);
        if (!finite(x))
                return (x * r);
        if (r < lnovft) {
                t = expm1(r);
                r = copysign((t + t / (1.0 + t)) * 0.5, x);
        } else {
                if (r < 1000.0)
                        x = copysign(exp((r - 1024 * ln2hi) - 1024 * ln2lo), x);
                r = scalbn(x, 1023);
        }
        if (!finite(r))
                r = _SVID_libm_err(ox, ox, 25);
        return (r);
}
```

_____***unchanged_portion_omitted_***

****
1220 Tue Nov 4 18:59:55 2014
new/usr/src/lib/libm/common/C/sqrt.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sqrt = sqrt
#pragma weak sqrt = __sqrt

#include "libm.h"


extern double __inline_sqrt(double);

double
sqrt(double x) {
        double  z = __inline_sqrt(x);

        if (isnan(x))
                return (z);
        return ((x < 0.0)? _SVID_libm_err(x, x, 26) : z);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1852 Tue Nov  4 18:59:55 2014
new/usr/src/lib/libm/common/C/tan.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tan = tan
#pragma weak tan = __tan

/* INDENT OFF */
/*
 * tan(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_tan         ... tangent function on [-pi/4,pi/4]
 *      __rem_pio2      ... argument reduction routine
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>

double
tan(double x) {
        double y[2], z = 0.0;
        int n, ix;

        /* high word of x */
        ix = ((int *) &x)[HIWORD];

        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3fe921fb)
                return (__k_tan(x, z, 0));
```



```
        /* tan(Inf or NaN) is NaN */
        else if (ix >= 0x7ff00000) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (ix >= 0x7ff80000 ? x : x - x);  /* NaN */
                /* assumes sparc-like QNaN */
#else
                return (x - x);                         /* NaN */
#endif
        }

        /* argument reduction needed */
        else {
                n = __rem_pio2(x, y);
                return (__k_tan(y[0], y[1], n & 1));
        }
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2494 Tue Nov  4 18:59:55 2014
new/usr/src/lib/libm/common/C/tanh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tanh = tanh
#pragma weak tanh = __tanh

/* INDENT OFF */
/*
 * TANH(X)
 * RETURN THE HYPERBOLIC TANGENT OF X
 * code based on 4.3bsd
 * Modified by K.C. Ng for sun 4.0, Jan 31, 1987
 *
 * Method :
 *      1. reduce x to non-negative by tanh(-x) = - tanh(x).
 *      2.
 *          0      <  x <=  1.e-10 :  tanh(x) := x
 *                                                -expm1(-2x)
 *          1.e-10 <  x <=  1      :  tanh(x) := --------------
 *                                               expm1(-2x) + 2
 *                                                     2
 *          1      <= x <=  22.0   :  tanh(x) := 1 -  ---------------
 *                                                    expm1(2x) + 2
 *          22.0   <  x <= INF     :  tanh(x) := 1.
 *
 *      Note: 22 was chosen so that fl(1.0+2/(expm1(2*22)+2)) == 1.
 *
 * Special cases:
 *      tanh(NaN) is NaN;
 *      only tanh(0)=0 is exact for finite argument.
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
```



```
#include <math.h>

static const double
        one = 1.0,
        two = 2.0,
        small = 1.0e-10,
        big = 1.0e10;
/* INDENT ON */

double
tanh(double x) {
        double t, y, z;
        int signx;
        volatile double dummy;

        if (isnan(x))
                return (x * x); /* + -> * for Cheetah */
        signx = signbit(x);
        t = fabs(x);
        z = one;
        if (t <= 22.0) {
                if (t > one)
                        z = one - two / (expm1(t + t) + two);
                else if (t > small) {
                        y = expm1(-t - t);
                        z = -y / (y + two);
                } else {
                        /* raise the INEXACT flag for non-zero t */
                        dummy = t + big;
#ifdef lint
                        dummy = dummy;
#endif
                        return (x);
                }
        } else if (!finite(t))
                return (copysign(1.0, x));
        else
                return (signx == 1 ? -z + small * small : z - small * small);

        return (signx == 1 ? -z : z);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
   14533 Tue Nov  4 18:59:56 2014
new/usr/src/lib/libm/common/LD/__lgammal.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double __k_lgammal(long double x, int *signgamlp);
 * K.C. Ng, August, 1989.
 *
 * We choose [1.5,2.5] to be the primary interval. Our algorithms
 * are mainly derived from
 *
 *
 *                              zeta(2)-1    2    zeta(3)-1    3
 * lgamma(2+s) = s*(1-euler) + --------- * s  -  --------- * s  + ...
 *                                 2               3
 *
 * Note 1. Since gamma(1+s)=s*gamma(s), hence
 *              lgamma(1+s) = log(s) + lgamma(s), or
 *              lgamma(s) = lgamma(1+s) - log(s).
 *         When s is really tiny (like roundoff), lgamma(1+s) ~ s(1-enler)
 *         Hence lgamma(s) ~ -log(s) for tiny s
 *
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

static long double neg(long double, int *);
static long double poly(long double, const long double *, int);
static long double polytail(long double);
static long double primary(long double);

static const long double
c0 =     0.0L,
```



```
ch =     0.5L,
c1 =     1.0L,
c2 =     2.0L,
c3 =     3.0L,
c4 =     4.0L,
c5 =     5.0L,
c6 =     6.0L,
pi =     3.14159265358979323846264338327950288419 71L,
tiny =   1.0e-40L;

long double
__k_lgammal(long double x, int *signgamlp) {
        long double t, y;
        int i;

        /* purge off +-inf, NaN and negative arguments */
        if (!finitel(x))
                return (x*x);
        *signgamlp = 1;
        if (signbitl(x))
                return (neg(x, signgamlp));

        /* for x < 8.0 */
        if (x < 8.0L) {
            y = anintl(x);
            i = (int) y;
            switch (i) {
            case 0:
                if (x < 1.0e-40L)
                        return (-logl(x));
                else
                        return (primary(x)-log1pl(x))-logl(x);
            case 1:
                return (primary(x-y)-logl(x));
            case 2:
                return (primary(x-y));
            case 3:
                return (primary(x-y)+logl(x-c1));
            case 4:
                return (primary(x-y)+logl((x-c1)*(x-c2)));
            case 5:
                return (primary(x-y)+logl((x-c1)*(x-c2)*(x-c3)));
            case 6:
                return (primary(x-y)+logl((x-c1)*(x-c2)*(x-c3)*(x-c4)));
            case 7:
                return (primary(x-y)+logl((x-c1)*(x-c2)*(x-c3)*(x-c4)*(x-c5)));
            case 8:
                return primary(x-y)+
                        logl((x-c1)*(x-c2)*(x-c3)*(x-c4)*(x-c5)*(x-c6));
            }
        }

        /* 8.0 <= x < 1.0e40 */
        if (x < 1.0e40L) {
            t = logl(x);
            return (x*(t-c1)-(ch*t-polytail(c1/x)));
        }

        /* 1.0e40 <= x <= inf */
        return (x*(logl(x)-c1));
}
```

_____*unchanged_portion_omitted_*

****
1162 Tue Nov 4 18:59:56 2014
new/usr/src/lib/libm/common/LD/__poly_libmq.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "libm.h"
#include "libm_synonyms.h"

long double __poly_libmq(x,n,p)
long double x,p[];
int n;
{
        long double t; int i;
        t = p[n-1];
        for(i=n-2;i>=0;i--) t = p[i] + x*t;
        return t;
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1471 Tue Nov  4 18:59:57 2014
new/usr/src/lib/libm/common/LD/acoshl.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __acoshl = acoshl
#pragma weak acoshl = __acoshl

#include "libm.h"

static const long double
        zero    = 0.0L,
        ln2     = 6.93147180559945309417232121458176568075e-0001L,
        one     = 1.0L,
        big     = 1.e+20L;

long double
acoshl(long double x) {
        long double t;

        if (isnanl(x))
                return (x + x);
        else if (x > big)
                return (logl(x) + ln2);
        else if (x > one) {
                t = sqrtl(x - one);
                return (log1pl(t * (t + sqrtl(x + one))));
        } else if (x == one)
                return (zero);
        else
                return ((x - x) / (x - x));
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    1610 Tue Nov  4 18:59:57 2014
new/usr/src/lib/libm/common/LD/asinhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinhl = asinhl
#pragma weak asinhl = __asinhl

#include "libm.h"

static const long double
        ln2     = 6.93147180559945309417232121458176568075 5e-0001L,
        one     = 1.0L,
        big     = 1.0e+20L,
        tiny    = 1.0e-20L;

long double
asinhl(long double x) {
        long double t, w;
#ifndef lint
        volatile long double dummy;
#endif

        w = fabsl(x);
        if (isnanl(x))
                return (x + x); /* x is NaN */
        if (w < tiny) {
#ifndef lint
                dummy = x + big;        /* inexact if x != 0 */
#endif
                return (x);     /* tiny x */
        } else if (w < big) {
                t = one / w;
                return (copysignl(log1pl(w + w / (t + sqrtl(one + t * t))), x));
        } else
                return (copysignl(logl(w) + ln2, x));
}
```

```
**********************************************************
    1321 Tue Nov  4 18:59:58 2014
new/usr/src/lib/libm/common/LD/atan2pil.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak atan2pil = __atan2pil

#include "libm.h"
#include "libm_synonyms.h"

#define GENERIC long double
#define ATAN2PI atan2pil
#define ATAN2   atan2l

/* ATAN2PI(y,x)
 *
 *      ATAN2PI(y,x) = ATAN2(y,x)/pi
 */

extern GENERIC  ATAN2();

static GENERIC
invpi = (GENERIC) 3.183098861837906715377675267450287240689e-0001L;

GENERIC ATAN2PI(y,x)
GENERIC y,x;
{
        return ATAN2(y,x)*invpi;
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1979 Tue Nov  4 18:59:58 2014
new/usr/src/lib/libm/common/LD/atanhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanhl = atanhl
#pragma weak atanhl = __atanhl

#include "libm.h"
#include "libm_synonyms.h"

#define GENERIC long double
#define ATANH   atanhl

/* ATANH(x)
 *                    1             2x                          x
 *      ATANH(x) = --- * LOG(1 + -------) = 0.5 * LOG1P(2 * --------)
 *                    2           1 - x                       1 - x
 * Note: to guarantee ATANH(-x) = -ATANH(x), we use
 *                  sign(x)             |x|
 *      ATANH(x) = ------- * LOG1P(2*-------).
 *                    2               1 - |x|
 *
 * Special cases:
 *      ATANH(x) is NaN if |x| > 1 with signal;
 *      ATANH(NaN) is that NaN with no signal;
 *      ATANH(+-1) is +-INF with signal.
 *
 */

#define FABS    fabsl
#define LOG1P   log1pl
#define COPYSIGN        copysignl


extern GENERIC  FABS(),LOG1P(),COPYSIGN();
```



```
static GENERIC
zero    = (GENERIC) 0.0,
half    = (GENERIC) 0.5,
one     = (GENERIC) 1.0;

GENERIC ATANH(x)
GENERIC x;
{
        GENERIC t;
        t = FABS(x);
        if (t == one) return x/zero;
        t = t/(one-t);
        return COPYSIGN(half,x)*LOG1P(t+t);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1752 Tue Nov  4 18:59:59 2014
new/usr/src/lib/libm/common/LD/cbrtl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cbrtl = cbrtl
#pragma weak cbrtl = __cbrtl

#include "libm.h"
#include "longdouble.h"

static const double d_one = 1.0;

long double
cbrtl(long double x) {
        long double s, t, r, w, y;
        double dx, dy;
        int *py = (int *) &dy;
        int n, m, m3, n0, sx;

        if (!finitel(x))
                return (x + x);
        if (iszerol(x))
                return (x);
        n0 = 0;
        if (*((int *) &d_one) == 0)
                n0 = 1;
        sx = signbitl(x);
        x = fabsl(x);
        n = ilogbl(x);
        m = n / 3;
        m3 = m + m + m;
        y = scalbnl(x, -m3);
        dx = (double) y;
        dy = cbrt(dx);
        py[1 - n0] += 2;
        if (py[1 - n0] == 0)
                py[n0] += 1;

        /* one step newton iteration to 113 bits with error < 0.667ulps */
        t = (long double) dy;
        t = scalbnl(t, m);
        s = t * t;
        r = x / s;
        w = t + t;
        r = (r - t) / (w + r);
        t += t * r;

        return (sx == 0 ? t : -t);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2788 Tue Nov  4 18:59:59 2014
new/usr/src/lib/libm/common/LD/coshl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __coshl = coshl
#pragma weak coshl = __coshl

#include "libm.h"
#include "longdouble.h"

/*
 * COSH(X)
 * RETURN THE HYPERBOLIC COSINE OF X
 *
 * Method :
 *      1. Replace x by |x| (COSH(x) = COSH(-x)).
 *      2.
 *                                                  [ EXP(x) - 1 ]^2
 *          0       <= x <= 0.3465  :  COSH(x) := 1 + -------------------
 *                                                     2*EXP(x)
 *
 *                                             EXP(x) +  1/EXP(x)
 *          0.3465  <= x <= thresh  :  COSH(x) := -------------------
 *                                                2
 *          thresh  <= x <= lnovft  :  COSH(x) := EXP(x)/2
 *          lnovft  <= x <  INF     :  COSH(x) := SCALBN(EXP(x-MEP1*ln2),ME)
 *
 *
 * here
 *      0.3465          a number that is near one half of ln2.
 *      thresh          a number such that
 *                              EXP(thresh)+EXP(-thresh)=EXP(thresh)
 *      lnovft          logarithm of the overflow threshold
 *                      = MEP1*ln2 chopped to machine precision.
 *      ME              maximum exponent
 *      MEP1            maximum exponent plus 1
```



```
 *
 * Special cases:
 *      COSH(x) is |x| if x is +INF, -INF, or NaN.
 *      only COSH(0)=1 is exact for finite x.
 */

static const long double C[] = {
        0.5L,
        1.0L,
        0.3465L,
        45.0L,
        1.13565234062941439487914 9e+04L,
        7.00444768624254908785898 5e-16L,
        2.71050543121376108501863 2e-20L,               /* 2^-65 */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    2789 Tue Nov  4 19:00:00 2014
new/usr/src/lib/libm/common/LD/cosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cosl = cosl
#pragma weak cosl = __cosl

/* INDENT OFF */
/* cosl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sinl        ... sin function on [-pi/4,pi/4]
 *      __k_cosl        ... cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
```



```
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

long double
cosl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;
        int *px = (int *) &x;

        /* trig(Inf or NaN) is NaN */
        if (!finitel(x))
                return x - x;

        /* High word of x. */
#if defined(__i386) || defined(__amd64)
        XTOI(px, ix);
#else
        ix = px[0];
#endif

        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)
                return __k_cosl(x, z);

        /* argument reduction needed */
        else {
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                case 0:
                        return __k_cosl(y[0], y[1]);
                case 1:
                        return -__k_sinl(y[0], y[1]);
                case 2:
                        return -__k_cosl(y[0], y[1]);
                case 3:
                        return __k_sinl(y[0], y[1]);
                /* NOTREACHED */
                }
        }
    return 0.0L;
}
```

*unchanged_portion_omitted_*

```
**********************************************************
   12826 Tue Nov  4 19:00:00 2014
new/usr/src/lib/libm/common/LD/erfl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/* long double function erf,erfc (long double x)
 * K.C. Ng, September, 1989.
 *                           x
 *                    2      |\
 *     erf(x)  =  ---------  | exp(-t*t)dt
 *                 sqrt(pi) \|
 *                           0
 *
 *     erfc(x) =  1-erf(x)
 *
 * method:
 *      Since erf(-x) = -erf(x), we assume x>=0.
 *      For x near 0, we have the expansion
 *
 *          erf(x) = (2/sqrt(pi))*(x - x^3/3 + x^5/10 - x^7/42 + ....).
 *
 *      Since 2/sqrt(pi) = 1.128379167095512573896158903121545171688,
 *      we use x + x*P(x^2) to approximate erf(x). This formula will
 *      guarantee the error less than one ulp where x is not too far
 *      away from 0. We note that erf(x)=x at x = 0.6174..... After
 *      some experiment, we choose the following approximation on
 *      interval [0,0.84375].
 *
 *      For x in [0,0.84375]
 *                 2               2        4              40
 *         P =  P(x ) = (p0 + p1 * x + p2 * x + ... + p20 * x  )
 *
 *         erf(x)  = x + x*P
 *         erfc(x) = 1 - erf(x)          if x<=0.25
 *                 = 0.5 + ((0.5-x)-x*P)  if x in [0.25,0.84375]
 *      precision: |P(x^2)-(erf(x)-x)/x| <= 2**-122.50
 *
```



```
 *      For x in [0.84375,1.25], let s = x - 1, and
 *      c = 0.84506291151 rounded to single (24 bits)
 *         erf(x)  = c  + P1(s)/Q1(s)
 *         erfc(x) = (1-c)  - P1(s)/Q1(s)
 *      precision: |P1/Q1 - (erf(x)-c)| <= 2**-118.41
 *
 *
 *      For x in [1.25,1.75], let s = x - 1.5, and
 *      c = 0.95478588343 rounded to single (24 bits)
 *         erf(x)  = c  + P2(s)/Q2(s)
 *         erfc(x) = (1-c)  - P2(s)/Q2(s)
 *      precision: |P1/Q1 - (erf(x)-c)| <= 2**-123.83
 *
 *
 *      For x in [1.75,16/3]
 *         erfc(x) = exp(-x*x)*(1/x)*R1(1/x)/S1(1/x)
 *         erf(x)  = 1 - erfc(x)
 *      precision: absolute error of R1/S1 is bounded by 2**-124.03
 *
 *      For x in [16/3,107]
 *         erfc(x) = exp(-x*x)*(1/x)*R2(1/x)/S2(1/x)
 *         erf(x)  = 1 - erfc(x) (if x>=9 simple return erf(x)=1 with inexact)
 *      precision: absolute error of R2/S2 is bounded by 2**-120.07
 *
 *      Else if inf > x >= 107
 *         erf(x)  = 1 with inexact
 *         erfc(x) = 0 with underflow
 *
 *      Special case:
 *         erf(inf)  = 1
 *         erfc(inf) = 0
 */

#pragma weak __erfl = erfl
#pragma weak __erfcl = erfcl
#pragma weak erfl = __erfl
#pragma weak erfcl = __erfcl

#include "libm.h"
#include "longdouble.h"

static long double
tiny        = 1e-40L,
nearunfl    = 1e-4000L,
half        = 0.5L,
one         = 1.0L,
onehalf     = 1.5L,
L16_3       = 16.0L/3.0L;
/*
 * Coefficients for even polynomial P for erf(x)=x+x*P(x^2) on [0,0.84375]
 */
static long double P[] = {      /* 21 coeffs */
   1.28379167095512573896158903121545171556e-0001L,
  -3.76126389031837524632052967707181559460 3e-0001L,
   1.12837916709551257389615890312120589913 5e-0001L,
  -2.68661706451312517594323548334462504609 2e-0002L,
   5.22397762544218784211184665298045456838 9e-0003L,
  -8.54832702345085283254662627108386272435 8e-0004L,
   1.20553329817896642510216471590223197667 2e-0004L,
  -1.49256503584062509767494490502789783899 6e-0005L,
   1.64621143658892473360464884917293669202 4e-0006L,
  -1.63658446912349197681583470479973351498 7e-0007L,
   1.48071928158789744530252900714477073930 5e-0008L,
  -1.22905553017078284304646798646472204717 5e-0009L,
   9.42275906432030735755395494576065434163 3e-0011L,
  -6.71136684665343903616210510499143338092 6e-0012L,
```

```
 126    4.463224090341893165100275380693843116240e-0013L,
 127   -2.783513452582658245422635662559779162312e-0014L,
 128    1.634227412586960195251346878863754661546e-0015L,
 129   -9.060782672889577722765711455623117802795e-0017L,
 130    4.741341801266246873412159213893613602354e-0018L,
 131   -2.272417596497826188374846636534317381203e-0019L,
 132    8.069088733716068462496835658928566920933e-0021L,
 133 };
______unchanged_portion_omitted_
```

****
1412 Tue Nov 4 19:00:00 2014
new/usr/src/lib/libm/common/LD/finitel.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak finitel = __finitel

#include "libm.h"

#if defined(__sparc)
int
finitel(long double x) {
        int *px = (int *) &x;
        return ((px[0] & ~0x80000000) < 0x7fff0000);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1319 Tue Nov  4 19:00:01 2014
new/usr/src/lib/libm/common/LD/gammal.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __gammal = gammal
#pragma weak gammal = __gammal

/*
 * long double gammal(long double x);
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

extern int signgam;
extern int signgaml;

long double
gammal(long double x) {
        long double y = __k_lgammal(x, &signgaml);

        signgam = signgaml;     /* SUSv3 requires the setting of signgam */
        return y;
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1231 Tue Nov  4 19:00:01 2014
new/usr/src/lib/libm/common/LD/gammal_r.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double gammal_r(long double x, int *signgamlp);
 */

#pragma weak __gammal_r = gammal_r
#pragma weak gammal_r = __gammal_r

#include "libm.h"
#include "longdouble.h"

long double
gammal_r(long double x, int *signgamlp) {
        return __k_lgammal(x, signgamlp);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    3742 Tue Nov  4 19:00:02 2014
new/usr/src/lib/libm/common/LD/hypotl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __hypotl = hypotl
#pragma weak hypotl = __hypotl

/*
 * hypotl(x,y)
 * Method :
 *      If z=x*x+y*y has error less than sqrt(2)/2 ulp than sqrt(z) has
 *      error less than 1 ulp.
 *      So, compute sqrt(x*x+y*y) with some care as follows:
 *      Assume x>y>0;
 *      1. save and set rounding to round-to-nearest
 *      2. if x > 2y  use
 *              x1*x1+(y*y+(x2*(x+x2))) for x*x+y*y
 *      where x1 = x with lower 32 bits cleared, x2 = x-x1; else
 *      3. if x <= 2y use
 *              t1*y1+((x-y)*(x-y)+(t1*y2+t2*y))
 *      where t1 = 2x with lower 64 bits cleared, t2 = 2x-t1, y1= y with
 *      lower 32 bits cleared, y2 = y-y1.
 *
 *      NOTE: DO NOT remove parenthsis!
 *
 * Special cases:
 *      hypot(x,y) is INF if x or y is +INF or -INF; else
 *      hypot(x,y) is NAN if x or y is NAN.
 *
 * Accuracy:
 *      hypot(x,y) returns sqrt(x^2+y^2) with error less than 1 ulps (units
 *      in the last place)
 */

#include "libm.h"
```



```
#if defined(__x86)
extern enum fp_direction_type __swap87RD(enum fp_direction_type);

#define k       0x7fff

long double
hypotl(long double x, long double y) {
        long double t1, t2, y1, y2, w;
        int *px = (int *) &x, *py = (int *) &y;
        int *pt1 = (int *) &t1, *py1 = (int *) &y1;
        enum fp_direction_type rd;
        int j, nx, ny, nz;

        px[2] &= 0x7fff;        /* clear sign bit and padding bits of x and y */
        py[2] &= 0x7fff;
        nx = px[2];             /* biased exponent of x and y */
        ny = py[2];
        if (ny > nx) {
                w = x;
                x = y;
                y = w;
                nz = ny;
                ny = nx;
                nx = nz;
        }                       /* force nx >= ny */
        if (nx - ny >= 66)
                return (x + y); /* x / y >= 2**65 */
        if (nx < 0x5ff3 && ny > 0x205b) {       /* medium x,y */
                /* save and set RD to Rounding to nearest */
                rd = __swap87RD(fp_nearest);
                w = x - y;
                if (w > y) {
                        pt1[2] = px[2];
                        pt1[1] = px[1];
                        pt1[0] = 0;
                        t2 = x - t1;
                        x = sqrtl(t1 * t1 - (y * (-y) - t2 * (x + t1)));
                } else {
                        x += x;
                        py1[2] = py[2];
                        py1[1] = py[1];
                        py1[0] = 0;
                        y2 = y - y1;
                        pt1[2] = px[2];
                        pt1[1] = px[1];
                        pt1[0] = 0;
                        t2 = x - t1;
                        x = sqrtl(t1 * y1 - (w * (-w) - (t2 * y1 + y2 * x)));
                }
                if (rd != fp_nearest)
                        __swap87RD(rd); /* restore rounding mode */
                return (x);
        } else {
                if (nx == k || ny == k) {       /* x or y is INF or NaN */
                        /* since nx >= ny; nx is always k within this block */
                        if (px[1] == 0x80000000 && px[0] == 0)
                                return (x);
                        else if (ny == k && py[1] == 0x80000000 && py[0] == 0)
                                return (y);
                        else
                                return (x + y);
                }
                if (ny == 0) {
                        if (y == 0.L || x == 0.L)
                                return (x + y);
                        pt1[2] = 0x3fff + 16381;
```

```
127                         pt1[1] = 0x80000000;
128                         pt1[0] = 0;
129                         py1[2] = 0x3fff - 16381;
130                         py1[1] = 0x80000000;
131                         py1[0] = 0;
132                         x *= t1;
133                         y *= t1;
134                         return (y1 * hypotl(x, y));
135                 }
136                 j = nx - 0x3fff;
137                 px[2] -= j;
138                 py[2] -= j;
139                 pt1[2] = nx;
140                 pt1[1] = 0x80000000;
141                 pt1[0] = 0;
142                 return (t1 * hypotl(x, y));
143         }
144 }
```

*unchanged_portion_omitted_*

****
1585 Tue Nov 4 19:00:02 2014
new/usr/src/lib/libm/common/LD/isnanl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __isnanl = isnanl
#pragma weak isnanl = __isnanl

#include "libm.h"

#if defined(__sparc)
int
isnanl(long double x) {
        int *px = (int *) &x;
        return ((px[0] & ~0x80000000) >= 0x7fff0000 &&
                ((px[0] & ~0xffff0000) | px[1] | px[2] | px[3]) != 0);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   27917 Tue Nov  4 19:00:02 2014
new/usr/src/lib/libm/common/LD/j0l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Floating point Bessel's function of the first and second kinds
 * of order zero: j0(x),y0(x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 */

#pragma weak __j0l = j0l
#pragma weak __y0l = y0l
#pragma weak j0l = __j0l
#pragma weak y0l = __y0l

#include "libm.h"
#include "libm_synonyms.h"

#include "longdouble.h"

#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif

#define GENERIC long double
static GENERIC
zero    = 0.0L,
small   = 1.0e-9L,
tiny    = 1.0e-38L,
one     = 1.0L,
five    = 5.0L,
eight   = 8.0L,
```



```
invsqrtpi = 5.64189583547756286948079451560772585844l-0001L,
tpi     = 0.63661977236758134307553505349005744l8L;

static GENERIC pzero(), qzero();
static GENERIC r0[7] = {
  -2.49999999999999999999999999999998934492e-0001L,
   1.27265792736004978632761845113376371488e-0002L,
  -2.69449976371296327690063669340065960089e-0004L,
   2.72487747505897757690323407091961644788e-0006L,
  -1.43261710321433023696747749539307632028e-0008L,
   3.82324880408007916870668354051379222447e-0011L,
  -4.18317427756798364733756850428631366506e-0014L,
};
______unchanged_portion_omitted_
```

```
**********************************************************
   28271 Tue Nov  4 19:00:03 2014
new/usr/src/lib/libm/common/LD/j1l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * floating point Bessel's function of the first and second kinds
 * of order zero: j1(x),y1(x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 */

#pragma weak __j1l = j1l
#pragma weak __y1l = y1l
#pragma weak j1l = __j1l
#pragma weak y1l = __y1l

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"
#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif

#define GENERIC long double
static GENERIC
zero    = 0.0L,
small   = 1.0e-9L,
tiny    = 1.0e-38L,
one     = 1.0L,
five    = 5.0L,
invsqrtpi = 5.64189583547756286948079451560772585844l1e-0001L,
tpi     = 0.63661977236758134307553505349005744l8L;
```



```
static GENERIC pone(), qone();
static GENERIC r0[7] = {
   -6.24999999999999999999999999999999627320e-0002L,
    1.94060672719404171620538461849464l1565464e-0003L,
   -3.00563042315573370185648146998645904l3883e-0005L,
    2.34558621940391866746834104736957216l358e-0007L,
   -9.97680928588525358752901010913333666l9724e-0010L,
    2.21874325836362394607895878377510747l381e-0012L,
   -2.07107965621870060476765092410357804l6280e-0015L,
};
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    6885 Tue Nov  4 19:00:04 2014
new/usr/src/lib/libm/common/LD/jnl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __jnl = jnl
#pragma weak __ynl = ynl
#pragma weak jnl = __jnl
#pragma weak ynl = __ynl

/*
 * floating point Bessel's function of the 1st and 2nd kind
 * of order n: jn(n,x),yn(n,x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 * Note 2. About jn(n,x), yn(n,x)
 *      For n=0, j0(x) is called,
 *      for n=1, j1(x) is called,
 *      for n<x, forward recursion us used starting
 *      from values of j0(x) and j1(x).
 *      for n>x, a continued fraction approximation to
 *      j(n,x)/j(n-1,x) is evaluated and then backward
 *      recursion is used starting from a supposed value
 *      for j(n,x). The resulting value of j(0,x) is
 *      compared with the actual value to correct the
 *      supposed value of j(n,x).
 *
 *      yn(n,x) is similar in all respects, except
 *      that forward recursion is used for all
 *      values of n>1.
 *
 */

#include "libm.h"
#include "longdouble.h"
```



```
#include <float.h>      /* LDBL_MAX */

#define GENERIC long double

static const GENERIC
invsqrtpi = 5.641895835477562869480794515607725858441e-0001L,
two  = 2.0L,
zero = 0.0L,
one  = 1.0L;

GENERIC
jnl(n, x) int n; GENERIC x; {
        int i, sgn;
        GENERIC a, b, temp = 0, z, w;

        /*
         * J(-n,x) = (-1)^n * J(n, x), J(n, -x) = (-1)^n * J(n, x)
         * Thus, J(-n,x) = J(n,-x)
         */
        if (n < 0) {
                n = -n;
                x = -x;
        }
        if (n == 0) return (j0l(x));
        if (n == 1) return (j1l(x));
        if (x != x) return x+x;
        if ((n&1) == 0)
                sgn = 0;                        /* even n */
        else
                sgn = signbitl(x);      /* old n  */
        x = fabsl(x);
        if (x == zero || !finitel(x)) b = zero;
        else if ((GENERIC)n <= x) {
                        /*
                         * Safe to use
                         * J(n+1,x)=2n/x *J(n,x)-J(n-1,x)
                         */
            if (x > 1.0e91L) {
                                /*
                                 * x >> n**2
                                 *  Jn(x) = cos(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *  Yn(x) = sin(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *  Let s=sin(x), c=cos(x),
                                 *  xn=x-(2n+1)*pi/4, sqt2 = sqrt(2),then
                                 *
                                 *         n    sin(xn)*sqt2    cos(xn)*sqt2
                                 *      ----------------------------------
                                 *         0     s-c             c+s
                                 *         1    -s-c            -c+s
                                 *         2    -s+c            -c-s
                                 *         3     s+c             c-s
                                 */
                switch (n&3) {
                    case 0: temp =  cosl(x)+sinl(x); break;
                    case 1: temp = -cosl(x)+sinl(x); break;
                    case 2: temp = -cosl(x)-sinl(x); break;
                    case 3: temp =  cosl(x)-sinl(x); break;
                }
                b = invsqrtpi*temp/sqrtl(x);
            } else {
                        a = j0l(x);
                        b = j1l(x);
                        for (i = 1; i < n; i++) {
                    temp = b;
                    b = b*((GENERIC)(i+i)/x) - a; /* avoid underflow */
                    a = temp;
```

```
126                         }
127             }
128         } else {
129             if (x < 1e-17L) {   /* use J(n,x) = 1/n!*(x/2)^n */
130                 b = powl(0.5L*x, (GENERIC) n);
131                 if (b != zero) {
132                     for (a = one, i = 1; i <= n; i++) a *= (GENERIC)i;
133                     b = b/a;
134                 }
135             } else {
136                 /*
137                  * use backward recurrence
138                  *                          x      x^2      x^2
139                  *  J(n,x)/J(n-1,x) =  ----   ------   ------   .....
140                  *                          2n  - 2(n+1) - 2(n+2)
141                  *
142                  *                          1      1        1
143                  *  (for large x)   =  ----  ------   ------   .....
144                  *                          2n   2(n+1)   2(n+2)
145                  *                          -- - ------ - ------ -
146                  *                           x     x         x
147                  *
148                  * Let w = 2n/x and h=2/x, then the above quotient
149                  * is equal to the continued fraction:
150                  *                     1
151                  *      = -----------------------
152                  *                    1
153                  *         w - -----------------
154                  *                      1
155                  *              w+h - ---------
156                  *                    w+2h - ...
157                  *
158                  * To determine how many terms needed, let
159                  * Q(0) = w, Q(1) = w(w+h) - 1,
160                  * Q(k) = (w+k*h)*Q(k-1) - Q(k-2),
161                  * When Q(k) > 1e4      good for single
162                  * When Q(k) > 1e9      good for double
163                  * When Q(k) > 1e17     good for quaduple
164                  */
165             /* determin k */
166                 GENERIC t, v;
167                 double q0, q1, h, tmp; int k, m;
168                 w  = (n+n)/(double)x; h = 2.0/(double)x;
169                 q0 = w;  z = w+h; q1 = w*z - 1.0; k = 1;
170                 while (q1 < 1.0e17) {
171                         k += 1; z += h;
172                         tmp = z*q1 - q0;
173                         q0 = q1;
174                         q1 = tmp;
175                 }
176                 m = n+n;
177                 for (t = zero, i = 2*(n+k); i >= m; i -= 2) t = one/(i/x-t);
178                 a = t;
179                 b = one;
180                         /*
181                          * Estimate log((2/x)^n*n!) = n*log(2/x)+n*ln(n)
182                          * hence, if n*(log(2n/x)) > ...
183                          * single 8.8722839355e+01
184                          * double 7.09782712893383973096e+02
185                          * long double 1.135652340629414394949193107797076500617
186                          * then recurrent value may overflow and the result is
187                          * likely underflow to zero.
188                          */
189                 tmp = n;
190                 v = two/x;
191                 tmp = tmp*logl(fabsl(v*tmp));
```



```
192                 if (tmp < 1.135652340629414394949193107797076500e+04L) {
193                         for (i = n-1; i > 0; i--) {
194                         temp = b;
195                         b = ((i+i)/x)*b - a;
196                         a = temp;
197                         }
198                 } else {
199                         for (i = n-1; i > 0; i--) {
200                         temp = b;
201                         b = ((i+i)/x)*b - a;
202                         a = temp;
203                         if (b > 1e1000L) {
204                                 a /= b;
205                                 t /= b;
206                                 b  = 1.0;
207                                 }
208                         }
209                 }
210                         b = (t*j0l(x)/b);
211             }
212         }
213         if (sgn == 1)
214                 return -b;
215         else
216                 return b;
217 }
```

*_____unchanged_portion_omitted_*

****
1323 Tue Nov 4 19:00:04 2014
new/usr/src/lib/libm/common/LD/lgammal.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgammal = lgammal
#pragma weak lgammal = __lgammal

/*
 * long double lgammal(long double x);
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

extern int signgam;
extern int signgaml;

long double
lgammal(long double x) {
        long double y = __k_lgammal(x, &signgaml);

        signgam = signgaml;     /* SUSv3 requires the setting of signgam */
        return y;
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1235 Tue Nov  4 19:00:05 2014
new/usr/src/lib/libm/common/LD/lgammal_r.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double lgammal_r(long double x, int *signgamlp);
 */

#pragma weak __lgammal_r = lgammal_r
#pragma weak lgammal_r = __lgammal_r

#include "libm.h"
#include "longdouble.h"

long double
lgammal_r(long double x, int *signgamlp) {
        return __k_lgammal(x, signgamlp);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1592 Tue Nov  4 19:00:05 2014
new/usr/src/lib/libm/common/LD/log1pl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log1pl = log1pl
#pragma weak log1pl = __log1pl

/*
 * log1pl(x)
 * Kahan's trick based on log(1+x)/x being a slow varying function.
 */

#include "libm.h"

#if defined(__x86)
#define __swapRD        __swap87RD
#endif
extern enum fp_direction_type __swapRD(enum fp_direction_type);

long double
log1pl(long double x) {
        long double y;
        enum fp_direction_type rd;

        if (x != x)
                return (x + x);
        if (x < -1.L)
                return (logl(x));
        rd = __swapRD(fp_nearest);
        y = 1.L + x;
        if (y != 1.L) {
                if (y == x)
                        x = logl(x);
                else
                        x *= logl(y) / (y - 1.L);
        }
```



```
        if (rd != fp_nearest)
                (void) __swapRD(rd);
        return (x);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2468 Tue Nov  4 19:00:05 2014
new/usr/src/lib/libm/common/LD/logbl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __logbl = logbl
#pragma weak logbl = __logbl

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */
#define _C99SUSv3_logb  _C99SUSv3_logb_subnormal_is_like_ilogb

#if defined(__sparc)
#define ISNORMALL(k, x) (k != 0x7fff)                   /* assuming k != 0 */
#define X86PDNRM(k, x)
#define XSCALE_OFFSET   0x406f                          /* 0x3fff + 112 */
static const long double xscale = 5192296858534827628530496329220096.0L;
                                                                /* 2^112 */
#elif defined(__x86)
/*
 * if pseudo-denormal, replace by the equivalent normal
 */
#define X86PDNRM(k, x)  if (k == 0 && (((int *) &x)[1] & 0x80000000) != 0) \
                                ((int *) &x)[2] |= k = 1
#if defined(HANDLE_UNSUPPORTED)                         /* assuming k != 0 */
#define ISNORMALL(k, x) (k != 0x7fff && (((int *) &x)[1] & 0x80000000) != 0)
#else
#define ISNORMALL(k, x) (k != 0x7fff)
#endif
#define XSCALE_OFFSET   0x403e                          /* 0x3fff + 63 */
static const long double xscale = 9223372036854775808.0L;       /* 2^63 */
#endif

static long double
raise_division(long double v) {
#pragma STDC FENV_ACCESS ON
        static const long double zero = 0.0L;
```



```
        return (v / zero);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2732 Tue Nov  4 19:00:06 2014
new/usr/src/lib/libm/common/LD/nextafterl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __nextafterl = nextafterl
#pragma weak nextafterl = __nextafterl

#include "libm.h"
#include <float.h>              /* LDBL_MAX, LDBL_MIN */

#if defined(__sparc)
#define n0      0
#define n1      1
#define n2      2
#define n3      3
#define X86PDNRM1(x)
#define INC(px) { \
                        if (++px[n3] == 0) \
                                if (++px[n2] == 0) \
                                        if (++px[n1] == 0) \
                                                ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n3] == 0xffffffff) \
                                if (--px[n2] == 0xffffffff) \
                                        if (--px[n1] == 0xffffffff) \
                                                --px[n0]; \
                }
#elif defined(__x86)
#define n0      2
#define n1      1
#define n2      0
#define n3      0
/*
 * if pseudo-denormal, replace by the equivalent normal
 */
```



```
#define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
                                0x80000000) != 0) \
                                ((int *) &x)[2] |= 1
#define INC(px) { \
                        if (++px[n2] == 0) \
                                if ((++px[n1] & ~0x80000000) == 0) \
                                        px[n1] = 0x80000000, ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n2] == 0xffffffff) \
                                if (--px[n1] == 0x7fffffff) \
                                        if ((--px[n0] & 0x7fff) != 0) \
                                                px[n1] |= 0x80000000; \
                }
#endif

long double
nextafterl(long double x, long double y) {
        int *px = (int *) &x;
        int *py = (int *) &y;

        if (x == y)
                return (y);             /* C99 requirement */
        if (x != x || y != y)
                return (x * y);

        if (ISZEROL(x)) {       /* x == 0.0 */
                px[n0] = py[n0] & XSGNMSK;
                px[n1] = px[n2] = 0;
                px[n3] = 1;
        } else {
                X86PDNRM1(x);
                if ((px[n0] & XSGNMSK) == 0) {  /* x > 0.0 */
                        if (x > y)      /* x > y */
                                DEC(px)
                        else
                                INC(px)
                } else {
                        if (x < y)      /* x < y */
                                DEC(px)
                        else
                                INC(px)
                }
        }
#ifndef lint
        {
                volatile long double dummy;
                int k = XBIASED_EXP(x);

                if (k == 0)
                        dummy = LDBL_MIN * copysignl(LDBL_MIN, x);
                else if (k == 0x7fff)
                        dummy = LDBL_MAX * copysignl(LDBL_MAX, x);
        }
#endif
        return (x);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1768 Tue Nov  4 19:00:06 2014
new/usr/src/lib/libm/common/LD/scalbl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbl = scalbl
#pragma weak scalbl = __scalbl

/*
 * scalbl(x,n): return x * 2**n by manipulating exponent.
 */

#include "libm.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

long double
scalbl(long double x, long double fn) {
        int *py = (int *) &fn, n;
        long double z;

        if (isnanl(x) || isnanl(fn))
                return x * fn;

        /* fn is +/-Inf */
#if defined(_BIG_ENDIAN)
        if ((py[0] & 0x7fff0000) == 0x7fff0000) {
                if ((py[0] & 0x80000000) != 0)
#else
        if ((py[2] & 0x7fff) == 0x7fff) {
                if ((py[2] & 0x8000) != 0)
#endif
                        return x / (-fn);
                else
                        return x * fn;
        }
```



```
        if (rintl(fn) != fn)
                return (fn - fn) / (fn - fn);
        if (fn > 65000.0L)
                z = scalbnl(x, 65000);
        else if (-fn > 65000.0L)
                z = scalbnl(x, -65000);
        else {
                n = (int) fn;
                z = scalbnl(x, n);
        }
        return z;
}
```

_____*unchanged_portion_omitted_*

****
1094 Tue Nov 4 19:00:07 2014
new/usr/src/lib/libm/common/LD/signgaml.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __signgaml = signgaml
#pragma weak signgaml = __signgaml

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

int signgaml = 0;
```

****
1226 Tue Nov 4 19:00:07 2014
new/usr/src/lib/libm/common/LD/significandl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __significandl = significandl
#pragma weak significandl = __significandl

#include "libm.h"

long double
significandl(long double x) {
        if (ISZEROL(x) || XBIASED_EXP(x) == 0x7fff)     /* 0/+-Inf/NaN */
                return (x + x);
        else
                return (scalbnl(x, -ilogbl(x)));
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2900 Tue Nov  4 19:00:07 2014
new/usr/src/lib/libm/common/LD/sincosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sincosl = sincosl
#pragma weak sincosl = __sincosl

/* INDENT OFF */
/* cosl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sincosl     ... sin and cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
```



```
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

void
sincosl(long double x, long double *s, long double *c) {
        long double y[2], z = 0.0L;
        int n, ix;
#if defined(__i386) || defined(__amd64)
        int *px = (int *) &x;
#endif

        /* trig(Inf or NaN) is NaN */
        if (!finitel(x)) {
                *s = *c = x - x;
                return;
        }

        /* High word of x. */
#if defined(__i386) || defined(__amd64)
        XTOI(px, ix);
#else
        ix = *(int *) &x;
#endif

        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)
                *s = __k_sincosl(x, z, c);

        /* argument reduction needed */
        else {
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                case 0:
                        *s = __k_sincosl(y[0], y[1], c);
                        break;
                case 1:
                        *c = -__k_sincosl(y[0], y[1], s);
                        break;
                case 2:
                        *s = -__k_sincosl(y[0], y[1], c);
                        *c = -*c;
                        break;
                case 3:
                        *c = __k_sincosl(y[0], y[1], s);
                        *s = -*s;
                }
        }
}
______unchanged_portion_omitted_
```

```
**********************************************************
    5985 Tue Nov  4 19:00:08 2014
new/usr/src/lib/libm/common/LD/sincospil.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sincospil = __sincospil

/*
 * void sincospil(long double x, long double *s, long double *c)
 * *s = sinl(pi*x); *c = cosl(pi*x);
 *
 * Algorithm, 10/17/2002, K.C. Ng
 * ------------------------------
 * Let y = |4x|, z = floor(y), and n = (int)(z mod 8.0) (displayed in binary).
 *      1. If y == z, then x is a multiple of pi/4. Return the following values:
 *              ---------------------------------------------------
 *               n  x mod 2    sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ---------------------------------------------------
 *               000  0.00       +0 ___       +1 ___       +0
 *               001  0.25       +\/0.5       +\/0.5       +1
 *               010  0.50       +1 ___       +0 ___       +inf
 *               011  0.75       +\/0.5       -\/0.5       -1
 *               100  1.00       -0 ___       -1 ___       +0
 *               101  1.25       -\/0.5       -\/0.5       +1
 *               110  1.50       -1 ___       -0 ___       +inf
 *               111  1.75       -\/0.5       +\/0.5       -1
 *              ---------------------------------------------------
 *      2. Otherwise,
 *              ---------------------------------------------------
 *                n     t        sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ---------------------------------------------------
 *               000  (y-z)/4     sinpi(t)     cospi(t)    tanpi(t)
 *               001  (z+1-y)/4   cospi(t)     sinpi(t)    1/tanpi(t)
 *               010  (y-z)/4     cospi(t)    -sinpi(t)   -1/tanpi(t)
 *               011  (z+1-y)/4   sinpi(t)    -cospi(t)   -tanpi(t)
 *               100  (y-z)/4    -sinpi(t)    -cospi(t)    tanpi(t)
 *               101  (z+1-y)/4  -cospi(t)    -sinpi(t)    1/tanpi(t)
```



```
 *               110  (y-z)/4    -cospi(t)     sinpi(t)   -1/tanpi(t)
 *               111  (z+1-y)/4  -sinpi(t)     cospi(t)   -tanpi(t)
 *              ---------------------------------------------------
 *
 * NOTE. This program compute sinpi/cospi(t<0.25) by __k_sin/cos(pi*t, 0.0).
 * This will return a result with error slightly more than one ulp (but less
 * than 2 ulp). If one wants accurate result,  one may break up pi*t in
 * high (tpi_h) and low (tpi_l) parts and call __k_sin/cos(tip_h, tip_lo)
 * instead.
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

#define I(q, m) ((int *) &(q))[m]
#define U(q, m) ((unsigned *) &(q))[m]
#if defined(__i386) || defined(__amd64)
#define LDBL_MOST_SIGNIF_I(ld)  ((I(ld, 2) << 16) | (0xffff & (I(ld, 1) >> 15)))
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, 0)
#define PREC    64
#define PRECM1  63
#define PRECM2  62
static const long double twoPRECM2 = 9.22337203685477580800000000000000000e+18L;
#else
#define LDBL_MOST_SIGNIF_I(ld)  I(ld, 0)
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, sizeof(long double) / sizeof(int) - 1)
#define PREC    113
#define PRECM1  112
#define PRECM2  111
static const long double twoPRECM2 = 5.19229685853482762853049632922009 6e+33L;
#endif

static const long double
zero    = 0.0L,
quater  = 0.25L,
one     = 1.0L,
pi      = 3.14159265358979323846264338327950288419 7e+0000L,
sqrth   = 0.70710678118654752440084436210484903928 4835937688474,
tiny    = 1.0e-100;

void
sincospil(long double x, long double *s, long double *c) {
        long double y, z, t;
        int hx, n, k;
        unsigned lx;

        hx = LDBL_MOST_SIGNIF_I(x);
        lx = LDBL_LEAST_SIGNIF_U(x);
        k = ((hx & 0x7fff0000) >> 16) - 0x3fff;
        if (k >= PRECM2) {              /* |x| >= 2**(Prec-2) */
                if (k >= 16384) {
                        *s = *c = x - x;
                }
                else {
                        if (k >= PREC) {
                                *s = zero;
                                *c = one;
                        }
                        else if (k == PRECM1) {
                                if ((lx & 1) == 0) {
                                        *s = zero;
                                        *c = one;
                                }
```

```
125                                 else {
126                                         *s = -zero;
127                                         *c = -one;
128                                 }
129                         }
130                         else {  /* k = Prec - 2 */
131                                 if ((lx & 1) == 0) {
132                                         *s = zero;
133                                         *c = one;
134                                 }
135                                 else {
136                                         *s = one;
137                                         *c = zero;
138                                 }
139                                 if ((lx & 2) != 0) {
140                                         *s = -*s;
141                                         *c = -*c;
142                                 }
143                         }
144                 }
145         }
146         else if (k < -2)        /* |x| < 0.25 */
147                 *s = __k_sincosl(pi * fabsl(x), zero, c);
148         else {
149                 /* y = |4x|, z = floor(y), and n = (int)(z mod 8.0) */
150                 y = 4.0L * fabsl(x);
151                 if (k < PRECM2) {
152                         z = y + twoPRECM2;
153                         n = LDBL_LEAST_SIGNIF_U(z) & 7; /* 3 LSb of z */
154                         t = z - twoPRECM2;
155                         k = 0;
156                         if (t == y)
157                                 k = 1;
158                         else if (t > y) {
159                                 n -= 1;
160                                 t = quater + (y - t) * quater;
161                         }
162                         else
163                                 t = (y - t) * quater;
164                 }
165                 else {  /* k = Prec-3 */
166                         n = LDBL_LEAST_SIGNIF_U(y) & 7; /* 3 LSb of z */
167                         k = 1;
168                 }
169                 if (k) {        /* x = N/4 */
170                         if ((n & 1) != 0)
171                                 *s = *c = sqrth + tiny;
172                         else
173                                 if ((n & 2) == 0) {
174                                         *s = zero;
175                                         *c = one;
176                                 }
177                                 else {
178                                         *s = one;
179                                         *c = zero;
180                                 }
181                         if ((n & 4) != 0)
182                                 *s = -*s;
183                         if (((n + 1) & 4) != 0)
184                                 *c = -*c;
185                 }
186                 else {
187                         if ((n & 1) != 0)
188                                 t = quater - t;
189                         if (((n + (n & 1)) & 2) == 0)
190                                 *s = __k_sincosl(pi * t, zero, c);
```



```
191                         else
192                                 *c = __k_sincosl(pi * t, zero, s);
193                         if ((n & 4) != 0)
194                                 *s = -*s;
195                         if (((n + 2) & 4) != 0)
196                                 *c = -*c;
197                 }
198         }
199         if (hx < 0)
200                 *s = -*s;
201 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2237 Tue Nov  4 19:00:09 2014
new/usr/src/lib/libm/common/LD/sinhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinhl = sinhl
#pragma weak sinhl = __sinhl

#include "libm.h"
#include "longdouble.h"

/* SINH(X)
 * RETURN THE HYPERBOLIC SINE OF X
 *
 * Method :
 *      1. reduce x to non-negative by SINH(-x) = - SINH(x).
 *      2.
 *                                                 EXPM1(x) + EXPM1(x)/(EXPM1(x)+1)
 *          0 <= x <= lnovft     : SINH(x) := -------------------------------
 *                                                            2
 *
 *     lnovft <= x <  INF        : SINH(x) := EXP(x-MEP1*ln2)*2**ME
 *
 * here
 *      lnovft          logarithm of the overflow threshold
 *                      = MEP1*ln2 chopped to machine precision.
 *      ME              maximum exponent
 *      MEP1            maximum exponent plus 1
 *
 * Special cases:
 *      SINH(x) is x if x is +INF, -INF, or NaN.
 *      only SINH(0)=0 is exact for finite argument.
 *
 */

static const long double C[] = {
```



```
        0.5L,
        1.0L,
        1.13565234062941394879149e+04L,
        7.00444768624254908785898 5e-16L
};
```

_____unchanged_portion_omitted_

```
**********************************************************
    2839 Tue Nov  4 19:00:10 2014
new/usr/src/lib/libm/common/LD/sinl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinl = sinl
#pragma weak sinl = __sinl

/* INDENT OFF */
/* sinl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sinl                ... sin function on [-pi/4,pi/4]
 *      __k_cosl                ... cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
```



```
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

long double
sinl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;
#if defined(__i386) || defined(__amd64)
        int *px = (int *) &x;
#endif

        /* sin(Inf or NaN) is NaN */
        if (!finitel(x))
                return x - x;

        /* High word of x. */
#if defined(__i386) || defined(__amd64)
        XTOI(px, ix);
#else
        ix = *(int *) &x;
#endif
        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)
                return __k_sinl(x, z);

        /* argument reduction needed */
        else {
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                case 0:
                        return __k_sinl(y[0], y[1]);
                case 1:
                        return __k_cosl(y[0], y[1]);
                case 2:
                        return -__k_sinl(y[0], y[1]);
                case 3:
                        return -__k_cosl(y[0], y[1]);
                /* NOTREACHED */
                }
        }
        return 0.0L;
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    5538 Tue Nov  4 19:00:10 2014
new/usr/src/lib/libm/common/LD/sinpil.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sinpil = __sinpil

/* long double sinpil(long double x),
 * return long double precision sinl(pi*x).
 *
 * Algorithm, 10/17/2002, K.C. Ng
 * ------------------------------
 * Let y = |4x|, z = floor(y), and n = (int)(z mod 8.0) (displayed in binary).
 *      1. If y == z, then x is a multiple of pi/4. Return the following values:
 *              ------------------------------------------------
 *                n  x mod 2    sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ------------------------------------------------
 *               000  0.00       +0 ___       +1 ___      +0
 *               001  0.25       +\/0.5       +\/0.5      +1
 *               010  0.50       +1 ___       +0 ___      +inf
 *               011  0.75       +\/0.5       -\/0.5      -1
 *               100  1.00       -0 ___       -1 ___      +0
 *               101  1.25       -\/0.5       -\/0.5      +1
 *               110  1.50       -1 ___       -0 ___      +inf
 *               111  1.75       -\/0.5       +\/0.5      -1
 *              ------------------------------------------------
 *      2. Otherwise,
 *             ------------------------------------------------
 *               n     t        sin(x*pi)    cos(x*pi)   tan(x*pi)
 *             ------------------------------------------------
 *              000  (y-z)/4     sinpi(t)     cospi(t)    tanpi(t)
 *              001  (z+1-y)/4   cospi(t)     sinpi(t)    1/tanpi(t)
 *              010  (y-z)/4     cospi(t)    -sinpi(t)   -1/tanpi(t)
 *              011  (z+1-y)/4   sinpi(t)    -cospi(t)   -tanpi(t)
 *              100  (y-z)/4    -sinpi(t)    -cospi(t)    tanpi(t)
 *              101  (z+1-y)/4  -cospi(t)    -sinpi(t)    1/tanpi(t)
 *              110  (y-z)/4    -cospi(t)     sinpi(t)   -1/tanpi(t)
```



```
 *              111  (z+1-y)/4  -sinpi(t)     cospi(t)   -tanpi(t)
 *             ------------------------------------------------
 *
 * NOTE. This program compute sinpi/cospi(t<0.25) by __k_sin/cos(pi*t, 0.0).
 * This will return a result with error slightly more than one ulp (but less
 * than 2 ulp). If one wants accurate result,  one may break up pi*t in
 * high (tpi_h) and low (tpi_l) parts and call __k_sin/cos(tip_h, tip_lo)
 * instead.
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

#define I(q, m) ((int *) &(q))[m]
#define U(q, m) ((unsigned *) &(q))[m]
#if defined(__i386) || defined(__amd64)
#define LDBL_MOST_SIGNIF_I(ld)  ((I(ld, 2) << 16) | (0xffff & (I(ld, 1) >> 15)))
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, 0)
#define PREC    64
#define PRECM1  63
#define PRECM2  62
static const long double twoPRECM2 = 9.22337203685477580800000000000000000e+18L;
#else
#define LDBL_MOST_SIGNIF_I(ld)  I(ld, 0)
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, sizeof(long double) / sizeof(int) - 1)
#define PREC    113
#define PRECM1  112
#define PRECM2  111
static const long double twoPRECM2 = 5.19229685853482762853049632922009 6e+33L;
#endif

static const long double
zero    = 0.0L,
quater  = 0.25L,
one     = 1.0L,
pi      = 3.14159265358979323846264338327950288419 7e+0000L,
sqrth   = 0.70710678118654752440084436210484903928 4835937688474,
tiny    = 1.0e-100;

long double
sinpil(long double x) {
        long double y, z, t;
        int hx, n, k;
        unsigned lx;

        hx = LDBL_MOST_SIGNIF_I(x);
        lx = LDBL_LEAST_SIGNIF_U(x);
        k = ((hx & 0x7fff0000) >> 16) - 0x3fff;
        if (k >= PRECM2) {              /* |x| >= 2**(Prec-2) */
                if (k >= 16384)
                        y = x - x;
                else {
                        if (k >= PREC)
                                y = zero;
                        else if (k == PRECM1)
                                y = (lx & 1) == 0 ? zero: -zero;
                        else {  /* k = Prec - 2 */
                                y = (lx & 1) == 0 ? zero : one;
                                if ((lx & 2) != 0)
                                        y = -y;
                        }
                }
        }
```

```
125         else if (k < -2)        /* |x| < 0.25 */
126                 y = __k_sinl(pi * fabsl(x), zero);
127         else {
128                 /* y = |4x|, z = floor(y), and n = (int)(z mod 8.0) */
129                 y = 4.0L * fabsl(x);
130                 if (k < PRECM2) {
131                         z = y + twoPRECM2;
132                         n = LDBL_LEAST_SIGNIF_U(z) & 7; /* 3 LSb of z */
133                         t = z - twoPRECM2;
134                         k = 0;
135                         if (t == y)
136                                 k = 1;
137                         else if (t > y) {
138                                 n -= 1;
139                                 t = quater + (y - t) * quater;
140                         }
141                         else
142                                 t = (y - t) * quater;
143                 }
144                 else {  /* k = Prec-3 */
145                         n = LDBL_LEAST_SIGNIF_U(y) & 7; /* 3 LSb of z */
146                         k = 1;
147                 }
148                 if (k) {        /* x = N/4 */
149                         if ((n & 1) != 0)
150                                 y = sqrth + tiny;
151                         else
152                                 y = (n & 2) == 0 ? zero : one;
153                         if ((n & 4) != 0)
154                                 y = -y;
155                 }
156                 else {
157                         if ((n & 1) != 0)
158                                 t = quater - t;
159                         if (((n + (n & 1)) & 2) == 0)
160                                 y = __k_sinl(pi * t, zero);
161                         else
162                                 y = __k_cosl(pi * t, zero);
163                         if ((n & 4) != 0)
164                                 y = -y;
165                 }
166         }
167         return hx >= 0 ? y : -y;
168 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2601 Tue Nov  4 19:00:10 2014
new/usr/src/lib/libm/common/LD/tanhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tanhl = tanhl
#pragma weak tanhl = __tanhl

/*
 * tanhl(x) returns the hyperbolic tangent of x
 *
 * Method :
 *      1. reduce x to non-negative:  tanhl(-x) = - tanhl(x).
 *      2.
 *        0      <  x <=  small    :  tanhl(x) := x
 *                                                -expm1l(-2x)
 *        small  <  x <=  1        :  tanhl(x) := --------------
 *                                                 expm1l(-2x) + 2
 *                                                       2
 *        1      <= x <= threshold :  tanhl(x) := 1 -  ---------------
 *                                                     expm1l(2x) + 2
 *     threshold <  x <= INF       :  tanhl(x) := 1.
 *
 * where
 *      single :        small = 1.e-5           threshold = 11.0
 *      double :        small = 1.e-10          threshold = 22.0
 *      quad   :        small = 1.e-20          threshold = 45.0
 *
 * Note: threshold was chosen so that
 *              fl(1.0+2/(expm1(2*threshold)+2)) == 1.
 *
 * Special cases:
 *      tanhl(NaN) is NaN;
 *      only tanhl(0.0)=0.0 is exact for finite argument.
 */

#include "libm.h"
```



```
#include "longdouble.h"

static const long double small = 1.0e-20L, one = 1.0, two = 2.0,
#ifndef lint
        big = 1.0e+20L,
#endif
        threshold = 45.0L;

long double
tanhl(long double x) {
        long double t, y, z;
        int signx;
#ifndef lint
        volatile long double dummy;
#endif

        if (isnanl(x))
                return (x + x);         /* x is NaN */
        signx = signbitl(x);
        t = fabsl(x);
        z = one;
        if (t <= threshold) {
                if (t > one)
                        z = one - two / (expm1l(t + t) + two);
                else if (t > small) {
                        y = expm1l(-t - t);
                        z = -y / (y + two);
                } else {
#ifndef lint
                        dummy = t + big;
                                                        /* inexact if t != 0 */
#endif
                        return (x);
                }
        } else if (!finitel(t))
                return (copysignl(one, x));
        else
                return (signx ? -z + small * small : z - small * small);
        return (signx ? -z : z);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2613 Tue Nov  4 19:00:11 2014
new/usr/src/lib/libm/common/LD/tanl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tanl = tanl
#pragma weak tanl = __tanl

/* INDENT OFF */
/* cosl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_tanl        ... tangent function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
```



```
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "longdouble.h"

#include <sys/isa_defs.h>

long double
tanl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;
#if defined(__i386) || defined(__amd64)
        int *px = (int *) &x;
#endif

        /* trig(Inf or NaN) is NaN */
        if (!finitel(x))
                return x - x;

        /* High word of x. */
#if defined(__i386) || defined(__amd64)
        XTOI(px, ix);
#else
        ix = *(int *) &x;
#endif

        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)
                return __k_tanl(x, z, 0);

        /* argument reduction needed */
        else {
                n = __rem_pio2l(x, y);
                return __k_tanl(y[0], y[1], n & 1);
        }
}
```

_____***unchanged_portion_omitted_***

****
1471 Tue Nov 4 19:00:11 2014
new/usr/src/lib/libm/common/Q/acoshl.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */

22 /*
23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
24  */
25 /*
26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
27  * Use is subject to license terms.
28  */

30 #pragma weak __acoshl = acoshl
30 #pragma weak acoshl = __acoshl

32 #include "libm.h"

34 static const long double
35         zero    = 0.0L,
36         ln2     = 6.93147180559945309417232121458176568075e-0001L,
37         one     = 1.0L,
38         big     = 1.e+20L;

40 long double
41 acoshl(long double x) {
42         long double t;

44         if (isnanl(x))
45                 return (x + x);
46         else if (x > big)
47                 return (logl(x) + ln2);
48         else if (x > one) {
49                 t = sqrtl(x - one);
50                 return (log1pl(t * (t + sqrtl(x + one))));
51         } else if (x == one)
52                 return (zero);
53         else
54                 return ((x - x) / (x - x));
55 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1830 Tue Nov  4 19:00:12 2014
new/usr/src/lib/libm/common/Q/acosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * arccosin function
 *                              _______
 *                             / 1 - x
 *      acos(x) = 2*atan2(    / -------- , 1 )
 *                          \/   1 + x
 *
 *                              _______
 *                             / 1 - x
 *              = 2*atan (    / -------- ) for non-exceptional x.
 *                          \/   1 + x
 *
 * Special cases:
 *      if x is NaN, return x itself;
 *      if |x|>1, return NaN with invalid signal.
 */

#pragma weak __acosl = acosl
#pragma weak acosl = __acosl

#include "libm.h"

static const long double zero = 0.0L, one = 1.0L;

long double
acosl(long double x) {
        if (isnanl(x))
                return (x + x);
        else if (fabsl(x) < one)
                x = atanl(sqrtl((one - x) / (one + x)));
        else if (x == -one)
                x = atan2l(one, zero);  /* x <- PI */
```



```
        else if (x == one)
                x = zero;
        else {          /* |x| > 1  create invalid signal */
                return (zero / zero);
        }
        return (x + x);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1590 Tue Nov  4 19:00:12 2014
new/usr/src/lib/libm/common/Q/asinhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinhl = asinhl
#pragma weak asinhl = __asinhl

#include "libm.h"

static const long double
        ln2     = 6.93147180559945309417232121458176568075e-0001L,
        one     = 1.0L,
        big     = 1.0e+20L,
        tiny    = 1.0e-20L;

long double
asinhl(long double x) {
        long double t, w;
        volatile long double dummy;

        w = fabsl(x);
        if (isnanl(x))
                return (x + x); /* x is NaN */
        if (w < tiny) {
#ifndef lint
                dummy = x + big;        /* inexact if x != 0 */
#endif
                return (x);     /* tiny x */
        } else if (w < big) {
                t = one / w;
                return (copysignl(log1pl(w + w / (t + sqrtl(one + t * t))), x));
        } else
                return (copysignl(logl(w) + ln2, x));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2010 Tue Nov  4 19:00:13 2014
new/usr/src/lib/libm/common/Q/asinl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinl = asinl
#pragma weak asinl = __asinl

/*
 *      asinl(x) = atan2l(x,sqrt(1-x*x));
 *
 * For better accuracy, 1-x*x is computed as follows
 *      1-x*x                   if x <  0.5,
 *      2*(1-|x|)-(1-|x|)*(1-|x|) if x >= 0.5.
 *
 * Special cases:
 *      if x is NaN, return x itself;
 *      if |x|>1, return NaN with invalid signal.
 */

#include "libm.h"

static const long double zero = 0.0L, small = 1.0e-20L, half = 0.5L, one = 1.0L;
#ifndef lint
static const long double big = 1.0e+20L;
#endif

long double
asinl(long double x) {
        long double t, w;
        volatile long double dummy;

        w = fabsl(x);
        if (isnanl(x))
                return (x + x);
        else if (w <= half) {
                if (w < small) {
```



```
#ifndef lint
                        dummy = w + big;
                                                        /* inexact if w != 0 */
#endif
                        return (x);
                } else
                        return (atanl(x / sqrtl(one - x * x)));
        } else if (w < one) {
                t = one - w;
                w = t + t;
                return (atanl(x / sqrtl(w - t * t)));
        } else if (w == one)
                return (atan2l(x, zero));       /* asin(+-1) =  +- PI/2 */
        else
                return (zero / zero);           /* |x| > 1: invalid */
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    4154 Tue Nov  4 19:00:13 2014
new/usr/src/lib/libm/common/Q/atan2l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * atan2l(y,x)
 *
 * Method :
 *      1. Reduce y to positive by atan2(y,x)=-atan2(-y,x).
 *      2. Reduce x to positive by (if x and y are unexceptional):
 *              ARG (x+iy) = arctan(y/x)           ... if x > 0,
 *              ARG (x+iy) = pi - arctan[y/(-x)]   ... if x < 0,
 *
 * Special cases:
 *
 *      ATAN2((anything), NaN ) is NaN;
 *      ATAN2(NAN , (anything) ) is NaN;
 *      ATAN2(+-0, +(anything but NaN)) is +-0  ;
 *      ATAN2(+-0, -(anything but NaN)) is +-PI ;
 *      ATAN2(+-(anything but 0 and NaN), 0) is +-PI/2;
 *      ATAN2(+-(anything but INF and NaN), +INF) is +-0 ;
 *      ATAN2(+-(anything but INF and NaN), -INF) is +-PI;
 *      ATAN2(+-INF,+INF ) is +-PI/4 ;
 *      ATAN2(+-INF,-INF ) is +-3PI/4;
 *      ATAN2(+-INF, (anything but,0,NaN, and INF)) is +-PI/2;
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */

#pragma weak __atan2l = atan2l
#pragma weak atan2l = __atan2l
```



```
#include "libm.h"
#include "longdouble.h"

static const long double
        zero    =  0.0L,
        tiny    =  1.0e-40L,
        one     =  1.0L,
        half    =  0.5L,
        PI3o4   =  2.3561944901923449288469825374596271631 48L,
        PIo4    =  0.7853981633974483096156608458198757210 49L,
        PIo2    =  1.5707963267948966192313216916397514420 99L,
        PI      =  3.1415926535897932384626433832795028841 97L,
        PI_lo   =  8.6718101301237810247970440260433519687 62e-35L;

long double
atan2l(long double y, long double x) {
        long double t, z;
        int k, m, signy, signx;

        if (x != x || y != y)
                return (x + y); /* return NaN if x or y is NAN */
        signy = signbitl(y);
        signx = signbitl(x);
        if (x == one)
                return (atanl(y));
        m = signy + signx + signx;

        /* when y = 0 */
        if (y == zero)
                switch (m) {
                case 0:
                        return (y);     /* atan(+0,+anything) */
                case 1:
                        return (y);     /* atan(-0,+anything) */
                case 2:
                        return (PI + tiny);     /* atan(+0,-anything) */
                case 3:
                        return (-PI - tiny);    /* atan(-0,-anything) */
                }

        /* when x = 0 */
        if (x == zero)
                return (signy == 1 ? -PIo2 - tiny : PIo2 + tiny);

        /* when x is INF */
        if (!finitel(x)) {
                if (!finitel(y)) {
                        switch (m) {
                        case 0:
                                return (PIo4 + tiny);   /* atan(+INF,+INF) */
                        case 1:
                                return (-PIo4 - tiny);  /* atan(-INF,+INF) */
                        case 2:
                                return (PI3o4 + tiny);  /* atan(+INF,-INF) */
                        case 3:
                                return (-PI3o4 - tiny); /* atan(-INF,-INF) */
                        }
                } else {
                        switch (m) {
                        case 0:
                                return (zero);  /* atan(+...,+INF) */
                        case 1:
                                return (-zero); /* atan(-...,+INF) */
                        case 2:
                                return (PI + tiny);     /* atan(+...,-INF) */
                        case 3:
```

```
127                                 return (-PI - tiny);    /* atan(-...,-INF) */
128                         }
129                 }
130         }
131         /* when y is INF */
132         if (!finitel(y))
133                 return (signy == 1 ? -PIo2 - tiny : PIo2 + tiny);

135         /* compute y/x */
136         x = fabsl(x);
137         y = fabsl(y);
138         t = PI_lo;
139         k = (ilogbl(y) - ilogbl(x));

141         if (k > 120)
142                 z = PIo2 + half * t;
143         else if (m > 1 && k < -120)
144                 z = zero;
145         else
146                 z = atanl(y / x);

148         switch (m) {
149         case 0:
150                 return (z);     /* atan(+,+) */
151         case 1:
152                 return (-z);    /* atan(-,+) */
153         case 2:
154                 return (PI - (z - t));  /* atan(+,-) */
155         case 3:
156                 return ((z - t) - PI);  /* atan(-,-) */
157         }
158         /* NOTREACHED */
159     return 0.0L;
160 }
```

_____***unchanged_portion_omitted_***

****
1229 Tue Nov 4 19:00:14 2014
new/usr/src/lib/libm/common/Q/atan2pil.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak atan2pil = __atan2pil

#include "libm.h"

/*
 * atan2pil(y,x) = atan2l(y, x) / pi
 */

static const long double invpi = 3.183098861837906715377675267450287240689e-1L;

long double
atan2pil(long double y, long double x) {
        return (atan2l(y, x) * invpi);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1833 Tue Nov  4 19:00:14 2014
new/usr/src/lib/libm/common/Q/atanhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanhl = atanhl
#pragma weak atanhl = __atanhl

#include "libm.h"

/*
 *                    1             2x                          x
 *      atanhl(x) = --- * LOG(1 + -------) = 0.5 * log1pl(2 * --------)
 *                    2           1 - x                       1 - x
 * Note: to guarantee atanhl(-x) = -atanhl(x), we use
 *                   sign(x)              |x|
 *      atanhl(x) = ------- * log1pl(2*-------).
 *                     2               1 - |x|
 *
 * Special cases:
 *      atanhl(x) is NaN if |x| > 1 with signal;
 *      atanhl(NaN) is that NaN with no signal;
 *      atanhl(+-1) is +-INF with signal.
 *
 */

static const long double zero = 0.0L, half = 0.5L, one = 1.0L;

long double
atanhl(long double x) {
        long double t;

        t = fabsl(x);
        if (t == one)
                return (x / zero);
        t = t / (one - t);
        return (copysignl(half, x) * log1pl(t + t));
```



```
}
______unchanged_portion_omitted_
```

```
**********************************************************
    6575 Tue Nov  4 19:00:15 2014
new/usr/src/lib/libm/common/Q/atanl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanl = atanl
#pragma weak atanl = __atanl

/*
 * atanl(x)
 * Table look-up algorithm
 * By K.C. Ng, March 9, 1989
 *
 * Algorithm.
 *
 * The algorithm is based on atan(x)=atan(y)+atan((x-y)/(1+x*y)).
 * We use poly1(x) to approximate atan(x) for x in [0,1/8] with
 * error (relative)
 *      |(atan(x)-poly1(x))/x|<= 2^-115.94      long double
 *      |(atan(x)-poly1(x))/x|<= 2^-58.85       double
 *      |(atan(x)-poly1(x))/x|<= 2^-25.53       float
 * and use poly2(x) to approximate atan(x) for x in [0,1/65] with
 * error (absolute)
 *      |atan(x)-poly2(x)|<= 2^-122.15          long double
 *      |atan(x)-poly2(x)|<= 2^-64.79           double
 *      |atan(x)-poly2(x)|<= 2^-35.36           float
 * Here poly1 and poly2 are odd polynomial with the following form:
 *              x + x^3*(a1+x^2*(a2+...))
 *
 * (0). Purge off Inf and NaN and 0
 * (1). Reduce x to positive by atan(x) = -atan(-x).
 * (2). For x <= 1/8, use
 *      (2.1) if x < 2^(-prec/2-2), atan(x) =  x  with inexact
 *      (2.2) Otherwise
 *              atan(x) = poly1(x)
 * (3). For x >= 8 then
 *      (3.1) if x >= 2^(prec+2),   atan(x) = atan(inf) - pio2lo
```



```
 *      (3.2) if x >= 2^(prec/3+2), atan(x) = atan(inf) - 1/x
 *      (3.3) if x >  65,           atan(x) = atan(inf) - poly2(1/x)
 *      (3.4) Otherwise,            atan(x) = atan(inf) - poly1(1/x)
 *
 * (4). Now x is in (0.125, 8)
 *      Find y that match x to 4.5 bit after binary (easy).
 *      If iy is the high word of y, then
 *              single : j = (iy - 0x3e000000) >> 19
 *              double : j = (iy - 0x3fc00000) >> 16
 *              quad   : j = (iy - 0x3ffc0000) >> 12
 *
 *      Let s = (x-y)/(1+x*y). Then
 *      atan(x) = atan(y) + poly1(s)
 *              = _TBL_atanl_hi[j] + (_TBL_atanl_lo[j] + poly2(s) )
 *
 *      Note. |s| <= 1.5384615385e-02 = 1/65. Maxium occurs at x = 1.03125
 *
 */

#include "libm.h"

extern const long double _TBL_atanl_hi[], _TBL_atanl_lo[];
static const long double
        one     =   1.0L,
        p1      =  -3.33333333333333333333333333331344526118e-0001L,
        p2      =   1.99999999999999999999999989931277668570e-0001L,
        p3      =  -1.42857142857142857142855360622130953090 1e-0001L,
        p4      =   1.11111111111111111109521984273719747418e-0001L,
        p5      =  -9.09090909090909082550360383524806112332 3e-0002L,
        p6      =   7.69230769230766405213074321470892525890 4e-0002L,
        p7      =  -6.66666666666021383518771322836371738826 6e-0002L,
        p8      =   5.88235294015243939909728335960866194950 4e-0002L,
        p9      =  -5.26315778044753399304661404050952966848 7e-0002L,
        p10     =   4.76189581687818493317585599088678843944 7e-0002L,
        p11     =  -4.34734500583227402268101972455353813592 2e-0002L,
        p12     =   3.98303191457963503750258920464775204273 6e-0002L,
        p13     =  -3.34820670446983057519665774941389489755 4e-0002L,
        q1      =  -3.33333333333333333333333333195273650186e-0001L,
        q2      =   1.99999999999999999999988146114392615808e-0001L,
        q3      =  -1.42857142857142857105763031943546711125 3e-0001L,
        q4      =   1.11111111111105373263048208994541544098e-0001L,
        q5      =  -9.09090909042183420916737325868102181644 1e-0002L,
        q6      =   7.69230537781369270685017176715070164453 9e-0002L,
        q7      =  -6.66089664439386149991473173430571790133 0e-0002L,
        pio2hi  =   1.57079632679489661923132169163975139874 0e+0000L,
        pio2lo  =   4.33590506506189051239852201302167598438 1e-0035L;

#define i0      0
#define i1      3

long double
atanl(long double x) {
        long double y, z, r, p, s;
        int *px = (int *) &x, *py = (int *) &y;
        int ix, iy, sign, j;

        ix = px[i0];
        sign = ix & 0x80000000;
        ix ^= sign;

        /* for |x| < 1/8 */
        if (ix < 0x3ffc0000) {
                if (ix < 0x3feb0000) {  /* when |x| < 2**(-prec/6-2) */
                        if (ix < 0x3fc50000) {  /* if |x| < 2**(-prec/2-2) */
                                s = one;
                                *(3 - i0 + (int *) &s) = -1;    /* s = 1-ulp */
```

```
127                                 *(1 + (int *) &s) = -1;
128                                 *(2 + (int *) &s) = -1;
129                                 *(i0 + (int *) &s) -= 1;
130                                 if ((int) (s * x) < 1)
131                                         return (x);     /* raise inexact */
132                         }
133                         z = x * x;
134                         if (ix < 0x3fe20000) {  /* if |x| < 2**(-prec/4-1) */
135                                 return (x + (x * z) * p1);
136                         } else {        /* if |x| < 2**(-prec/6-2) */
137                                 return (x + (x * z) * (p1 + z * p2));
138                         }
139                 }
140                 z = x * x;
141                 return (x + (x * z) * (p1 + z * (p2 + z * (p3 + z * (p4 +
142                         z * (p5 + z * (p6 + z * (p7 + z * (p8 + z * (p9 +
143                         z * (p10 + z * (p11 + z * (p12 + z * p13)))))))))));
144         }

146         /* for |x| >= 8.0 */
147         if (ix >= 0x40020000) {
148                 px[i0] = ix;
149                 if (ix < 0x40050400) {  /* x <  65 */
150                         r = one / x;
151                         z = r * r;
152                         /*
153                          * poly1
154                          */
155                         y = r * (one + z * (p1 + z * (p2 + z * (p3 +
156                                 z * (p4 + z * (p5 + z * (p6 + z * (p7 +
157                                 z * (p8 + z * (p9 + z * (p10 + z * (p11 +
158                                 z * (p12 + z * p13)))))))))));
159                         y -= pio2lo;
160                 } else if (ix < 0x40260000) {   /* x <  2**(prec/3+2) */
161                         r = one / x;
162                         z = r * r;
163                         /*
164                          * poly2
165                          */
166                         y = r * (one + z * (q1 + z * (q2 + z * (q3 + z * (q4 +
167                                 z * (q5 + z * (q6 + z * q7)))))));
168                         y -= pio2lo;
169                 } else if (ix < 0x40720000) {   /* x <  2**(prec+2) */
170                         y = one / x - pio2lo;
171                 } else if (ix < 0x7fff0000) {   /* x <  inf */
172                         y = -pio2lo;
173                 } else {                /* x is inf or NaN */
174                         if (((ix - 0x7fff0000) | px[1] | px[2] | px[i1]) != 0)
175                                 return (x - x);
176                         y = -pio2lo;
177                 }

179                 if (sign == 0)
180                         return (pio2hi - y);
181                 else
182                         return (y - pio2hi);
183         }

185         /* now x is between 1/8 and 8 */
186         px[i0] = ix;
187         iy = (ix + 0x00000800) & 0x7ffff000;
188         py[i0] = iy;
189         py[1] = py[2] = py[i1] = 0;
190         j = (iy - 0x3ffc0000) >> 12;

192         if (sign == 0)
```



```
193                 s = (x - y) / (one + x * y);
194         else
195                 s = (y - x) / (one + x * y);
196         z = s * s;
197         if (ix == iy)
198                 p = s * (one + z * (q1 + z * (q2 + z * (q3 + z * q4))));
199         else
200                 p = s * (one + z * (q1 + z * (q2 + z * (q3 + z * (q4 +
201                         z * (q5 + z * (q6 + z * q7)))))));
202         if (sign == 0) {
203                 r = p + _TBL_atanl_lo[j];
204                 return (r + _TBL_atanl_hi[j]);
205         } else {
206                 r = p - _TBL_atanl_lo[j];
207                 return (r - _TBL_atanl_hi[j]);
208         }
209 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1680 Tue Nov  4 19:00:15 2014
new/usr/src/lib/libm/common/Q/cbrtl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cbrtl = cbrtl
#pragma weak cbrtl = __cbrtl

#include "libm.h"
#include "longdouble.h"

#define n0      0

long double
cbrtl(long double x) {
        long double s, t, r, w, y;
        double dx, dy;
        int *py = (int *) &dy;
        int n, m, m3, sx;

        if (!finitel(x))
                return (x + x);
        if (iszerol(x))
                return (x);
        sx = signbitl(x);
        x = fabsl(x);
        n = ilogbl(x);
        m = n / 3;
        m3 = m + m + m;
        y = scalbnl(x, -m3);
        dx = (double) y;
        dy = cbrt(dx);
        py[1 - n0] += 2;
        if (py[1 - n0] == 0)
                py[n0] += 1;

        /* one step newton iteration to 113 bits with error < 0.667ulps */
```



```
        t = (long double) dy;
        t = scalbnl(t, m);
        s = t * t;
        r = x / s;
        w = t + t;
        r = (r - t) / (w + r);
        t += t * r;

        return (sx == 0 ? t : -t);
}
```

*_____unchanged_portion_omitted_*

****
1237 Tue Nov 4 19:00:16 2014
new/usr/src/lib/libm/common/Q/copysignl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __copysignl = copysignl
#pragma weak copysignl = __copysignl

#include "libm.h"

long double
copysignl(long double x, long double y) {
        int *px = (int *) &x;
        int *py = (int *) &y;

        px[HIXWORD] = (px[HIXWORD] & ~XSGNMSK) | (py[HIXWORD] & XSGNMSK);
        return (x);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2933 Tue Nov  4 19:00:16 2014
new/usr/src/lib/libm/common/Q/coshl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __coshl = coshl
#pragma weak coshl = __coshl

#include "libm.h"
#include "longdouble.h"


/*
 * coshl(X)
 * RETURN THE HYPERBOLIC COSINE OF X
 *
 * Method :
 *      1. Replace x by |x| (coshl(x) = coshl(-x)).
 *      2.
 *                                                      [ expl(x) - 1 ]^2
 *          0       <= x <= 0.3465 : coshl(x) := 1 + -------------------
 *                                                         2*expl(x)
 *
 *                                              expl(x) + 1/expl(x)
 *          0.3465  <= x <= thresh : coshl(x) := -------------------
 *                                                     2
 *          thresh  <= x <= lnovft : coshl(x) := expl(x)/2
 *          lnovft  <= x <  INF    : coshl(x) := scalbnl(expl(x-1024*ln2),1023)
 *
 * here
 *      thr1            a number that is near one half of ln2.
 *      thr2            a number such that
 *                              expl(thresh)+expl(-thresh)=expl(thresh)
 *      lnovft:         logrithm of the overflow threshold
 *                      = MEP1*ln2 chopped to machine precision.
 *      ME              maximum exponent
 *      MEP1            maximum exponent plus 1
```



```
 *
 * Special cases:
 *      coshl(x) is |x| if x is +INF, -INF, or NaN.
 *      only coshl(0)=1 is exact for finite x.
 */

#define ME      16383
#define MEP1    16384
#define LNOVFT  1.13565234062941439494919310779707634284 5e+4L
                /* last 32 bits of LN2HI is zero */
#define LN2HI   6.9314718055994530941723195474958448502 03e-0001L
#define LN2LO   1.6670859208305522088904493304003797541 69e-0025L
#define THR1    0.3465L
#define THR2    45.L

static const long double
        half    = 0.5L,
        tinyl   = 7.5e-37L,
        one     = 1.0L,
        ln2hi   = LN2HI,
        ln2lo   = LN2LO,
        lnovftL = LNOVFT,
        thr1    = THR1,
        thr2    = THR2;

long double
coshl(long double x) {
        long double t, w;

        w = fabsl(x);
        if (!finitel(w))
                return (w + w);         /* x is INF or NaN */
        if (w < thr1) {
                t = w < tinyl ? w : expm1l(w);
                w = one + t;
                if (w != one)
                        w = one + (t * t) / (w + w);
                return (w);
        } else if (w < thr2) {
                t = expl(w);
                return (half * (t + one / t));
        } else if (w <= lnovftL)
                return (half * expl(w));
        else {
                return (scalbnl(expl((w - MEP1 * ln2hi) - MEP1 * ln2lo), ME));
        }
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2670 Tue Nov  4 19:00:16 2014
new/usr/src/lib/libm/common/Q/cosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * cosl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sinl        ... sin function on [-pi/4,pi/4]
 *      __k_cosl        ... cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */
```



```
#pragma weak __cosl = cosl
#pragma weak cosl = __cosl

#include "libm.h"
#include "longdouble.h"

long double
cosl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;

        ix = *(int *) &x;                       /* High word of x */

        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)                   /* |x| ~< pi/4 */
                return (__k_cosl(x, z));
        else if (ix >= 0x7fff0000)              /* trig(Inf or NaN) is NaN */
                return (x - x);
        else {                                  /* argument reduction needed */
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                case 0:
                        return (__k_cosl(y[0], y[1]));
                case 1:
                        return (-__k_sinl(y[0], y[1]));
                case 2:
                        return (-__k_cosl(y[0], y[1]));
                case 3:
                        return (__k_sinl(y[0], y[1]));
                }
        }
        /* NOTREACHED */
    return 0.0L;
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   13609 Tue Nov  4 19:00:17 2014
new/usr/src/lib/libm/common/Q/erfl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double function erf,erfc (long double x)
 * K.C. Ng, September, 1989.
 *                          x
 *                   2      |\
 *     erf(x)  =  ---------  | exp(-t*t)dt
 *                sqrt(pi) \|
 *                          0
 *
 *     erfc(x) =  1-erf(x)
 *
 * method:
 *      Since erf(-x) = -erf(x), we assume x>=0.
 *      For x near 0, we have the expansion
 *
 *          erf(x) = (2/sqrt(pi))*(x - x^3/3 + x^5/10 - x^7/42 + ....).
 *
 *      Since 2/sqrt(pi) = 1.128379167095512573896158903121545171688,
 *      we use x + x*P(x^2) to approximate erf(x). This formula will
 *      guarantee the error less than one ulp where x is not too far
 *      away from 0. We note that erf(x)=x at x = 0.6174..... After
 *      some experiment, we choose the following approximation on
 *      interval [0,0.84375].
 *
 *      For x in [0,0.84375]
 *                 2             2       4              40
 *         P =  P(x ) = (p0 + p1 * x + p2 * x + ... + p20 * x  )
 *
 *         erf(x)  = x + x*P
 *         erfc(x) = 1 - erf(x)          if x<=0.25
 *                 = 0.5 + ((0.5-x)-x*P)  if x in [0.25,0.84375]
 *      precision: |P(x^2)-(erf(x)-x)/x| <= 2**-122.50
```



```
 *
 *      For x in [0.84375,1.25], let s = x - 1, and
 *      c = 0.84506291151 rounded to single (24 bits)
 *         erf(x)  = c  + P1(s)/Q1(s)
 *         erfc(x) = (1-c)  - P1(s)/Q1(s)
 *      precision: |P1/Q1 - (erf(x)-c)| <= 2**-118.41
 *
 *
 *      For x in [1.25,1.75], let s = x - 1.5, and
 *      c = 0.95478588343 rounded to single (24 bits)
 *         erf(x)  = c  + P2(s)/Q2(s)
 *         erfc(x) = (1-c)  - P2(s)/Q2(s)
 *      precision: |P1/Q1 - (erf(x)-c)| <= 2**-123.83
 *
 *
 *      For x in [1.75,16/3]
 *         erfc(x) = exp(-x*x)*(1/x)*R1(1/x)/S1(1/x)
 *         erf(x)  = 1 - erfc(x)
 *      precision: absolute error of R1/S1 is bounded by 2**-124.03
 *
 *      For x in [16/3,107]
 *         erfc(x) = exp(-x*x)*(1/x)*R2(1/x)/S2(1/x)
 *         erf(x)  = 1 - erfc(x) (if x>=9 simple return erf(x)=1 with inexact)
 *      precision: absolute error of R2/S2 is bounded by 2**-120.07
 *
 *      Else if inf > x >= 107
 *         erf(x)  = 1 with inexact
 *         erfc(x) = 0 with underflow
 *
 *      Special case:
 *         erf(inf)  = 1
 *         erfc(inf) = 0
 */

#pragma weak __erfl = erfl
#pragma weak __erfcl = erfcl
#pragma weak erfl = __erfl
#pragma weak erfcl = __erfcl

#include "libm.h"
#include "longdouble.h"

static const long double
        tiny        = 1e-40L,
        nearunfl    = 1e-4000L,
        half        = 0.5L,
        one         = 1.0L,
        onehalf     = 1.5L,
        L16_3       = 16.0L/3.0L;
/*
 * Coefficients for even polynomial P for erf(x)=x+x*P(x^2) on [0,0.84375]
 */
static const long double P[] = {        /* 21 coeffs */
   1.28379167095512573896158903121545171556e-0001L,
  -3.76126389031837524632052967707181559460 3e-0001L,
   1.12837916709551257389615890312120589913 5e-0001L,
  -2.68661706451312517594323548334462504609 2e-0002L,
   5.22397762544218784211184665298045456838 9e-0003L,
  -8.54832702345085283254662627108386272435 8e-0004L,
   1.20553329817896642510216471590223197667 2e-0004L,
  -1.49256503584062509767494490502789783899 6e-0005L,
   1.64621143658892473360464884917293669202 4e-0006L,
  -1.63658446912349197681583470479973351498 7e-0007L,
   1.48071928158789744530252900714477073930 5e-0008L,
  -1.22905553017078284304646798464722047175 e-0009L,
   9.42275906432030735755395494576065434163 3e-0011L,
```

```
 126   -6.711366846653439036162105104991433380926e-0012L,
 127    4.463224090341893165100275380693843116240e-0013L,
 128   -2.783513452582658245422635662559779162312e-0014L,
 129    1.634227412586960195251346878863754661546e-0015L,
 130   -9.060782672889577722765711455623117802795e-0017L,
 131    4.741341801266246873412159213893613602354e-0018L,
 132   -2.272417596497826188374846636534317381203e-0019L,
 133    8.069088733716068462496835658928566920933e-0021L,
 134 };
______unchanged_portion_omitted_
```

```
**********************************************************
    2846 Tue Nov  4 19:00:17 2014
new/usr/src/lib/libm/common/Q/exp10l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak exp10l = __exp10l

#include "libm.h"
#include "longdouble.h"

/*
 * exp10l(x)
 *      n = nint(x*(log10/log2)) ;
 *      exp10(x) = 10**x = exp(x*ln(10)) = exp(n*ln2+(x*ln10-n*ln2))
 *               = 2**n*exp(ln10*(x-n*log2/log10)))
 *      If x is an integer <= M then use repeat multiplication. For
 *      10**M is the largest representable integer, where
 *              M = 10          single precision (24 bits)
 *              M = 22          double precision (53 bits)
 *              M = 48          quadruple precision (113 bits)
 */

#define TINY    1.0e-20L        /* single: 1e-5, double: 1e-10, quad: 1e-20 */
#define LG10OVT 4933.L          /* single:  39, double:  309, quad:  4933 */
#define LG10UFT -4966.L         /* single: -45, double: -323, quad: -4966 */
#define M       48
                        /* logt2hi : last 32 bits is zero for quad prec */
#define LOGT2HI 0.30102999566398119521373889472420986034688L
#define LOGT2LO 2.83166421308946816789666437195 3e-31L

static const long double
        zero    = 0.0L,
        tiny    = TINY * TINY,
        one     = 1.0L,
        lg10    = 3.32192809488736234787031942948939017586 5e+0000L,
        ln10    = 2.30258509299404568401799145468436420760 1e+0000L,
        logt2hi = LOGT2HI,
```



```
        logt2lo = LOGT2LO,
        lg10ovt = LG10OVT,
        lg10uft = LG10UFT;

long double
exp10l(long double x) {
        long double t, tenp;
        int k;

        if (!finitel(x)) {
                if (isnanl(x) || x > zero)
                        return (x + x);
                else
                        return (zero);
        }
        if (fabsl(x) < tiny)
                return (one + x);
        if (x <= lg10ovt)
                if (x >= lg10uft) {
                        k = (int) x;
                        tenp = 10.0L;
                                        /* x is a small +integer */
                        if (0 <= k && k <= M && (long double) k == x) {
                                t = one;
                                if (k & 1)
                                        t *= tenp;
                                k >>= 1;
                                while (k) {
                                        tenp *= tenp;
                                        if (k & 1)
                                                t *= tenp;
                                        k >>= 1;
                                }
                                return (t);
                        }
                        t = anintl(x * lg10);
                        return (scalbnl(expl(ln10 * ((x - t * logt2hi) -
                                t * logt2lo)), (int) t));
                } else
                        return (scalbnl(one, -50000));  /* underflow */
        else
                        return (scalbnl(one, 50000));   /* overflow  */
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2095 Tue Nov  4 19:00:18 2014
new/usr/src/lib/libm/common/Q/exp2l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __exp2l = exp2l
#pragma weak exp2l = __exp2l

#include "libm.h"
#include "longdouble.h"

/*
 *      exp2l(x) = 2**x = 2**((x-anint(x))+anint(x))
 *               = 2**anint(x)*2**(x-anint(x))
 *               = 2**anint(x)*exp((x-anint(x))*ln2)
 */

#define TINY    1.0e-20L        /* single: 1e-5, double: 1e-10, quad: 1e-20 */
#define OVFLEXP 16400           /* single: 130,  double  1030,  quad: 16400 */
#define UNFLEXP -16520          /* single:-155,  double -1080,  quad:-16520 */

static const long double
        zero = 0.0L,
        tiny = TINY * TINY,
        half = 0.5L,
        ln2 = 6.93147180559945309417232121458176568075e-0001L,
        one = 1.0L;

static const int
        ovflexp = OVFLEXP,
        unflexp = UNFLEXP;

long double
exp2l(long double x) {
        long double t;

        if (!finitel(x)) {
```



```
                if (isnanl(x) || x > zero)
                        return (x + x);
                else
                        return (zero);
        }
        t = fabsl(x);
        if (t < half) {
                if (t < tiny)
                        return (one + x);
                else
                        return (expl(ln2 * x));
        }
        t = anintl(x);
        if (t < ovflexp) {
                if (t >= unflexp)
                        return (scalbnl(expl(ln2 * (x - t)), (int) t));
                else
                        return (scalbnl(one, unflexp)); /* underflow */
        } else
                return (scalbnl(one, ovflexp)); /* overflow  */
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3930 Tue Nov  4 19:00:18 2014
new/usr/src/lib/libm/common/Q/expl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * expl(x)
 * Table driven method
 * Written by K.C. Ng, November 1988.
 * Algorithm :
 *      1. Argument Reduction: given the input x, find r and integer k
 *         and j such that
 *                     x = (32k+j)*ln2 + r,  |r| <= (1/64)*ln2 .
 *
 *      2. expl(x) = 2^k * (2^(j/32) + 2^(j/32)*expm1(r))
 *         Note:
 *         a. expm1(r) = (2r)/(2-R), R = r - r^2*(t1 + t2*r^2)
 *         b. 2^(j/32) is represented as
 *                      _TBL_expl_hi[j]+_TBL_expl_lo[j]
 *         where
 *              _TBL_expl_hi[j] = 2^(j/32) rounded
 *              _TBL_expl_lo[j] = 2^(j/32) - _TBL_expl_hi[j].
 *
 * Special cases:
 *      expl(INF) is INF, expl(NaN) is NaN;
 *      expl(-INF)=  0;
 *      for finite argument, only expl(0)=1 is exact.
 *
 * Accuracy:
 *      according to an error analysis, the error is always less than
 *      an ulp (unit in the last place).
 *
 * Misc. info.
 *      For 113 bit long double
 *              if x >  1.13565234062941439494919310779707634284 5e+4
 *      then expl(x) overflow;
 *              if x < -1.14334627433362978788372438434526215 0341e+4
```



```
 *      then expl(x) underflow
 *
 * Constants:
 * Only decimal values are given. We assume that the compiler will convert
 * from decimal to binary accurately enough to produce the correct
 * hexadecimal values.
 */

#pragma weak __expl = expl
#pragma weak expl = __expl

#include "libm.h"

extern const long double _TBL_expl_hi[], _TBL_expl_lo[];

static const long double
one             =  1.0L,
two             =  2.0L,
ln2_64          =  1.08304246962491454596442518977840089856 8e-2L,
ovflthreshold   =  1.13565234062941439494919310779707634284 5e+4L,
unflthreshold   = -1.14334627433362978788372438434526215 0341e+4L,
invln2_32       =  4.61662413084468290355175897920605483976 5e+1L,
ln2_32hi        =  2.16608493924982909192884985859245151568 8e-2L,
ln2_32lo        =  5.20964350259547565278265415750118673177 9e-27L;

/* rational approximation coeffs for [-(ln2)/64,(ln2)/64] */
static const long double
t1 =   1.66666666666666666666666666760876387437e-1L,
t2 =  -2.77777777777777777777770781209317347875 6e-3L,
t3 =   6.61375661375661348207428093287422120242 4e-5L,
t4 =  -1.65343915339213995416960982274223585112 0e-6L,
t5 =   4.17531485176953975138785211661097379605 3e-8L;

long double
expl(long double x) {
        int *px = (int *) &x, ix, j, k, m;
        long double t, r;

        ix = px[0];                             /* high word of x */
        if (ix >= 0x7fff0000)
                return (x + x);                 /* NaN of +inf */
        if (((unsigned) ix) >= 0xffff0000)
                return (-one / x);              /* NaN or -inf */
        if ((ix & 0x7fffffff) < 0x3fc30000) {
                if ((int) x < 1)
                        return (one + x);       /* |x|<2^-60 */
        }
        if (ix > 0) {
                if (x > ovflthreshold)
                        return (scalbnl(x, 20000));
                k = (int) (invln2_32 * (x + ln2_64));
        } else {
                if (x < unflthreshold)
                        return (scalbnl(-x, -40000));
                k = (int) (invln2_32 * (x - ln2_64));
        }
        j  = k&0x1f;
        m  = k>>5;
        t  = (long double) k;
        x  = (x - t * ln2_32hi) - t * ln2_32lo;
        t  = x * x;
        r  = (x - t * (t1 + t * (t2 + t * (t3 + t * (t4 + t * t5))))) - two;
        x  = _TBL_expl_hi[j] - ((_TBL_expl_hi[j] * (x + x)) / r -
                _TBL_expl_lo[j]);
        return (scalbnl(x, m));
}
```

```
**********************************************************
    5997 Tue Nov  4 19:00:19 2014
new/usr/src/lib/libm/common/Q/expm1l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __expm1l = expm1l

#pragma weak expm1l = __expm1l
#if !defined(__sparc)
#error Unsupported architecture
#endif

/*
 * expm1l(x)
 *
 * Table driven method
 * Written by K.C. Ng, June 1995.
 * Algorithm :
 *      1. expm1(x) = x if x<2**-114
 *      2. if |x| <= 0.0625 = 1/16, use approximation
 *              expm1(x) = x + x*P/(2-P)
 * where
 *      P = x - z*(P1+z*(P2+z*(P3+z*(P4+z*(P5+z*P6+z*P7))))), z = x*x;
 * (this formula is derived from
 *      2-P+x = R = x*(exp(x)+1)/(exp(x)-1) ~ 2 + x*x/6 - x^4/360 + ...)
 *
 * P1 =   1.66666666666666666666666666666638500528074603030e-0001
 * P2 =  -2.77777777777777777777777759668391122822266551158e-0003
 * P3 =   6.61375661375661375657437408890138814721051293054e-0005
 * P4 =  -1.65343915343915303310185228411892601606669528828e-0006
 * P5 =   4.17535139755122945763580609663414647067443411178e-0008
 * P6 =  -1.05683795988668526689182102605260986731620026832e-0009
 * P7 =   2.67544168821852702827123344217198187229611470514e-0011
 *
 * Accuracy: |R-x*(exp(x)+1)/(exp(x)-1)|<=2**-119.13
 *
 *      3. For 1/16 < |x| < 1.125, choose x(+-i) ~ +-(i+4.5)/64, i=0,..,67
```



```
 *         since
 *              exp(x) = exp(xi+(x-xi))= exp(xi)*exp((x-xi))
 *         we have
 *              expm1(x) = expm1(xi)+(exp(xi))*(expm1(x-xi))
 *         where
 *              |s=x-xi| <= 1/128
 *         and
 *      expm1(s)=2s/(2-R), R= s-s^2*(T1+s^2*(T2+s^2*(T3+s^2*(T4+s^2*T5))))
 *
 * T1 =   1.66666666666666666666666666660876387437e-1L,
 * T2 =  -2.77777777777777777777770781209317347875 6e-3L,
 * T3 =   6.61375661375661348207428093287422120 2424e-5L,
 * T4 =  -1.65343915339213995416960982274223585 1120e-6L,
 * T5 =   4.17531485176953975138785211661097379 6053e-8L;
 *
 *      4. For |x| >= 1.125, return exp(x)-1.
 *          (see algorithm for exp)
 *
 * Special cases:
 *      expm1l(INF) is INF, expm1l(NaN) is NaN;
 *      expm1l(-INF)= -1;
 *      for finite argument, only expm1l(0)=0 is exact.
 *
 * Accuracy:
 *      according to an error analysis, the error is always less than
 *      2 ulp (unit in the last place).
 *
 * Misc. info.
 *      For 113 bit long double
 *              if x >  1.13565234062941439494919310779707 6342845e+4
 *      then expm1l(x) overflow;
 *
 * Constants:
 * Only decimal values are given. We assume that the compiler will convert
 * from decimal to binary accurately enough to produce the correct
 * hexadecimal values.
 */

#include "libm.h"

extern const long double _TBL_expl_hi[], _TBL_expl_lo[];
extern const long double _TBL_expm1lx[], _TBL_expm1l[];

static const long double
        zero            = +0.0L,
        one             = +1.0L,
        two             = +2.0L,
        ln2_64          = +1.08304246962491454596442518977840089856 8e-2L,
        ovflthreshold   = +1.13565234062941439494919310779707 6342845e+4L,
        invln2_32       = +4.61662413084468290355175897920605483976 5e+1L,
        ln2_32hi        = +2.16608493924982909192884985859245151568 8e-2L,
        ln2_32lo        = +5.20964350259547565278265415750118673177 9e-27L,
        huge            = +1.0e4000L,
        tiny            = +1.0e-4000L,
        P1 = +1.66666666666666666666666666638500528074603030e-0001L,
        P2 = -2.77777777777777777777777759668391122822266551158e-0003L,
        P3 = +6.61375661375661375657437408890138814721051293054e-0005L,
        P4 = -1.65343915343915303310185228411892601606669528828e-0006L,
        P5 = +4.17535139755122945763580609663414647067443411178e-0008L,
        P6 = -1.05683795988668526689182102605260986731620026832e-0009L,
        P7 = +2.67544168821852702827123344217198187229611470514e-0011L,
/* rational approximation coeffs for [-(ln2)/64,(ln2)/64] */
        T1 = +1.66666666666666666666666666660876387437e-1L,
        T2 = -2.77777777777777777777770781209317347875 6e-3L,
        T3 = +6.61375661375661348207428093287422120 2424e-5L,
        T4 = -1.65343915339213995416960982274223585 1120e-6L,
```

```
 127         T5 = +4.17531485176953975138785211661097379605 3e-8L;

 129 long double
 130 expm1l(long double x) {
 131         int hx, ix, j, k, m;
 132         long double t, r, s, w;

 134         hx = ((int *) &x)[HIXWORD];
 135         ix = hx & ~0x80000000;
 136         if (ix >= 0x7fff0000) {
 137                 if (x != x)
 138                         return (x + x); /* NaN */
 139                 if (x < zero)
 140                         return (-one);  /* -inf */
 141                 return (x);     /* +inf */
 142         }
 143         if (ix < 0x3fff4000) {  /* |x| < 1.25 */
 144                 if (ix < 0x3ffb0000) {  /* |x| < 0.0625 */
 145                         if (ix < 0x3f8d0000) {
 146                                 if ((int) x == 0)
 147                                         return (x);     /* |x|<2^-114 */
 148                         }
 149                         t = x * x;
 150                         r = (x - t * (P1 + t * (P2 + t * (P3 + t * (P4 + t *
 151                                 (P5 + t * (P6 + t * P7)))))));
 152                         return (x + (x * r) / (two - r));
 153                 }
 154                 /* compute i = [64*x] */
 155                 m = 0x4009 - (ix >> 16);
 156                 j = ((ix & 0x0000ffff) | 0x10000) >> m; /* j=4,...,67 */
 157                 if (hx < 0)
 158                         j += 82;                        /* negative */
 159                 s = x - _TBL_expm1lx[j];
 160                 t = s * s;
 161                 r = s - t * (T1 + t * (T2 + t * (T3 + t * (T4 + t * T5))));
 162                 r = (s + s) / (two - r);
 163                 w = _TBL_expm1l[j];
 164                 return (w + (w + one) * r);
 165         }
 166         if (hx > 0) {
 167                 if (x > ovflthreshold)
 168                         return (huge * huge);
 169                 k = (int) (invln2_32 * (x + ln2_64));
 170         } else {
 171                 if (x < -80.0)
 172                         return (tiny - x / x);
 173                 k = (int) (invln2_32 * (x - ln2_64));
 174         }
 175         j = k & 0x1f;
 176         m = k >> 5;
 177         t = (long double) k;
 178         x = (x - t * ln2_32hi) - t * ln2_32lo;
 179         t = x * x;
 180         r = (x - t * (T1 + t * (T2 + t * (T3 + t * (T4 + t * T5))))) - two;
 181         x = _TBL_expl_hi[j] - ((_TBL_expl_hi[j] * (x + x)) / r -
 182                 _TBL_expl_lo[j]);
 183         return (scalbnl(x, m) - one);
 184 }
```

*_____unchanged_portion_omitted_*

****
1142 Tue Nov 4 19:00:20 2014
new/usr/src/lib/libm/common/Q/fabsl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fabsl = fabsl
#pragma weak fabsl = __fabsl

#include "libm.h"

long double
fabsl(long double x) {
        int *px = (int *) &x;

        px[0] &= 0x7fffffff;
        return (x);
}
```

_____***unchanged_portion_omitted_***

****
1412 Tue Nov 4 19:00:20 2014
new/usr/src/lib/libm/common/Q/finitel.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak finitel = __finitel

#include "libm.h"

#if defined(__sparc)
int
finitel(long double x) {
        int *px = (int *) &x;
        return ((px[0] & ~0x80000000) < 0x7fff0000);
}
```

_____unchanged_portion_omitted_

****
1849 Tue Nov 4 19:00:20 2014
new/usr/src/lib/libm/common/Q/floorl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * ceill(x)     return the biggest integral value below x
 * floorl(x)    return the least integral value above x
 *
 * NOTE: aintl(x), anintl(x), ceill(x), floorl(x), and rintl(x) return result
 * with the same sign as x's,  including 0.0.
 */

#pragma weak __ceill = ceill
#pragma weak __floorl = floorl
#pragma weak ceill = __ceill
#pragma weak floorl = __floorl

#include "libm.h"
#include "longdouble.h"

static const long double qone = 1.0L;

long double
ceill(long double x) {
        long double t;

        if (!finitel(x))
                return (x + x);
        t = rintl(x);
        if (t >= x)                     /* already ceil(x) */
                return (t);
        else                            /* t < x case: return t+1  */
                return (copysignl(t + qone, x));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5771 Tue Nov  4 19:00:21 2014
new/usr/src/lib/libm/common/Q/fmodl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fmodl = fmodl
#pragma weak fmodl = __fmodl

#include "libm.h"

static const int
        is = -0x7fffffff - 1,
        im = 0x0000ffff,
        iu = 0x00010000;

static const long double
        zero = 0.0L,
        one = 1.0L;

#ifdef __LITTLE_ENDIAN
#define __H0(x) *(3 + (int *) &x)
#define __H1(x) *(2 + (int *) &x)
#define __H2(x) *(1 + (int *) &x)
#define __H3(x) *(0 + (int *) &x)
#else
#define __H0(x) *(0 + (int *) &x)
#define __H1(x) *(1 + (int *) &x)
#define __H2(x) *(2 + (int *) &x)
#define __H3(x) *(3 + (int *) &x)
#endif

long double
fmodl(long double x, long double y) {
        long double a, b;
        int n, ix, iy, k, sx;
        int hx;
        int x0, y0, z0, carry;
```



```
        unsigned x1, x2, x3, y1, y2, y3, z1, z2, z3;

        hx = __H0(x);
        x1 = __H1(x);
        x2 = __H2(x);
        x3 = __H3(x);
        y0 = __H0(y);
        y1 = __H1(y);
        y2 = __H2(y);
        y3 = __H3(y);

        sx = hx & 0x80000000;
        x0 = hx ^ sx;
        y0 &= 0x7fffffff;

        /* purge off exception values */
        if (x0 >= 0x7fff0000 || /* !finitel(x) */
            (y0 > 0x7fff0000) || (y0 == 0x7fff0000 && ((y1 | y2 | y3) != 0)) ||
            (y0 | y1 | y2 | y3) == 0)   /* isnanl(y) || y = 0 */
                return ((x * y) / (x * y));
        a = fabsl(x);
        b = fabsl(y);
        if (a <= b) {
                if (a < b)
                        return (x);
                else
                        return (zero * x);
        }
        /* determine ix = ilogbl(x) */
        if (x0 < iu) {          /* subnormal x */
                ix = -16382;
                while (x0 == 0) {
                        ix -= 16;
                        x0 = x1 >> 16;
                        x1 = (x1 << 16) | (x2 >> 16);
                        x2 = (x2 << 16) | (x3 >> 16);
                        x3 = (x3 << 16);
                }
                while (x0 < iu) {
                        ix -= 1;
                        x0 = (x0 << 1) | (x1 >> 31);
                        x1 = (x1 << 1) | (x2 >> 31);
                        x2 = (x2 << 1) | (x3 >> 31);
                        x3 <<= 1;
                }
        } else {
                ix = (x0 >> 16) - 16383;
                x0 = iu | (x0 & im);
        }

        /* determine iy = ilogbl(y) */
        if (y0 < iu) {          /* subnormal y */
                iy = -16382;
                while (y0 == 0) {
                        iy -= 16;
                        y0 = y1 >> 16;
                        y1 = (y1 << 16) | (y2 >> 16);
                        y2 = (y2 << 16) | (y3 >> 16);
                        y3 = (y3 << 16);
                }
                while (y0 < iu) {
                        iy -= 1;
                        y0 = (y0 << 1) | (y1 >> 31);
                        y1 = (y1 << 1) | (y2 >> 31);
                        y2 = (y2 << 1) | (y3 >> 31);
                        y3 <<= 1;
```

```
127                 }
128         } else {
129                 iy = (y0 >> 16) - 16383;
130                 y0 = iu | (y0 & im);
131         }

133         /* fix point fmod */
134         n = ix - iy;
135         while (n--) {
136                 while (x0 == 0 && n >= 16) {
137                         n -= 16;
138                         x0 = x1 >> 16;
139                         x1 = (x1 << 16) | (x2 >> 16);
140                         x2 = (x2 << 16) | (x3 >> 16);
141                         x3 = (x3 << 16);
142                 }
143                 while (x0 < iu && n >= 1) {
144                         n -= 1;
145                         x0 = (x0 << 1) | (x1 >> 31);
146                         x1 = (x1 << 1) | (x2 >> 31);
147                         x2 = (x2 << 1) | (x3 >> 31);
148                         x3 = (x3 << 1);
149                 }
150                 carry = 0;
151                 z3 = x3 - y3;
152                 carry = (z3 > x3);
153                 if (carry == 0) {
154                         z2 = x2 - y2;
155                         carry = (z2 > x2);
156                 } else {
157                         z2 = x2 - y2 - 1;
158                         carry = (z2 >= x2);
159                 }
160                 if (carry == 0) {
161                         z1 = x1 - y1;
162                         carry = (z1 > x1);
163                 } else {
164                         z1 = x1 - y1 - 1;
165                         carry = (z1 >= x1);
166                 }
167                 z0 = x0 - y0 - carry;
168                 if (z0 < 0) {   /* double x */
169                         x0 = x0 + x0 + ((x1 & is) != 0);
170                         x1 = x1 + x1 + ((x2 & is) != 0);
171                         x2 = x2 + x2 + ((x3 & is) != 0);
172                         x3 = x3 + x3;
173                 } else {
174                         if (z0 == 0) {
175                                 if ((z1 | z2 | z3) == 0) {      /* 0: done */
176                                         __H0(a) = hx & is;
177                                         __H1(a) = __H2(a) = __H3(a) = 0;
178                                         return (a);
179                                 }
180                         }
181                         /* x = z << 1 */
182                         z0 = z0 + z0 + ((z1 & is) != 0);
183                         z1 = z1 + z1 + ((z2 & is) != 0);
184                         z2 = z2 + z2 + ((z3 & is) != 0);
185                         z3 = z3 + z3;
186                         x0 = z0;
187                         x1 = z1;
188                         x2 = z2;
189                         x3 = z3;
190                 }
191         }
```



```
193         carry = 0;
194         z3 = x3 - y3;
195         carry = (z3 > x3);
196         if (carry == 0) {
197                 z2 = x2 - y2;
198                 carry = (z2 > x2);
199         } else {
200                 z2 = x2 - y2 - 1;
201                 carry = (z2 >= x2);
202         }
203         if (carry == 0) {
204                 z1 = x1 - y1;
205                 carry = (z1 > x1);
206         } else {
207                 z1 = x1 - y1 - 1;
208                 carry = (z1 >= x1);
209         }
210         z0 = x0 - y0 - carry;
211         if (z0 >= 0) {
212                 x0 = z0;
213                 x1 = z1;
214                 x2 = z2;
215                 x3 = z3;
216         }
217         /* convert back to floating value and restore the sign */
218         if ((x0 | x1 | x2 | x3) == 0) {
219                 __H0(a) = hx & is;
220                 __H1(a) = __H2(a) = __H3(a) = 0;
221                 return (a);
222         }
223         while (x0 < iu) {
224                 if (x0 == 0) {
225                         iy -= 16;
226                         x0 = x1 >> 16;
227                         x1 = (x1 << 16) | (x2 >> 16);
228                         x2 = (x2 << 16) | (x3 >> 16);
229                         x3 = (x3 << 16);
230                 } else {
231                         x0 = x0 + x0 + ((x1 & is) != 0);
232                         x1 = x1 + x1 + ((x2 & is) != 0);
233                         x2 = x2 + x2 + ((x3 & is) != 0);
234                         x3 = x3 + x3;
235                         iy -= 1;
236                 }
237         }

239         /* normalize output */
240         if (iy >= -16382) {
241                 __H0(a) = sx | (x0 - iu) | ((iy + 16383) << 16);
242                 __H1(a) = x1;
243                 __H2(a) = x2;
244                 __H3(a) = x3;
245         } else {                /* subnormal output */
246                 n = -16382 - iy;
247                 k = n & 31;
248                 if (k != 0) {
249                         if (k <= 16) {
250                                 x3 = (x2 << (32 - k)) | (x3 >> k);
251                                 x2 = (x1 << (32 - k)) | (x2 >> k);
252                                 x1 = (x0 << (32 - k)) | (x1 >> k);
253                                 x0 >>= k;
254                         } else {
255                                 x3 = (x2 << (32 - k)) | (x3 >> k);
256                                 x2 = (x1 << (32 - k)) | (x2 >> k);
257                                 x1 = (x0 << (32 - k)) | (x1 >> k);
258                                 x0 = 0;
```

```
259                         }
260                 }
261                 while (n >= 32) {
262                         n -= 32;
263                         x3 = x2;
264                         x2 = x1;
265                         x1 = x0;
266                         x0 = 0;
267                 }
268                 __H0(a) = x0 | sx;
269                 __H1(a) = x1;
270                 __H2(a) = x2;
271                 __H3(a) = x3;
272                 a *= one;
273         }
274         return (a);
275 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1321 Tue Nov  4 19:00:21 2014
new/usr/src/lib/libm/common/Q/gammal.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __gammal = gammal
#pragma weak gammal = __gammal

/*
 * long double gammal(long double x);
 */

#include "libm.h"
#include "longdouble.h"

extern int signgam;
extern int signgaml;

long double
gammal(long double x) {
        long double y = __k_lgammal(x, &signgaml);

        signgam = signgaml;     /* SUSv3 requires the setting of signgam */
        return (y);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1233 Tue Nov  4 19:00:22 2014
new/usr/src/lib/libm/common/Q/gammal_r.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double gammal_r(long double x, int *signgamlp);
 */

#pragma weak __gammal_r = gammal_r
#pragma weak gammal_r = __gammal_r

#include "libm.h"
#include "longdouble.h"

long double
gammal_r(long double x, int *signgamlp) {
        return (__k_lgammal(x, signgamlp));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    3902 Tue Nov  4 19:00:22 2014
new/usr/src/lib/libm/common/Q/hypotl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __hypotl = hypotl
#pragma weak hypotl = __hypotl

/*
 * long double hypotl(long double x, long double y);
 * Method :
 *      If z=x*x+y*y has error less than sqrt(2)/2 ulp than sqrt(z) has
 *      error less than 1 ulp.
 *      So, compute sqrt(x*x+y*y) with some care as follows:
 *      Assume x>y>0;
 *      1. save and set rounding to round-to-nearest
 *      2. if x > 2y  use
 *              x1*x1+(y*y+(x2*(x+x2))) for x*x+y*y
 *      where x1 = x with lower 64 bits cleared, x2 = x-x1; else
 *      3. if x <= 2y use
 *              t1*y1+((x-y)*(x-y)+(t1*y2+t2*y))
 *      where t1 = 2x with lower 64 bits cleared, t2 = 2x-t1, y1= y with
 *      lower 64 bits chopped, y2 = y-y1.
 *
 *      NOTE: DO NOT remove parenthsis!
 *
 * Special cases:
 *      hypot(x,y) is INF if x or y is +INF or -INF; else
 *      hypot(x,y) is NAN if x or y is NAN.
 *
 * Accuracy:
 *      hypot(x,y) returns sqrt(x^2+y^2) with error less than 1 ulps (units
 *      in the last place)
 */

#include "libm.h"
#include "longdouble.h"
```



```
extern enum fp_direction_type __swapRD(enum fp_direction_type);

static const long double zero = 0.0L, one = 1.0L;

long double
hypotl(long double x, long double y) {
        int n0, n1, n2, n3;
        long double t1, t2, y1, y2, w;
        int *px = (int *) &x, *py = (int *) &y;
        int *pt1 = (int *) &t1, *py1 = (int *) &y1;
        enum fp_direction_type rd;
        int j, k, nx, ny, nz;

        if ((*(int *) &one) != 0) {     /* determine word ordering */
                n0 = 0;
                n1 = 1;
                n2 = 2;
                n3 = 3;
        } else {
                n0 = 3;
                n1 = 2;
                n2 = 1;
                n3 = 0;
        }

        px[n0] &= 0x7fffffff;   /* clear sign bit of x and y */
        py[n0] &= 0x7fffffff;
        k = 0x7fff0000;
        nx = px[n0] & k;        /* exponent of x and y */
        ny = py[n0] & k;
        if (ny > nx) {
                w = x;
                x = y;
                y = w;
                nz = ny;
                ny = nx;
                nx = nz;
        }                       /* force x > y */
        if ((nx - ny) >= 0x00730000)
                return (x + y); /* x/y >= 2**116 */
        if (nx < 0x5ff30000 && ny > 0x205b0000) {       /* medium x,y */
                /* save and set RD to Rounding to nearest */
                rd = __swapRD(fp_nearest);
                w = x - y;
                if (w > y) {
                        pt1[n0] = px[n0];
                        pt1[n1] = px[n1];
                        pt1[n2] = pt1[n3] = 0;
                        t2 = x - t1;
                        x = sqrtl(t1 * t1 - (y * (-y) - t2 * (x + t1)));
                } else {
                        x = x + x;
                        py1[n0] = py[n0];
                        py1[n1] = py[n1];
                        py1[n2] = py1[n3] = 0;
                        y2 = y - y1;
                        pt1[n0] = px[n0];
                        pt1[n1] = px[n1];
                        pt1[n2] = pt1[n3] = 0;
                        t2 = x - t1;
                        x = sqrtl(t1 * y1 - (w * (-w) - (t2 * y1 + y2 * x)));
                }
                if (rd != fp_nearest)
                        (void) __swapRD(rd);    /* restore rounding mode */
                return (x);
```

```
127         } else {
128                 if (nx == k || ny == k) {       /* x or y is INF or NaN */
129                         if (isinfl(x))
130                                 t2 = x;
131                         else if (isinfl(y))
132                                 t2 = y;
133                         else
134                                 t2 = x + y;     /* invalid if x or y is sNaN */
135                         return (t2);
136                 }
137                 if (ny == 0) {
138                         if (y == zero || x == zero)
139                                 return (x + y);
140                         t1 = scalbnl(one, 16381);
141                         x *= t1;
142                         y *= t1;
143                         return (scalbnl(one, -16381) * hypotl(x, y));
144                 }
145                 j = nx - 0x3fff0000;
146                 px[n0] -= j;
147                 py[n0] -= j;
148                 pt1[n0] = nx;
149                 pt1[n1] = pt1[n2] = pt1[n3] = 0;
150                 return (t1 * hypotl(x, y));
151         }
152 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    2552 Tue Nov  4 19:00:23 2014
new/usr/src/lib/libm/common/Q/ieee_funcl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak isinfl = __isinfl
#pragma weak isnormall = __isnormall
#pragma weak issubnormall = __issubnormall
#pragma weak iszerol = __iszerol
#pragma weak signbitl = __signbitl

#include "libm.h"

#if defined(__sparc)
int
isinfl(long double x) {
        int *px = (int *) &x;
        return ((px[0] & ~0x80000000) == 0x7fff0000 && px[1] == 0 &&
                px[2] == 0 && px[3] == 0);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    2452 Tue Nov  4 19:00:23 2014
new/usr/src/lib/libm/common/Q/ilogbl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ilogbl = ilogbl
#pragma weak ilogbl = __ilogbl

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */

#if defined(__sparc)
#define ISNORMALL(k, x) (k != 0x7fff)                   /* assuming k != 0 */
#define X86PDNRM(k, x)
#define XSCALE_OFFSET   0x406f                          /* 0x3fff + 112 */
static const long double xscale = 5192296858534827628530496329220096.0L;
                                                                /* 2^112 */
#elif defined(__x86)
/*
 * if pseudo-denormal, replace by the equivalent normal
 */
#define X86PDNRM(k, x)  if (k == 0 && (((int *) &x)[1] & 0x80000000) != 0) \
                            ((int *) &x)[2] |= k = 1
#if defined(HANDLE_UNSUPPORTED)                         /* assuming k != 0 */
#define ISNORMALL(k, x) (k != 0x7fff && (((int *) &x)[1] & 0x80000000) != 0)
#else
#define ISNORMALL(k, x) (k != 0x7fff)
#endif
#define XSCALE_OFFSET   0x403e                          /* 0x3fff + 63 */
static const long double xscale = 9223372036854775808.0L;       /* 2^63 */
#endif

static int
raise_invalid(int v) {  /* SUSv3 requires ilogbl(0,+/-Inf,NaN) raise invalid */
#ifndef lint
        if ((__xpg6 & _C99SUSv3_ilogb_0InfNaN_raises_invalid) != 0) {
                static const double zero = 0.0;
```



```
                volatile double dummy;

                dummy = zero / zero;
        }
#endif
        return (v);
}
```

_____***unchanged_portion_omitted_***

****
1581 Tue Nov 4 19:00:24 2014
new/usr/src/lib/libm/common/Q/isnanl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __isnanl = isnanl
#pragma weak isnanl = __isnanl

#include "libm.h"

#if defined(__sparc)
int
isnanl(long double x) {
        int *px = (int *) &x;
        return ((px[0] & ~0x80000000) >= 0x7fff0000 &&
                ((px[0] & ~0xffff0000) | px[1] | px[2] | px[3]) != 0);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   27911 Tue Nov  4 19:00:25 2014
new/usr/src/lib/libm/common/Q/j0l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Floating point Bessel's function of the first and second kinds
 * of order zero: j0(x),y0(x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 */

#pragma weak __j0l = j0l
#pragma weak __y0l = y0l
#pragma weak j0l = __j0l
#pragma weak y0l = __y0l

#include "libm.h"
#include "longdouble.h"

#define GENERIC long double
static const GENERIC
zero    = 0.0L,
small   = 1.0e-9L,
tiny    = 1.0e-38L,
one     = 1.0L,
five    = 5.0L,
eight   = 8.0L,
invsqrtpi= 5.641895835477562869480794515607725858441e-0001L,
tpi     = 0.636619772367581343075535053490057448L;

static GENERIC pzero(GENERIC);
static GENERIC qzero(GENERIC);

static GENERIC r0[7] = {
```



```
   -2.4999999999999999999999999999998934492e-0001L,
    1.2726579273600497863276184511337637148880e-0002L,
   -2.6944997637129632769006366934006596008980e-0004L,
    2.7248774750589775769032340709196164478830e-0006L,
   -1.4326171032143302369674774953930763202810e-0008L,
    3.8232488040800791687066835405137922244710e-0011L,
   -4.1831742775679836473375685042863136650650e-0014L,
};
_____unchanged_portion_omitted_
```

```
**********************************************************
   28240 Tue Nov  4 19:00:25 2014
new/usr/src/lib/libm/common/Q/j1l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 /*
  31  * floating point Bessel's function of the first and second kinds
  32  * of order zero: j1(x),y1(x);
  33  *
  34  * Special cases:
  35  *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
  36  *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
  37  */

  39 #pragma weak __j1l = j1l
  40 #pragma weak __y1l = y1l
  39 #pragma weak j1l = __j1l
  40 #pragma weak y1l = __y1l

  42 #include "libm.h"
  43 #include "longdouble.h"

  45 #define GENERIC long double
  46 static const GENERIC
  47 zero    = 0.0L,
  48 small   = 1.0e-9L,
  49 tiny    = 1.0e-38L,
  50 one     = 1.0L,
  51 five    = 5.0L,
  52 invsqrtpi= 5.641895835477562869480794515607725858441e-0001L,
  53 tpi     = 0.636619772367581343075535053490057448L;

  55 static GENERIC pone(GENERIC);
  56 static GENERIC qone(GENERIC);

  58 static const GENERIC r0[7] = {
  59   -6.249999999999999999999999999999627320e-0002L,
```



```
  60    1.940606727194041716205384618494641565464e-0003L,
  61   -3.005630423155733701856481469986459043883e-0005L,
  62    2.345586219403918667468341047369572169358e-0007L,
  63   -9.976809285885253587529010109133336669724e-0010L,
  64    2.218743258363623946078958783775107473381e-0012L,
  65   -2.071079656218700604767650924103578046280e-0015L,
  66 };
______unchanged_portion_omitted_
```

```
**********************************************************
    7136 Tue Nov  4 19:00:25 2014
new/usr/src/lib/libm/common/Q/jnl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __jnl = jnl
#pragma weak __ynl = ynl
#pragma weak jnl = __jnl
#pragma weak ynl = __ynl

/*
 * floating point Bessel's function of the 1st and 2nd kind
 * of order n: jn(n,x),yn(n,x);
 *
 * Special cases:
 *      y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
 *      y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
 * Note 2. About jn(n,x), yn(n,x)
 *      For n=0, j0(x) is called,
 *      for n=1, j1(x) is called,
 *      for n<x, forward recursion us used starting
 *      from values of j0(x) and j1(x).
 *      for n>x, a continued fraction approximation to
 *      j(n,x)/j(n-1,x) is evaluated and then backward
 *      recursion is used starting from a supposed value
 *      for j(n,x). The resulting value of j(0,x) is
 *      compared with the actual value to correct the
 *      supposed value of j(n,x).
 *
 *      yn(n,x) is similar in all respects, except
 *      that forward recursion is used for all
 *      values of n>1.
 *
 */

#include "libm.h"
#include "longdouble.h"
```



```
#include <float.h>      /* LDBL_MAX */

#define GENERIC long double

static const GENERIC
invsqrtpi = 5.641895835477562869480794515607725858441e-0001L,
two  = 2.0L,
zero = 0.0L,
one  = 1.0L;

GENERIC
jnl(n, x) int n; GENERIC x; {
        int i, sgn;
        GENERIC a, b, temp, z, w;

        /*
         * J(-n,x) = (-1)^n * J(n, x), J(n, -x) = (-1)^n * J(n, x)
         * Thus, J(-n,x) = J(n,-x)
         */
        if (n < 0) {
                n = -n;
                x = -x;
        }
        if (n == 0)
                return (j0l(x));
        if (n == 1)
                return (j1l(x));
        if (x != x)
                return (x+x);
        if ((n&1) == 0)
                sgn = 0;                        /* even n */
        else
                sgn = signbitl(x);      /* old n  */
        x = fabsl(x);
        if (x == zero || !finitel(x)) b = zero;
        else if ((GENERIC)n <= x) {
                                        /*
                                         * Safe to use
                                         * J(n+1,x)=2n/x *J(n,x)-J(n-1,x)
                                         */
            if (x > 1.0e91L) {
                                /*
                                 * x >> n**2
                                 *  Jn(x) = cos(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *   Yn(x) = sin(x-(2n+1)*pi/4)*sqrt(2/x*pi)
                                 *   Let s=sin(x), c=cos(x),
                                 *      xn=x-(2n+1)*pi/4, sqt2 = sqrt(2),then
                                 *
                                 *         n    sin(xn)*sqt2    cos(xn)*sqt2
                                 *      ----------------------------------
                                 *         0     s-c             c+s
                                 *         1    -s-c            -c+s
                                 *         2    -s+c            -c-s
                                 *         3     s+c             c-s
                                 */
                switch (n&3) {
                    case 0: temp =  cosl(x)+sinl(x); break;
                    case 1: temp = -cosl(x)+sinl(x); break;
                    case 2: temp = -cosl(x)-sinl(x); break;
                    case 3: temp =  cosl(x)-sinl(x); break;
                }
                b = invsqrtpi*temp/sqrtl(x);
            } else {
                        a = j0l(x);
                        b = j1l(x);
                        for (i = 1; i < n; i++) {
```

```
126                     temp = b;
127                     b = b*((GENERIC)(i+i)/x) - a; /* avoid underflow */
128                     a = temp;
129                         }
130             }
131         } else {
132             if (x < 1e-17L) {   /* use J(n,x) = 1/n!*(x/2)^n */
133                 b = powl(0.5L*x, (GENERIC)n);
134                 if (b != zero) {
135                     for (a = one, i = 1; i <= n; i++) a *= (GENERIC)i;
136                     b = b/a;
137                 }
138             } else {
139                 /* use backward recurrence */
140                 /*
141                  *                          x      x^2      x^2
142                  *  J(n,x)/J(n-1,x) =  ----   ------   ------   .....
143                  *                          2n  - 2(n+1) - 2(n+2)
144                  *
145                  *                          1      1        1
146                  *  (for large x)   =  ----  ------   ------   .....
147                  *                          2n   2(n+1)   2(n+2)
148                  *                          -- - ------ - ------ -
149                  *                           x     x         x
150                  *
151                  * Let w = 2n/x and h=2/x, then the above quotient
152                  * is equal to the continued fraction:
153                  *                     1
154                  *      = -----------------------
155                  *                    1
156                  *         w - -----------------
157                  *                       1
158                  *              w+h - ---------
159                  *                    w+2h - ...
160                  *
161                  * To determine how many terms needed, let
162                  * Q(0) = w, Q(1) = w(w+h) - 1,
163                  * Q(k) = (w+k*h)*Q(k-1) - Q(k-2),
164                  * When Q(k) > 1e4      good for single
165                  * When Q(k) > 1e9      good for double
166                  * When Q(k) > 1e17     good for quaduple
167                  */
168             /* determin k */
169                 GENERIC t, v;
170                 double q0, q1, h, tmp; int k, m;
171                 w  = (n+n)/(double)x; h = 2.0/(double)x;
172                 q0 = w;  z = w+h; q1 = w*z - 1.0; k = 1;
173                 while (q1 < 1.0e17) {
174                         k += 1; z += h;
175                         tmp = z*q1 - q0;
176                         q0 = q1;
177                         q1 = tmp;
178                 }
179                 m = n+n;
180                 for (t = zero, i = 2*(n+k); i >= m; i -= 2) t = one/(i/x-t);
181                 a = t;
182                 b = one;
183                 /*
184                  * estimate log((2/x)^n*n!) = n*log(2/x)+n*ln(n)
185                  * hence, if n*(log(2n/x)) > ...
186                  *  single 8.8722839355e+01
187                  *  double 7.09782712893383973096e+02
188                  *  long double 1.13565234062941439494919310779707650060170e+04
189                  *  then recurrent value may overflow and the result is
190                  *  likely underflow to zero
191                  */
```



```
192                 tmp = n;
193                 v = two/x;
194                 tmp = tmp*logl(fabsl(v*tmp));
195                 if (tmp < 1.13565234062941439494919310779707650765e+04L) {
196                         for (i = n-1; i > 0; i--) {
197                     temp = b;
198                     b = ((i+i)/x)*b - a;
199                     a = temp;
200                 }
201                 } else {
202                         for (i = n-1; i > 0; i--) {
203                     temp = b;
204                     b = ((i+i)/x)*b - a;
205                     a = temp;
206                     if (b > 1e1000L) {
207                         a /= b;
208                         t /= b;
209                         b  = 1.0;
210                     }
211                 }
212                 }
213                 b = (t*j0l(x)/b);
214             }
215         }
216         if (sgn == 1)
217                 return (-b);
218         else
219                 return (b);
220 }
```

unchanged_portion_omitted_

****
1325 Tue Nov 4 19:00:26 2014
new/usr/src/lib/libm/common/Q/lgammal.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgammal = lgammal
#pragma weak lgammal = __lgammal

/*
 * long double lgammal(long double x);
 */

#include "libm.h"
#include "longdouble.h"

extern int signgam;
extern int signgaml;

long double
lgammal(long double x) {
        long double y = __k_lgammal(x, &signgaml);

        signgam = signgaml;     /* SUSv3 requires the setting of signgam */
        return (y);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1237 Tue Nov  4 19:00:26 2014
new/usr/src/lib/libm/common/Q/lgammal_r.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * long double lgammal_r(long double x, int *signgamlp);
 */

#pragma weak __lgammal_r = lgammal_r
#pragma weak lgammal_r = __lgammal_r

#include "libm.h"
#include "longdouble.h"

long double
lgammal_r(long double x, int *signgamlp) {
        return (__k_lgammal(x, signgamlp));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    3044 Tue Nov  4 19:00:27 2014
new/usr/src/lib/libm/common/Q/log10l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log10l = log10l
#pragma weak log10l = __log10l

/*
 * log10l(X)
 *
 * Method :
 *      Let log10_2hi = leading 98(SPARC)/49(x86) bits of log10(2) and
 *          log10_2lo = log10(2) - log10_2hi,
 *          ivln10   = 1/log(10) rounded.
 *      Then
 *              n = ilogb(x),
 *              if (n<0)  n = n+1;
 *              x = scalbn(x,-n);
 *              LOG10(x) := n*log10_2hi + (n*log10_2lo + ivln10*log(x))
 *
 * Note1:
 *      For fear of destroying log10(10**n)=n, the rounding mode is
 *      set to Round-to-Nearest.
 *
 * Special cases:
 *      log10(x) is NaN with signal if x < 0;
 *      log10(+INF) is +INF with no signal; log10(0) is -INF with signal;
 *      log10(NaN) is that NaN with no signal;
 *      log10(10**N) = N  for N=0,1,...,22.
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */
```



```
#include "libm.h"
#include "longdouble.h"

#if defined(__x86)
#define __swapRD        __swap87RD
#endif
extern enum fp_direction_type __swapRD(enum fp_direction_type);

static const long double
        zero      = 0.0L,
        ivln10    = 4.34294481903251827651128918916605082294e-0001L,
        one       = 1.0L,
#if defined(__x86)
        log10_2hi = 3.01029995663980365350198553642258048057e-01L,
        log10_2lo = 8.29863540341082234978710633729118358541e-16L;
#elif defined(__sparc)
        log10_2hi = 3.01029956639811952137388947242098603469e-01L,
        log10_2lo = 2.83166421308946816789666437195321094566e-31L;
#else
#error Unknown Architecture!
#endif

long double
log10l(long double x) {
        long double y, z;
        enum fp_direction_type rd;
        int n;

        if (!finitel(x))
                return (x + fabsl(x));  /* x is +-INF or NaN */
        else if (x > zero) {
                n = ilogbl(x);
                if (n < 0)
                        n += 1;
                rd = __swapRD(fp_nearest);
                y = n;
                x = scalbnl(x, -n);
                z = y * log10_2lo + ivln10 * logl(x);
                z += y * log10_2hi;
                if (rd != fp_nearest)
                        (void) __swapRD(rd);
                return (z);
        } else if (x == zero)   /* -INF */
                return (-one / zero);
        else                    /* x <0, return NaN */
                return (zero / zero);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6978 Tue Nov  4 19:00:27 2014
new/usr/src/lib/libm/common/Q/log1pl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifdef __LITTLE_ENDIAN
#define H0(x)   *(3 + (int *) &x)
#define H1(x)   *(2 + (int *) &x)
#define H2(x)   *(1 + (int *) &x)
#define H3(x)   *(int *) &x
#else
#define H0(x)   *(int *) &x
#define H1(x)   *(1 + (int *) &x)
#define H2(x)   *(2 + (int *) &x)
#define H3(x)   *(3 + (int *) &x)
#endif

/*
 * log1pl(x)
 * Table look-up algorithm by modifying logl.c
 * By K.C. Ng, July 6, 1995
 *
 * (a). For 1+x in [31/33,33/31], using a special approximation:
 *      s = x/(2.0+x);  ... here |s| <= 0.03125
 *      z = s*s;
 *      return x-s*(x-z*(B1+z*(B2+z*(B3+z*(B4+...+z*B9)...))));
 *      (i.e., x is in [-2/33,2/31])
 *
 * (b). Otherwise, normalize 1+x = 2^n * 1.f.
 *      Here we may need a correction term for 1+x rounded.
 *      Use a 6-bit table look-up: find a 6 bit g that match f to 6.5 bits,
 *      then
 *          log(1+x) = n*ln2 + log(1.g) + log(1.f/1.g).
 *      Here the leading and trailing values of log(1.g) are obtained from
 *      a size-64 table.
 *      For log(1.f/1.g), let s = (1.f-1.g)/(1.f+1.g). Note that
 *              1.f = 2^-n(1+x)
```



```
 *
 *      then
 *          log(1.f/1.g) = log((1+s)/(1-s)) = 2s + 2/3 s^3 + 2/5 s^5 +...
 *      Note that |s|<2**-8=0.00390625. We use an odd s-polynomial
 *      approximation to compute log(1.f/1.g):
 *              s*(A1+s^2*(A2+s^2*(A3+s^2*(A4+s^2*(A5+s^2*(A6+s^2*A7))))))
 *      (Precision is 2**-136.91 bits, absolute error)
 *
 *      CAUTION:
 *      For x>=1, compute 1+x will lost one bit (OK).
 *      For x in [-0.5,-1), 1+x is exact.
 *      For x in (-0.5,-2/33]U[2/31,1), up to 4 last bits of x will be lost
 *      in 1+x.  Therefore, to recover the lost bits, one need to compute
 *      1.f-1.g accurately.
 *
 *      Let hx = HI(x), m = (hx>>16)-0x3fff (=ilogbl(x)), note that
 *              -2/33 = -0.0606...= 2^-5 * 1.939...,
 *               2/31 =  0.09375  = 2^-4 * 1.500...,
 *      so for x in (-0.5,-2/33], -5<=m<=-2,  n= -1, 1+f=2*(1+x)
 *         for x in [2/33,1),     -4<=m<=-1,  n=  0, f=x
 *
 *      In short:
 *      if x>0, let g: hg= ((hx + (0x200<<(-m)))>>(10-m))<<(10-m)
 *      then 1.f-1.g = x-g
 *      if x<0, let g': hg' =((ix-(0x200)<<(-m-1))>>(9-m))<<(9-m)
 *      (ix=hx&0x7fffffff)
 *      then 1.f-1.g = 2*(g'+x),
 *
 * (c). The final result is computed by
 *              (n*ln2_hi+_TBL_logl_hi[j]) +
 *                      ( (n*ln2_lo+_TBL_logl_lo[j]) + s*(A1+...) )
 *
 * Note.
 *      For ln2_hi and _TBL_logl_hi[j], we force their last 32 bit to be zero
 *      so that n*ln2_hi + _TBL_logl_hi[j] is exact. Here
 *      _TBL_logl_hi[j] + _TBL_logl_lo[j] match log(1+j*2**-6) to 194 bits
 *
 *
 * Special cases:
 *      log(x) is NaN with signal if x < 0 (including -INF) ;
 *      log(+INF) is +INF; log(0) is -INF with signal;
 *      log(NaN) is that NaN with no signal.
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */

#pragma weak __log1pl = log1pl
#pragma weak log1pl = __log1pl

#include "libm.h"

extern const long double _TBL_logl_hi[], _TBL_logl_lo[];

static const long double
zero    =   0.0L,
one     =   1.0L,
two     =   2.0L,
ln2hi   =   6.9314718055994530941723195474958448502030e-0001L,
ln2lo   =   1.6670859208305522088904493304003797541690e-0025L,
A1      =   2.0000000000000000000000000000000000000024e+0000L,
A2      =   6.6666666666666666666666666666666091393804e-0001L,
A3      =   4.0000000000000000000000004071670706220671e-0001L,
```

```
A4      =   2.85714285714285714273007749061290368116 4e-0001L,
A5      =   2.22222222222224257770283692081288260509 9e-0001L,
A6      =   1.81818181643549339598591266710588582835 6e-0001L,
A7      =   1.53853783521183975111206751280549693172 5e-0001L,
B1      =   6.66666666666666666666666666666961498329 e-0001L,
B2      =   3.99999999999999999999999999037655042358 e-0001L,
B3      =   2.85714285714285714285727342642834745791 8e-0001L,
B4      =   2.22222222222222222135322904974791010956 6e-0001L,
B5      =   1.81818181818182150353255930630907013804 6e-0001L,
B6      =   1.53846153845380921048635608458735678855 6e-0001L,
B7      =   1.33333334446335875612145689264517879548 0e-0001L,
B8      =   1.17646090478389906485464517460336038379 2e-0001L,
B9      =   1.05729386995659899532636860251805699074 6e-0001L;

long double
log1pl(long double x) {
        long double f, s, z, qn, h, t, y, g;
        int i, j, ix, iy, n, hx, m;

        hx = H0(x);
        ix = hx & 0x7fffffff;
        if (ix < 0x3ffaf07c) {  /* |x|<2/33 */
                if (ix <= 0x3f8d0000) { /* x <= 2**-114, return x */
                        if ((int) x == 0)
                                return (x);
                }
                s = x / (two + x);      /* |s|<2**-8 */
                z = s * s;
                return (x - s * (x - z * (B1 + z * (B2 + z * (B3 + z * (B4 +
                    z * (B5 + z * (B6 + z * (B7 + z * (B8 + z * B9)))))))));
        }
        if (ix >= 0x7fff0000) { /* x is +inf or NaN */
                return (x + fabsl(x));
        }
        if (hx < 0 && ix >= 0x3fff0000) {
                if (ix > 0x3fff0000 || (H1(x) | H2(x) | H3(x)) != 0)
                        x = zero;
                return (x / zero);      /* log1p(x) is NaN  if x<-1 */
                /* log1p(-1) is -inf */
        }
        if (ix >= 0x7feeffff)
                y = x;          /* avoid spurious overflow */
        else
                y = one + x;
        iy = H0(y);
        n = ((iy + 0x200) >> 16) - 0x3fff;
        iy = (iy & 0x0000ffff) | 0x3fff0000;    /* scale 1+x to [1,2] */
        H0(y) = iy;
        z = zero;
        m = (ix >> 16) - 0x3fff;
        /* HI(1+x) = (((hx&0xffff)|0x10000)>>(-m))|0x3fff0000 */
        if (n == 0) {           /* x in [2/33,1) */
                g = zero;
                H0(g) = ((hx + (0x200 << (-m))) >> (10 - m)) << (10 - m);
                t = x - g;
                i = (((((hx & 0xffff) | 0x10000) >> (-m)) | 0x3fff0000) +
                        0x200) >> 10;
                H0(z) = i << 10;

        } else if ((1 + n) == 0 && (ix < 0x3ffe0000)) { /* x in (-0.5,-2/33] */
                g = zero;
                H0(g) = ((ix + (0x200 << (-m - 1))) >> (9 - m)) << (9 - m);
                t = g + x;
                t = t + t;
                /*
                 * HI(2*(1+x)) =
```



```
                 * ((0x10000-(((hx&0xffff)|0x10000)>>(-m)))<<1)|0x3fff0000
                 */
                /*
                 * i =
                 * ((((0x10000-(((hx&0xffff)|0x10000)>>(-m)))<<1)|0x3fff0000)+
                 * 0x200)>>10; H0(z)=i<<10;
                 */
                z = two * (one - g);
                i = H0(z) >> 10;
        } else {
                i = (iy + 0x200) >> 10;
                H0(z) = i << 10;
                t = y - z;
        }

        s = t / (y + z);
        j = i & 0x3f;
        z = s * s;
        qn = (long double) n;
        t = qn * ln2lo + _TBL_logl_lo[j];
        h = qn * ln2hi + _TBL_logl_hi[j];
        f = t + s * (A1 + z * (A2 + z * (A3 + z * (A4 + z * (A5 + z * (A6 +
                z * A7)))))));
        return (h + f);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1606 Tue Nov  4 19:00:27 2014
new/usr/src/lib/libm/common/Q/log2l.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * log2l(x)
 * RETURN THE BASE 2 LOGARITHM OF X
 *
 * Method:
 *      purge off 0,INF, and NaN.
 *      n = ilogb(x)
 *      if (n<0) n+=1
 *      z = scalbn(x,-n)
 *      LOG2(x) = n + (1/ln2)*log(x)
 */

#pragma weak __log2l = log2l
#pragma weak log2l = __log2l

#include "libm.h"
#include "longdouble.h"

static const long double
        zero    = 0.0L,
        half    = 0.5L,
        one     = 1.0L,
        invln2  = 1.44269504088896340735992468100189213742 7e+0000L;

long double
log2l(long double x) {
        int n;

        if (x == zero || !finitel(x))
                return (logl(x));
        n = ilogbl(x);
        if (n < 0)
```



```
                n += 1;
        x = scalbnl(x, -n);
        if (x == half)
                return (n - one);
        return (n + invln2 * logl(x));
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2454 Tue Nov  4 19:00:28 2014
new/usr/src/lib/libm/common/Q/logbl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #pragma weak __logbl = logbl
  30 #pragma weak logbl = __logbl

  32 #include "libm.h"
  33 #include "xpg6.h"       /* __xpg6 */
  34 #define _C99SUSv3_logb  _C99SUSv3_logb_subnormal_is_like_ilogb

  36 #if defined(__sparc)
  37 #define ISNORMALL(k, x) (k != 0x7fff)                   /* assuming k != 0 */
  38 #define X86PDNRM(k, x)
  39 #define XSCALE_OFFSET   0x406f                          /* 0x3fff + 112 */
  40 static const long double xscale = 5192296858534827628530496329220096.0L;
  41                                                                 /* 2^112 */
  42 #elif defined(__x86)
  43 /*
  44  * if pseudo-denormal, replace by the equivalent normal
  45  */
  46 #define X86PDNRM(k, x)  if (k == 0 && (((int *) &x)[1] & 0x80000000) != 0) \
  47                             ((int *) &x)[2] |= k = 1
  48 #if defined(HANDLE_UNSUPPORTED)                         /* assuming k != 0 */
  49 #define ISNORMALL(k, x) (k != 0x7fff && (((int *) &x)[1] & 0x80000000) != 0)
  50 #else
  51 #define ISNORMALL(k, x) (k != 0x7fff)
  52 #endif
  53 #define XSCALE_OFFSET   0x403e                          /* 0x3fff + 63 */
  54 static const long double xscale = 9223372036854775808.0L;      /* 2^63 */
  55 #endif

  57 static long double
  58 raise_division(long double v) {
  59 #pragma STDC FENV_ACCESS ON
  60         static const long double zero = 0.0L;
```



```
  61         return (v / zero);
  62 }
______unchanged_portion_omitted_
```

```
**********************************************************
    5512 Tue Nov  4 19:00:29 2014
new/usr/src/lib/libm/common/Q/logl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __logl = logl
#pragma weak logl = __logl

/*
 * logl(x)
 * Table look-up algorithm
 * By K.C. Ng, March 6, 1989
 *
 * (a). For x in [31/33,33/31], using a special approximation:
 *      f = x - 1;
 *      s = f/(2.0+f);  ... here |s| <= 0.03125
 *      z = s*s;
 *      return f-s*(f-z*(B1+z*(B2+z*(B3+z*(B4+...+z*B9)...))));
 *
 * (b). Otherwise, normalize x = 2^n * 1.f.
 *      Use a 6-bit table look-up: find a 6 bit g that match f to 6.5 bits,
 *      then
 *          log(x) = n*ln2 + log(1.g) + log(1.f/1.g).
 *      Here the leading and trailing values of log(1.g) are obtained from
 *      a size-64 table.
 *      For log(1.f/1.g), let s = (1.f-1.g)/(1.f+1.g), then
 *          log(1.f/1.g) = log((1+s)/(1-s)) = 2s + 2/3 s^3 + 2/5 s^5 +...
 *      Note that |s|<2**-8=0.00390625. We use an odd s-polynomial
 *      approximation to compute log(1.f/1.g):
 *              s*(A1+s^2*(A2+s^2*(A3+s^2*(A4+s^2*(A5+s^2*(A6+s^2*A7))))))
 *      (Precision is 2**-136.91 bits, absolute error)
 *
 * (c). The final result is computed by
 *              (n*ln2_hi+_TBL_logl_hi[j]) +
 *                      ( (n*ln2_lo+_TBL_logl_lo[j]) + s*(A1+...) )
 *
 * Note.
```



```
 *      For ln2_hi and _TBL_logl_hi[j], we force their last 32 bit to be zero
 *      so that n*ln2_hi + _TBL_logl_hi[j] is exact. Here
 *      _TBL_logl_hi[j] + _TBL_logl_lo[j] match log(1+j*2**-6) to 194 bits
 *
 *
 * Special cases:
 *      log(x) is NaN with signal if x < 0 (including -INF) ;
 *      log(+INF) is +INF; log(0) is -INF with signal;
 *      log(NaN) is that NaN with no signal.
 *
 * Constants:
 * The hexadecimal values are the intended ones for the following constants.
 * The decimal values may be used, provided that the compiler will convert
 * from decimal to binary accurately enough to produce the hexadecimal values
 * shown.
 */

#include "libm.h"

extern const long double _TBL_logl_hi[], _TBL_logl_lo[];

static const long double
        zero    =   0.0L,
        one     =   1.0L,
        two     =   2.0L,
        two113  =   10384593717069655257060992658440192.0L,
        ln2hi   =   6.93147180559945309417231954749584485020 3e-0001L,
        ln2lo   =   1.66708592083055220889044933040037975416 9e-0025L,
        A1      =   2.00000000000000000000000000000000000000 24e+0000L,
        A2      =   6.66666666666666666666666666666091393804e-0001L,
        A3      =   4.00000000000000000000000040716707022067 1e-0001L,
        A4      =   2.85714285714285714273007749061290368116 4e-0001L,
        A5      =   2.22222222222224257770283692081288260509 9e-0001L,
        A6      =   1.81818181643549339598591266710588582835 6e-0001L,
        A7      =   1.53853783521183975111206751280549693172 5e-0001L,
        B1      =   6.66666666666666666666666666666961498329e-0001L,
        B2      =   3.99999999999999999999999999003765504235 8e-0001L,
        B3      =   2.85714285714285714285727342642834745791 8e-0001L,
        B4      =   2.22222222222222222135322904974791010956 6e-0001L,
        B5      =   1.81818181818182150353255930630907013804 6e-0001L,
        B6      =   1.53846153845380921048635608458735678855 6e-0001L,
        B7      =   1.33333334446335875612145689264517879548 0e-0001L,
        B8      =   1.17646090478389906485464517460336038379 2e-0001L,
        B9      =   1.05729386995659899532636860251805699074 6e-0001L;

long double
logl(long double x) {
        long double f, s, z, qn, h, t;
        int *px = (int *) &x;
        int *pz = (int *) &z;
        int i, j, ix, i0, i1, n;

        /* get long double precision word ordering */
        if (*(int *) &one == 0) {
                i0 = 3;
                i1 = 0;
        } else {
                i0 = 0;
                i1 = 3;
        }

        n = 0;
        ix = px[i0];
        if (ix > 0x3ffee0f8) {  /* if x >  31/33 */
                if (ix < 0x3fff1084) {  /* if x < 33/31 */
                        f = x - one;
```

```
127                         z = f * f;
128                         if (((ix - 0x3fff0000) | px[i1] | px[2] | px[1]) == 0) {
129                                 return (zero);  /* log(1)= +0 */
130                         }
131                         s = f / (two + f);      /* |s|<2**-8 */
132                         z = s * s;
133                         return (f - s * (f - z * (B1 + z * (B2 + z * (B3 +
134                                 z * (B4 + z * (B5 + z * (B6 + z * (B7 +
135                                 z * (B8 + z * B9)))))))))));
136                 }
137                 if (ix >= 0x7fff0000)
138                         return (x + x); /* x is +inf or NaN */
139                 goto LARGE_N;
140         }
141         if (ix >= 0x00010000)
142                 goto LARGE_N;
143         i = ix & 0x7fffffff;
144         if ((i | px[i1] | px[2] | px[1]) == 0) {
145                 px[i0] |= 0x80000000;
146                 return (one / x);       /* log(0.0) = -inf */
147         }
148         if (ix < 0) {
149                 if ((unsigned) ix >= 0xffff0000)
150                         return (x - x); /* x is -inf or NaN */
151                 return (zero / zero);   /* log(x<0) is NaN  */
152         }
153         /* subnormal x */
154         x *= two113;
155         n = -113;
156         ix = px[i0];
157 LARGE_N:
158         n += ((ix + 0x200) >> 16) - 0x3fff;
159         ix = (ix & 0x0000ffff) | 0x3fff0000;    /* scale x to [1,2] */
160         px[i0] = ix;
161         i = ix + 0x200;
162         pz[i0] = i & 0xfffffc00;
163         pz[i1] = pz[1] = pz[2] = 0;
164         s = (x - z) / (x + z);
165         j = (i >> 10) & 0x3f;
166         z = s * s;
167         qn = (long double) n;
168         t = qn * ln2lo + _TBL_logl_lo[j];
169         h = qn * ln2hi + _TBL_logl_hi[j];
170         f = t + s * (A1 + z * (A2 + z * (A3 + z * (A4 + z * (A5 +
171                 z * (A6 + z * A7))))));
172         return (h + f);
173 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2732 Tue Nov  4 19:00:29 2014
new/usr/src/lib/libm/common/Q/nextafterl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __nextafterl = nextafterl
#pragma weak nextafterl = __nextafterl

#include "libm.h"
#include <float.h>              /* LDBL_MAX, LDBL_MIN */

#if defined(__sparc)
#define n0      0
#define n1      1
#define n2      2
#define n3      3
#define X86PDNRM1(x)
#define INC(px) { \
                        if (++px[n3] == 0) \
                                if (++px[n2] == 0) \
                                        if (++px[n1] == 0) \
                                                ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n3] == 0xffffffff) \
                                if (--px[n2] == 0xffffffff) \
                                        if (--px[n1] == 0xffffffff) \
                                                --px[n0]; \
                }
#elif defined(__x86)
#define n0      2
#define n1      1
#define n2      0
#define n3      0
/*
 * if pseudo-denormal, replace by the equivalent normal
 */
```



```
#define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
                                0x80000000) != 0) \
                                ((int *) &x)[2] |= 1
#define INC(px) { \
                        if (++px[n2] == 0) \
                                if ((++px[n1] & ~0x80000000) == 0) \
                                        px[n1] = 0x80000000, ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n2] == 0xffffffff) \
                                if (--px[n1] == 0x7fffffff) \
                                        if ((--px[n0] & 0x7fff) != 0) \
                                                px[n1] |= 0x80000000; \
                }
#endif

long double
nextafterl(long double x, long double y) {
        int *px = (int *) &x;
        int *py = (int *) &y;

        if (x == y)
                return (y);             /* C99 requirement */
        if (x != x || y != y)
                return (x * y);

        if (ISZEROL(x)) {       /* x == 0.0 */
                px[n0] = py[n0] & XSGNMSK;
                px[n1] = px[n2] = 0;
                px[n3] = 1;
        } else {
                X86PDNRM1(x);
                if ((px[n0] & XSGNMSK) == 0) {  /* x > 0.0 */
                        if (x > y)      /* x > y */
                                DEC(px)
                        else
                                INC(px)
                } else {
                        if (x < y)      /* x < y */
                                DEC(px)
                        else
                                INC(px)
                }
        }
#ifndef lint
        {
                volatile long double dummy;
                int k = XBIASED_EXP(x);

                if (k == 0)
                        dummy = LDBL_MIN * copysignl(LDBL_MIN, x);
                else if (k == 0x7fff)
                        dummy = LDBL_MAX * copysignl(LDBL_MAX, x);
        }
#endif
        return (x);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    9153 Tue Nov  4 19:00:30 2014
new/usr/src/lib/libm/common/Q/powl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
  1 /*
  2  * CDDL HEADER START
  3  *
  4  * The contents of this file are subject to the terms of the
  5  * Common Development and Distribution License (the "License").
  6  * You may not use this file except in compliance with the License.
  7  *
  8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9  * or http://www.opensolaris.org/os/licensing.
 10  * See the License for the specific language governing permissions
 11  * and limitations under the License.
 12  *
 13  * When distributing Covered Code, include this CDDL HEADER in each
 14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15  * If applicable, add the following below this CDDL HEADER, with the
 16  * fields enclosed by brackets "[]" replaced with your own identifying
 17  * information: Portions Copyright [yyyy] [name of copyright owner]
 18  *
 19  * CDDL HEADER END
 20  */

 22 /*
 23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 24  */
 25 /*
 26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 27  * Use is subject to license terms.
 28  */

 30 #pragma weak __powl = powl
 30 #pragma weak powl = __powl

 32 #include "libm.h"
 33 #include "xpg6.h"       /* __xpg6 */
 34 #define _C99SUSv3_pow   _C99SUSv3_pow_treats_Inf_as_an_even_int

 36 #if defined(__sparc)
 37 #define i0      0
 38 #define i1      1
 39 #define i2      2
 40 #define i3      3

 42 static const long double zero = 0.0L, one = 1.0L, two = 2.0L;

 44 extern const long double _TBL_logl_hi[], _TBL_logl_lo[];

 46 static const long double
 47         two113 = 10384593717069655257060992658440192.0L,
 48         ln2hi = 6.9314718055994530941723195474958448502O3e-0001L,
 49         ln2lo = 1.6670859208305522088904493304003797541690-0025L,
 50         A2 = 6.6666666666666666666666666666666091393804e-0001L,
 51         A3 = 4.0000000000000000000000004071670702206710-0001L,
 52         A4 = 2.8571428571428571427300774906129036811640-0001L,
 53         A5 = 2.2222222222222425777028369208128826050990-0001L,
 54         A6 = 1.8181818164354933959859126671058858283560-0001L,
 55         A7 = 1.5385378352118397511120675128054969317250-0001L,
 56         B1 = 6.6666666666666666666666666666666667787e-0001L,
 57         B2 = 3.9999999999999999999999999999848524411e-0001L,
 58         B3 = 2.8571428571428571428571428650845810750700-0001L,
 59         B4 = 2.2222222222222222222201078180064380849700-0001L,
 60         B5 = 1.8181818181818181850514421713370364036740-0001L,
```



```
 61         B6 = 1.5384615384615083635407202862920082076730-0001L,
 62         B7 = 1.3333333335067318420331806383293171084280-0001L,
 63         B8 = 1.1764699845874188906343027882839467616700-0001L,
 64         B9 = 1.0537948915614523317229699015648624971320-0001L;

 66 static long double
 67 logl_x(long double x, long double *w) {
 68         long double f, f1, v, s, z, qn, h, t;
 69         int *px = (int *) &x;
 70         int *pz = (int *) &z;
 71         int i, j, ix, n;

 73         n = 0;
 74         ix = px[i0];
 75         if (ix > 0x3ffef03f && ix < 0x3fff0820) {       /* 65/63 > x > 63/65 */
 76                 f = x - one;
 77                 z = f * f;
 78                 if (((ix - 0x3fff0000) | px[i1] | px[i2] | px[i3]) == 0) {
 79                         *w = zero;
 80                         return (zero);  /* log(1)= +0 */
 81                 }
 82                 qn = one / (two + f);
 83                 s = f * qn;     /* |s|<2**-6 */
 84                 v = s * s;
 85                 h = (long double) (2.0 * (double) s);
 86                 f1 = (long double) ((double) f);
 87                 t = ((two * (f - h) - h * f1) - h * (f - f1)) * qn +
 88                         s * (v * (B1 + v * (B2 + v * (B3 + v * (B4 +
 89                         v * (B5 + v * (B6 + v * (B7 + v * (B8 + v * B9)))))))));
 90                 s = (long double) ((double) (h + t));
 91                 *w = t - (s - h);
 92                 return (s);
 93         }
 94         if (ix < 0x00010000) {  /* subnormal x */
 95                 x *= two113;
 96                 n = -113;
 97                 ix = px[i0];
 98         }
 99         /* LARGE_N */
100         n += ((ix + 0x200) >> 16) - 0x3fff;
101         ix = (ix & 0x0000ffff) | 0x3fff0000;    /* scale x to [1,2] */
102         px[i0] = ix;
103         i = ix + 0x200;
104         pz[i0] = i & 0xfffffc00;
105         pz[i1] = pz[i2] = pz[i3] = 0;
106         qn = one / (x + z);
107         f = x - z;
108         s = f * qn;
109         f1 = (long double) ((double) f);
110         h = (long double) (2.0 * (double) s);
111         t = qn * ((two * (f - z * h) - h * f1) - h * (f - f1));
112         j = (i >> 10) & 0x3f;
113         v = s * s;
114         qn = (long double) n;
115         t += qn * ln2lo + _TBL_logl_lo[j];
116         t += s * (v * (A2 + v * (A3 + v * (A4 + v * (A5 + v * (A6 +
117                 v * A7))))));
118         v = qn * ln2hi + _TBL_logl_hi[j];
119         s = h + v;
120         t += (h - (s - v));
121         z = (long double) ((double) (s + t));
122         *w = t - (z - s);
123         return (z);
124 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2216 Tue Nov  4 19:00:30 2014
new/usr/src/lib/libm/common/Q/remainderl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remainderl = remainderl
#pragma weak remainderl = __remainderl

#include "libm.h"
#include "longdouble.h"

/*
 * remainderl(x,p)
 *      returns  x REM p  =  x - [x/p]*p as if in infinite
 *      precise arithmetic, where [x/p] is the (inifinite bit)
 *      integer nearest x/p (in half way case choose the even one).
 * Method :
 *      Based on fmodl() return x-[x/p]chopped*p exactly.
 */

#define HFMAX   5.94865747678615882542879663314003508098 2e+4931L
#define DBMIN   6.72420628622418701252535563464350520519 6e-4932L

static const long double
        zero = 0.0L,
        half = 0.5L,
        hfmax = HFMAX,  /* half of the maximum number */
        dbmin = DBMIN;  /* double of the minimum (normal) number */

long double
remainderl(long double x, long double p) {
        long double hp;
        int sx;

        if (isnanl(p))
                return (x + p);
        if (!finitel(x))
```



```
                return (x - x);
        p = fabsl(p);
        if (p <= hfmax)
                x = fmodl(x, p + p);
        sx = signbitl(x);
        x = fabsl(x);
        if (p < dbmin) {
                if (x + x > p) {
                        if (x == p)
                                x = zero;
                        else
                                x -= p; /* avoid x-x=-0 in RM mode */
                        if (x + x >= p)
                                x -= p;
                }
        } else {
                hp = half * p;
                if (x > hp) {
                        if (x == p)
                                x = zero;
                        else
                                x -= p; /* avoid x-x=-0 in RM mode */
                        if (x >= hp)
                                x -= p;
                }
        }
        return (sx == 0 ? x : -x);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2131 Tue Nov  4 19:00:31 2014
new/usr/src/lib/libm/common/Q/rintl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __rintl = rintl
#pragma weak rintl = __rintl

/*
 * rintl(long double x) return x rounded to integral according to
 * the prevailing rounding direction
 *
 * NOTE: aintl(x), anintl(x), ceill(x), floorl(x), and rintl(x) return result
 * with the same sign as x's, including 0.0L.
 */

#include "libm.h"
#include "longdouble.h"

extern enum fp_precision_type __swapRP(enum fp_precision_type);

static const double one = 1.0;
static const long double qzero = 0.0L;

long double
rintl(long double x) {
        enum fp_precision_type rp;
        long double t, w, two112;
        int *pt = (int *) &two112;

        if (!finitel(x))
                return (x + x);

        if (*(int *) &one != 0) {       /* set two112 = 2^112 */
                pt[0] = 0x406f0000;
                pt[1] = pt[2] = pt[3] = 0;
        } else {
```



```
                pt[3] = 0x406f0000;
                pt[0] = pt[1] = pt[2] = 0;
        }

        if (fabsl(x) >= two112)
                return (x);     /* already an integer */
        t = copysignl(two112, x);
        rp = __swapRP(fp_extended);     /* make sure precision is long double */
        w = x + t;                      /* x+sign(x)*2^112 rounded to integer */
        (void) __swapRP(rp);            /* restore precision mode */
        if (w == t)
                return (copysignl(qzero, x));   /* x rounded to zero */
        else
                return (w - t);
}
```

unchanged_portion_omitted_

****
2826 Tue Nov 4 19:00:31 2014
new/usr/src/lib/libm/common/Q/rndintl.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29 #pragma weak aintl = __aintl
30 #pragma weak anintl = __anintl
31 #pragma weak irintl = __irintl
32 #pragma weak nintl = __nintl

29 /*
30  * aintl(x)     return x chopped to integral value
31  * anintl(x)    return sign(x)*(|x|+0.5) chopped to integral value
32  * irintl(x)    return rint(x) in integer format
33  * nintl(x)     return anint(x) in integer format
34  *
35  * NOTE: aintl(x), anintl(x), ceill(x), floorl(x), and rintl(x) return result
36  * with the same sign as x's,  including 0.0.
37  */

39 #include "libm.h"
40 #include "longdouble.h"

42 extern enum fp_direction_type __swapRD(enum fp_direction_type);

44 static const long double qone = 1.0L, qhalf = 0.5L, qmhalf = -0.5L;

46 long double
47 aintl(long double x) {
48         long double t, w;

50         if (!finitel(x))
51                 return (x + x);
52         w = fabsl(x);
53         t = rintl(w);
54         if (t <= w)
55                 return (copysignl(t, x));       /* NaN or already aint(|x|) */
56         else    /* |t|>|x| case */
```



```
57                 return (copysignl(t - qone, x));        /* |t-1|*sign(x) */
58 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1650 Tue Nov  4 19:00:31 2014
new/usr/src/lib/libm/common/Q/scalbl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbl = scalbl
#pragma weak scalbl = __scalbl

/*
 * scalbl(x,n): return x * 2^n by manipulating exponent.
 */

#include "libm.h"

#define n0      0

long double
scalbl(long double x, long double fn) {
        int *py = (int *) &fn, n;
        long double z;

        if (isnanl(x) || isnanl(fn))
                return (x * fn);

        /*
         * fn is inf or NaN
         */
        if ((py[n0] & 0x7fff0000) == 0x7fff0000) {
                if ((py[n0] & 0x80000000) != 0)
                        return (x / (-fn));
                else
                        return (x * fn);
        }
        if (rintl(fn) != fn)
                return ((fn - fn) / (fn - fn));
        if (fn > 65000.0L)
                z = scalbnl(x, 65000);
```



```
        else if (-fn > 65000.0L)
                z = scalbnl(x, -65000);
        else {
                n = (int) fn;
                z = scalbnl(x, n);
        }
        return (z);
}
```

unchanged_portion_omitted_

```
**********************************************************
    2490 Tue Nov  4 19:00:32 2014
new/usr/src/lib/libm/common/Q/scalbnl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbnl = scalbnl
#pragma weak scalbnl = __scalbnl

#include "libm.h"
#include <float.h>              /* LDBL_MAX, LDBL_MIN */
#include <stdlib.h>             /* abs */

#if defined(__sparc)
#define XSET_EXP(k, x)  ((int *) &x)[0] = (((int *) &x)[0] & ~0x7fff0000) | \
                                (k << 16)
#define ISINFNANL(k, x) (k == 0x7fff)
#define XTWOT_OFFSET    113
static const long double xtwot = 10384593717069655257060992658440192.0L,
                                                        /* 2^113 */
        twomtm1 = 4.81482486096808963263994485646231829 6E-35L;  /* 2^-114 */
#elif defined(__x86)
#define XSET_EXP(k, x)  ((int *) &x)[2] = (((int *) &x)[2] & ~0x7fff) | k
#if defined(HANDLE_UNSUPPORTED)
#define ISINFNANL(k, x) (k == 0x7fff || k != 0 && \
                                (((int *) &x)[1] & 0x80000000) == 0)
#else
#define ISINFNANL(k, x) (k == 0x7fff)
#endif
#define XTWOT_OFFSET    64
static const long double xtwot = 18446744073709551616.0L,       /* 2^64 */
        twomtm1 = 2.710505431213761085018 6E-20L;               /* 2^-65 */
#endif

long double
scalbnl(long double x, int n) {
        int k = XBIASED_EXP(x);
```



```
        if (ISINFNANL(k, x))
                return (x + x);
        if (ISZEROL(x) || n == 0)
                return (x);
        if (k == 0) {
                x *= xtwot;
                k = XBIASED_EXP(x) - XTWOT_OFFSET;
        }
        if ((unsigned) abs(n) >= 131072)        /* cast to unsigned for -2^31 */
                n >>= 1;                /* avoid subsequent integer overflow */
        k += n;
        if (k > 0x7ffe)
                return (LDBL_MAX * copysignl(LDBL_MAX, x));
        if (k <= -XTWOT_OFFSET - 1)
                return (LDBL_MIN * copysignl(LDBL_MIN, x));
        if (k > 0) {
                XSET_EXP(k, x);
                return (x);
        }
        k += XTWOT_OFFSET + 1;
        XSET_EXP(k, x);
        return (x * twomtm1);
}
```

*_____unchanged_portion_omitted_*

****
1070 Tue Nov 4 19:00:32 2014
new/usr/src/lib/libm/common/Q/signgaml.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __signgaml = signgaml
#pragma weak signgaml = __signgaml

#include "libm.h"

int signgaml = 0;
```

****
1226 Tue Nov 4 19:00:33 2014
new/usr/src/lib/libm/common/Q/significandl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __significandl = significandl
#pragma weak significandl = __significandl

#include "libm.h"

long double
significandl(long double x) {
        if (ISZEROL(x) || XBIASED_EXP(x) == 0x7fff)     /* 0/+-Inf/NaN */
                return (x + x);
        else
                return (scalbnl(x, -ilogbl(x)));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    2709 Tue Nov  4 19:00:33 2014
new/usr/src/lib/libm/common/Q/sincosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * sincosl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sincosl     ... sin and cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */

#pragma weak __sincosl = sincosl
```



```
#pragma weak sincosl = __sincosl

#include "libm.h"
#include "longdouble.h"

void
sincosl(long double x, long double *s, long double *c) {
        long double y[2], z = 0.0L;
        int n, ix;

        ix = *(int *) &x;       /* High word of x */

        /* |x| ~< pi/4 */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)
                *s = __k_sincosl(x, z, c);
        else if (ix >= 0x7fff0000)
                *s = *c = x - x;        /* trig(Inf or NaN) is NaN */
        else {                          /* argument reduction needed */
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                case 0:
                        *s = __k_sincosl(y[0], y[1], c);
                        break;
                case 1:
                        *c = -__k_sincosl(y[0], y[1], s);
                        break;
                case 2:
                        *s = -__k_sincosl(y[0], y[1], c);
                        *c = -*c;
                        break;
                case 3:
                        *c = __k_sincosl(y[0], y[1], s);
                        *s = -*s;
                        break;
                }
        }
}
______unchanged_portion_omitted_
```

```
**********************************************************
    5939 Tue Nov  4 19:00:34 2014
new/usr/src/lib/libm/common/Q/sincospil.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sincospil = __sincospil

/*
 * void sincospil(long double x, long double *s, long double *c)
 * *s = sinl(pi*x); *c = cosl(pi*x);
 *
 * Algorithm, 10/17/2002, K.C. Ng
 * ------------------------------
 * Let y = |4x|, z = floor(y), and n = (int)(z mod 8.0) (displayed in binary).
 *      1. If y == z, then x is a multiple of pi/4. Return the following values:
 *              ------------------------------------------------
 *               n  x mod 2    sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ------------------------------------------------
 *               000  0.00       +0 ___       +1 ___       +0
 *               001  0.25       +\/0.5       +\/0.5       +1
 *               010  0.50       +1 ___       +0 ___       +inf
 *               011  0.75       +\/0.5       -\/0.5       -1
 *               100  1.00       -0 ___       -1 ___       +0
 *               101  1.25       -\/0.5       -\/0.5       +1
 *               110  1.50       -1 ___       -0 ___       +inf
 *               111  1.75       -\/0.5       +\/0.5       -1
 *              ------------------------------------------------
 *      2. Otherwise,
 *              ------------------------------------------------
 *                n     t        sin(x*pi)    cos(x*pi)   tan(x*pi)
 *              ------------------------------------------------
 *               000  (y-z)/4     sinpi(t)     cospi(t)    tanpi(t)
 *               001  (z+1-y)/4   cospi(t)     sinpi(t)    1/tanpi(t)
 *               010  (y-z)/4     cospi(t)    -sinpi(t)   -1/tanpi(t)
 *               011  (z+1-y)/4   sinpi(t)    -cospi(t)   -tanpi(t)
 *               100  (y-z)/4    -sinpi(t)    -cospi(t)    tanpi(t)
 *               101  (z+1-y)/4  -cospi(t)    -sinpi(t)    1/tanpi(t)
```



```
 *               110  (y-z)/4    -cospi(t)     sinpi(t)   -1/tanpi(t)
 *               111  (z+1-y)/4  -sinpi(t)     cospi(t)   -tanpi(t)
 *              ------------------------------------------------
 *
 * NOTE. This program compute sinpi/cospi(t<0.25) by __k_sin/cos(pi*t, 0.0).
 * This will return a result with error slightly more than one ulp (but less
 * than 2 ulp). If one wants accurate result,  one may break up pi*t in
 * high (tpi_h) and low (tpi_l) parts and call __k_sin/cos(tip_h, tip_lo)
 * instead.
 */

#include "libm.h"
#include "longdouble.h"

#define I(q, m) ((int *) &(q))[m]
#define U(q, m) ((unsigned *) &(q))[m]
#if defined(__LITTLE_ENDIAN) || defined(__x86)
#define LDBL_MOST_SIGNIF_I(ld)  ((I(ld, 2) << 16) | (0xffff & (I(ld, 1) >> 15)))
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, 0)
#define PREC    64
#define PRECM1  63
#define PRECM2  62
static const long double twoPRECM2 = 9.22337203685477580800000000000000e+18L;
#else
#define LDBL_MOST_SIGNIF_I(ld)  I(ld, 0)
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, sizeof (long double) / sizeof (int) - 1)
#define PREC    113
#define PRECM1  112
#define PRECM2  111
static const long double twoPRECM2 = 5.19229685853482762853049632922009 6e+33L;
#endif

static const long double
zero    = 0.0L,
quater  = 0.25L,
one     = 1.0L,
pi      = 3.14159265358979323846264338327950288419 7e+0000L,
sqrth   = 0.70710678118654752440084436210484903928 4835937688474,
tiny    = 1.0e-100;

void
sincospil(long double x, long double *s, long double *c) {
        long double y, z, t;
        int hx, n, k;
        unsigned lx;

        hx = LDBL_MOST_SIGNIF_I(x);
        lx = LDBL_LEAST_SIGNIF_U(x);
        k = ((hx & 0x7fff0000) >> 16) - 0x3fff;
        if (k >= PRECM2) {              /* |x| >= 2**(Prec-2) */
                if (k >= 16384) {
                        *s = *c = x - x;
                } else {
                        if (k >= PREC) {
                                *s = zero;
                                *c = one;
                        } else if (k == PRECM1) {
                                if ((lx & 1) == 0) {
                                        *s = zero;
                                        *c = one;
                                } else {
                                        *s = -zero;
                                        *c = -one;
                                }
                        } else {        /* k = Prec - 2 */
                                if ((lx & 1) == 0) {
```

```
126                                     *s = zero;
127                                     *c = one;
128                             } else {
129                                     *s = one;
130                                     *c = zero;
131                             }
132                             if ((lx & 2) != 0) {
133                                     *s = -*s;
134                                     *c = -*c;
135                             }
136                     }
137             }
138         } else if (k < -2)      /* |x| < 0.25 */
139                 *s = __k_sincosl(pi * fabsl(x), zero, c);
140         else {
141                 /* y = |4x|, z = floor(y), and n = (int)(z mod 8.0) */
142                 y = 4.0L * fabsl(x);
143                 if (k < PRECM2) {
144                         z = y + twoPRECM2;
145                         n = LDBL_LEAST_SIGNIF_U(z) & 7; /* 3 LSb of z */
146                         t = z - twoPRECM2;
147                         k = 0;
148                         if (t == y)
149                                 k = 1;
150                         else if (t > y) {
151                                 n -= 1;
152                                 t = quater + (y - t) * quater;
153                         } else
154                                 t = (y - t) * quater;
155                 } else {        /* k = Prec-3 */
156                         n = LDBL_LEAST_SIGNIF_U(y) & 7; /* 3 LSb of z */
157                         k = 1;
158                 }
159                 if (k) {        /* x = N/4 */
160                         if ((n & 1) != 0)
161                                 *s = *c = sqrth + tiny;
162                         else
163                                 if ((n & 2) == 0) {
164                                         *s = zero;
165                                         *c = one;
166                                 } else {
167                                         *s = one;
168                                         *c = zero;
169                                 }
170                         if ((n & 4) != 0)
171                                 *s = -*s;
172                         if (((n + 1) & 4) != 0)
173                                 *c = -*c;
174                 } else {
175                         if ((n & 1) != 0)
176                                 t = quater - t;
177                         if (((n + (n & 1)) & 2) == 0)
178                                 *s = __k_sincosl(pi * t, zero, c);
179                         else
180                                 *c = __k_sincosl(pi * t, zero, s);
181                         if ((n & 4) != 0)
182                                 *s = -*s;
183                         if (((n + 2) & 4) != 0)
184                                 *c = -*c;
185                 }
186         }
187         if (hx < 0)
188                 *s = -*s;
189 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2392 Tue Nov  4 19:00:34 2014
new/usr/src/lib/libm/common/Q/sinhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinhl = sinhl
#pragma weak sinhl = __sinhl

#include "libm.h"
#include "longdouble.h"

/*
 * sinhl(X)
 * RETURN THE HYPERBOLIC SINE OF X
 *
 * Method :
 *      1. reduce x to non-negative by sinhl(-x) = - sinhl(x).
 *      2.
 *                                              expm1l(x) + expm1l(x)/(expm1l(x)+1)
 *      0 <= x <= lnovft        : sinhl(x) := ----------------------------------
 *                                                      2
 *
 *     lnovft <= x <  INF       : sinhl(x) := expl(x-MEP1*ln2)*2**ME
 *
 * here
 *      lnovft:         logrithm of the overflow threshold
 *                      = MEP1*ln2 chopped to machine precision.
 *      ME              maximum exponent
 *      MEP1            maximum exponent plus 1
 *
 * Special cases:
 *      sinhl(x) is x if x is +INF, -INF, or NaN.
 *      only sinhl(0)=0 is exact for finite argument.
 *
 */
```



```
#define ME      16383
#define MEP1    16384
#define LNOVFT  1.13565234062941439494919310779707634284 5e+4L
                /* last 32 bits of LN2HI is zero */
#define LN2HI   6.93147180559945309417231954749584485020 3e-0001L
#define LN2LO   1.66708592083055220889044933040037975416 9e-0025L

static const long double
        half    = 0.5L,
        one     = 1.0L,
        ln2hi   = LN2HI,
        ln2lo   = LN2LO,
        lnovftL = LNOVFT;

long double
sinhl(long double x) {
        long double r, t;

        if (!finitel(x))
                return (x + x); /* sinh of NaN or +-INF is itself */
        r = fabsl(x);
        if (r < lnovftL) {
                t = expm1l(r);
                r = copysignl((t + t / (one + t)) * half, x);
        } else {
                r = copysignl(expl((r - MEP1 * ln2hi) - MEP1 * ln2lo), x);
                r = scalbnl(r, ME);
        }
        return (r);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2674 Tue Nov  4 19:00:35 2014
new/usr/src/lib/libm/common/Q/sinl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * sinl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_sinl                ... sin function on [-pi/4,pi/4]
 *      __k_cosl                ... cos function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */
```



```
#pragma weak __sinl = sinl
#pragma weak sinl = __sinl

#include "libm.h"
#include "longdouble.h"

long double
sinl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;

        ix = *(int *) &x;               /* High word of x */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)           /* |x| ~< pi/4 */
                return (__k_sinl(x, z));
        else if (ix >= 0x7fff0000)      /* sin(Inf or NaN) is NaN */
                return (x - x);
        else {                          /* argument reduction needed */
                n = __rem_pio2l(x, y);
                switch (n & 3) {
                        case 0:
                                return (__k_sinl(y[0], y[1]));
                        case 1:
                                return (__k_cosl(y[0], y[1]));
                        case 2:
                                return (-__k_sinl(y[0], y[1]));
                        case 3:
                                return (-__k_cosl(y[0], y[1]));
                }
        }
        /* NOTREACHED */
    return 0.0L;
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    5517 Tue Nov  4 19:00:35 2014
new/usr/src/lib/libm/common/Q/sinpil.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sinpil = __sinpil

/*
 * long double sinpil(long double x),
 * return long double precision sinl(pi*x).
 *
 * Algorithm, 10/17/2002, K.C. Ng
 * ------------------------------
 * Let y = |4x|, z = floor(y), and n = (int)(z mod 8.0) (displayed in binary).
 *      1. If y == z, then x is a multiple of pi/4. Return the following values:
 *             ------------------------------------------------
 *               n  x mod 2    sin(x*pi)    cos(x*pi)   tan(x*pi)
 *             ------------------------------------------------
 *              000  0.00       +0 ___       +1 ___       +0
 *              001  0.25       +\/0.5       +\/0.5       +1
 *              010  0.50       +1 ___       +0 ___       +inf
 *              011  0.75       +\/0.5       -\/0.5       -1
 *              100  1.00       -0 ___       -1 ___       +0
 *              101  1.25       -\/0.5       -\/0.5       +1
 *              110  1.50       -1 ___       -0 ___       +inf
 *              111  1.75       -\/0.5       +\/0.5       -1
 *             ------------------------------------------------
 *      2. Otherwise,
 *             ------------------------------------------------
 *               n     t        sin(x*pi)    cos(x*pi)   tan(x*pi)
 *             ------------------------------------------------
 *              000  (y-z)/4     sinpi(t)     cospi(t)    tanpi(t)
 *              001  (z+1-y)/4   cospi(t)     sinpi(t)    1/tanpi(t)
 *              010  (y-z)/4     cospi(t)    -sinpi(t)   -1/tanpi(t)
 *              011  (z+1-y)/4   sinpi(t)    -cospi(t)   -tanpi(t)
 *              100  (y-z)/4    -sinpi(t)    -cospi(t)    tanpi(t)
 *              101  (z+1-y)/4  -cospi(t)    -sinpi(t)    1/tanpi(t)
```



```
 *              110  (y-z)/4    -cospi(t)     sinpi(t)   -1/tanpi(t)
 *              111  (z+1-y)/4  -sinpi(t)     cospi(t)   -tanpi(t)
 *             ------------------------------------------------
 *
 * NOTE. This program compute sinpi/cospi(t<0.25) by __k_sin/cos(pi*t, 0.0).
 * This will return a result with error slightly more than one ulp (but less
 * than 2 ulp). If one wants accurate result,  one may break up pi*t in
 * high (tpi_h) and low (tpi_l) parts and call __k_sin/cos(tip_h, tip_lo)
 * instead.
 */

#include "libm.h"
#include "longdouble.h"

#define I(q, m) ((int *) &(q))[m]
#define U(q, m) ((unsigned *) &(q))[m]
#if defined(__LITTLE_ENDIAN) || defined(__x86)
#define LDBL_MOST_SIGNIF_I(ld)  ((I(ld, 2) << 16) | (0xffff & (I(ld, 1) >> 15)))
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, 0)
#define PREC    64
#define PRECM1  63
#define PRECM2  62
static const long double twoPRECM2 = 9.22337203685477580800000000000000e+18L;
#else
#define LDBL_MOST_SIGNIF_I(ld)  I(ld, 0)
#define LDBL_LEAST_SIGNIF_U(ld) U(ld, sizeof (long double) / sizeof (int) - 1)
#define PREC    113
#define PRECM1  112
#define PRECM2  111
static const long double twoPRECM2 = 5.19229685853482762853049632922009 6e+33L;
#endif

static const long double
zero    = 0.0L,
quater  = 0.25L,
one     = 1.0L,
pi      = 3.14159265358979323846264338327950288419 7e+0000L,
sqrth   = 0.70710678118654752440084436210484903928 4835937688474,
tiny    = 1.0e-100;

long double
sinpil(long double x) {
        long double y, z, t;
        int hx, n, k;
        unsigned lx;

        hx = LDBL_MOST_SIGNIF_I(x);
        lx = LDBL_LEAST_SIGNIF_U(x);
        k = ((hx & 0x7fff0000) >> 16) - 0x3fff;
        if (k >= PRECM2) {              /* |x| >= 2**(Prec-2) */
                if (k >= 16384)
                        y = x - x;
                else {
                        if (k >= PREC)
                                y = zero;
                        else if (k == PRECM1)
                                y = (lx & 1) == 0 ? zero: -zero;
                        else {  /* k = Prec - 2 */
                                y = (lx & 1) == 0 ? zero : one;
                                if ((lx & 2) != 0)
                                        y = -y;
                        }
                }
        } else if (k < -2)      /* |x| < 0.25 */
                y = __k_sinl(pi * fabsl(x), zero);
        else {
```

```
126                 /* y = |4x|, z = floor(y), and n = (int)(z mod 8.0) */
127                 y = 4.0L * fabsl(x);
128                 if (k < PRECM2) {
129                         z = y + twoPRECM2;
130                         n = LDBL_LEAST_SIGNIF_U(z) & 7; /* 3 LSb of z */
131                         t = z - twoPRECM2;
132                         k = 0;
133                         if (t == y)
134                                 k = 1;
135                         else if (t > y) {
136                                 n -= 1;
137                                 t = quater + (y - t) * quater;
138                         } else
139                                 t = (y - t) * quater;
140                 } else {        /* k = Prec-3 */
141                         n = LDBL_LEAST_SIGNIF_U(y) & 7; /* 3 LSb of z */
142                         k = 1;
143                 }
144                 if (k) {        /* x = N/4 */
145                         if ((n & 1) != 0)
146                                 y = sqrth + tiny;
147                         else
148                                 y = (n & 2) == 0 ? zero : one;
149                         if ((n & 4) != 0)
150                                 y = -y;
151                 } else {
152                         if ((n & 1) != 0)
153                                 t = quater - t;
154                         if (((n + (n & 1)) & 2) == 0)
155                                 y = __k_sinl(pi * t, zero);
156                         else
157                                 y = __k_cosl(pi * t, zero);
158                         if ((n & 4) != 0)
159                                 y = -y;
160                 }
161         }
162         return (hx >= 0 ? y : -y);
163 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    9966 Tue Nov  4 19:00:36 2014
new/usr/src/lib/libm/common/Q/sqrtl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sqrtl = sqrtl
#pragma weak sqrtl = __sqrtl

#include "libm.h"
#include "longdouble.h"

extern int __swapTE(int);
extern int __swapEX(int);
extern enum fp_direction_type __swapRD(enum fp_direction_type);

/*
 * in struct longdouble, msw consists of
 *      unsigned short  sgn:1;
 *      unsigned short  exp:15;
 *      unsigned short  frac1:16;
 */

#ifdef __LITTLE_ENDIAN

/* array indices used to access words within a double */
#define HIWORD  1
#define LOWORD  0

/* structure used to access words within a quad */
union longdouble {
        struct {
                unsigned int    frac4;
                unsigned int    frac3;
                unsigned int    frac2;
                unsigned int    msw;
        } l;
        long double     d;
```



```
};
______unchanged_portion_omitted_
```

```
**********************************************************
    2581 Tue Nov  4 19:00:36 2014
new/usr/src/lib/libm/common/Q/tanhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tanhl = tanhl
#pragma weak tanhl = __tanhl

/*
 * tanhl(x) returns the hyperbolic tangent of x
 *
 * Method :
 *      1. reduce x to non-negative:  tanhl(-x) = - tanhl(x).
 *      2.
 *        0      <  x <=  small    :  tanhl(x) := x
 *                                                -expm1l(-2x)
 *        small  <  x <=  1        :  tanhl(x) := --------------
 *                                                 expm1l(-2x) + 2
 *                                                       2
 *        1      <= x <= threshold :  tanhl(x) := 1 -  ---------------
 *                                                    expm1l(2x) + 2
 *     threshold <  x <= INF       :  tanhl(x) := 1.
 *
 * where
 *      single :        small = 1.e-5           threshold = 11.0
 *      double :        small = 1.e-10          threshold = 22.0
 *      quad   :        small = 1.e-20          threshold = 45.0
 *
 * Note: threshold was chosen so that
 *              fl(1.0+2/(expm1(2*threshold)+2)) == 1.
 *
 * Special cases:
 *      tanhl(NaN) is NaN;
 *      only tanhl(0.0)=0.0 is exact for finite argument.
 */

#include "libm.h"
```



```
#include "longdouble.h"

static const long double small = 1.0e-20L, one = 1.0, two = 2.0,
#ifndef lint
        big = 1.0e+20L,
#endif
        threshold = 45.0L;

long double
tanhl(long double x) {
        long double t, y, z;
        int signx;
        volatile long double dummy;

        if (isnanl(x))
                return (x + x);         /* x is NaN */
        signx = signbitl(x);
        t = fabsl(x);
        z = one;
        if (t <= threshold) {
                if (t > one)
                        z = one - two / (expm1l(t + t) + two);
                else if (t > small) {
                        y = expm1l(-t - t);
                        z = -y / (y + two);
                } else {
#ifndef lint
                        dummy = t + big;
                                                        /* inexact if t != 0 */
#endif
                        return (x);
                }
        } else if (!finitel(t))
                return (copysignl(one, x));
        else
                return (signx ? -z + small * small : z - small * small);
        return (signx ? -z : z);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2433 Tue Nov  4 19:00:37 2014
new/usr/src/lib/libm/common/Q/tanl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * tanl(x)
 * Table look-up algorithm by K.C. Ng, November, 1989.
 *
 * kernel function:
 *      __k_tanl        ... tangent function on [-pi/4,pi/4]
 *      __rem_pio2l     ... argument reduction routine
 *
 * Method.
 *      Let S and C denote the sin and cos respectively on [-PI/4, +PI/4].
 *      1. Assume the argument x is reduced to y1+y2 = x-k*pi/2 in
 *         [-pi/2 , +pi/2], and let n = k mod 4.
 *      2. Let S=S(y1+y2), C=C(y1+y2). Depending on n, we have
 *
 *          n        sin(x)      cos(x)        tan(x)
 *     ----------------------------------------------------------
 *          0          S           C             S/C
 *          1          C          -S            -C/S
 *          2         -S          -C             S/C
 *          3         -C           S            -C/S
 *     ----------------------------------------------------------
 *
 * Special cases:
 *      Let trig be any of sin, cos, or tan.
 *      trig(+-INF)  is NaN, with signals;
 *      trig(NaN)    is that NaN;
 *
 * Accuracy:
 *      computer TRIG(x) returns trig(x) nearly rounded.
 */

#pragma weak __tanl = tanl
```



```
#pragma weak tanl = __tanl

#include "libm.h"
#include "longdouble.h"

long double
tanl(long double x) {
        long double y[2], z = 0.0L;
        int n, ix;

        ix = *(int *) &x;               /* High word of x */
        ix &= 0x7fffffff;
        if (ix <= 0x3ffe9220)           /* |x| ~< pi/4 */
                return (__k_tanl(x, z, 0));
        else if (ix >= 0x7fff0000)      /* trig(Inf or NaN) is NaN */
                return (x - x);
        else {                          /* argument reduction needed */
                n = __rem_pio2l(x, y);
                return (__k_tanl(y[0], y[1], (n & 1)));
        }
}
______unchanged_portion_omitted_
```

****
1257 Tue Nov 4 19:00:37 2014
new/usr/src/lib/libm/common/R/acosf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __acosf = acosf
#pragma weak acosf = __acosf

#include "libm.h"

static const float zero = 0.0f;

float
acosf(float x) {
        int     ix;

        ix = *(int *)&x & ~0x80000000;
        if (ix > 0x3f800000)    /* |x| > 1 or x is nan */
                return ((x * zero) / zero);
        return ((float)acos((double)x));
}
```

_____***unchanged_portion_omitted_***

****
1264 Tue Nov 4 19:00:37 2014
new/usr/src/lib/libm/common/R/acoshf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __acoshf = acoshf
#pragma weak acoshf = __acoshf

#include "libm.h"

static const float zero = 0.0f;

float
acoshf(float x) {
        int     hx;

        hx = *(int *)&x;
        if (hx < 0x3f800000 || hx > 0x7f800000) /* x < 1 or x is nan */
                return ((x * zero) / zero);
        return ((float)acosh((double)x));
}
```

_____***unchanged_portion_omitted_***

****
1257 Tue Nov 4 19:00:38 2014
new/usr/src/lib/libm/common/R/asinf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinf = asinf
#pragma weak asinf = __asinf

#include "libm.h"

static const float zero = 0.0f;

float
asinf(float x) {
        int     ix;

        ix = *(int *)&x & ~0x80000000;
        if (ix > 0x3f800000)    /* |x| > 1 or x is nan */
                return ((x * zero) / zero);
        return ((float)asin((double)x));
}
```

*_____unchanged_portion_omitted_*

****
1187 Tue Nov 4 19:00:38 2014
new/usr/src/lib/libm/common/R/asinhf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __asinhf = asinhf
#pragma weak asinhf = __asinhf

#include "libm.h"

float
asinhf(float x) {
        if (isnanf(x)) {
                return (x * x);         /* + -> * for Cheetah */
        } else {
                return ((float) asinh((double) x));
        }
}
```

_____unchanged_portion_omitted_

****
8324 Tue Nov 4 19:00:39 2014
new/usr/src/lib/libm/common/R/atan2f.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atan2f = atan2f
#pragma weak atan2f = __atan2f

#include "libm.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

/*
 * For i = 0, ..., 192, let x[i] be the double precision number whose
 * high order 32 bits are 0x3f900000 + (i << 16) and whose low order
 * 32 bits are zero.  Then TBL[i] := atan(x[i]) to double precision.
 */

static const double TBL[] = {
        1.56237286204768313e-02,
        1.66000375562312640e-02,
        1.75763148444955872e-02,
        1.85525586258889763e-02,
        1.95287670414137082e-02,
        2.05049382324763683e-02,
        2.14810703409090559e-02,
        2.24571615089905717e-02,
        2.34332098794675855e-02,
        2.44092135955758099e-02,
        2.53851708010611396e-02,
        2.63610796402007873e-02,
        2.73369382578244127e-02,
        2.83127447993351995e-02,
        2.92884974107309737e-02,
        3.02641942386252458e-02,
        3.12398334302682774e-02,
```



```
        3.31909314971115949e-02,
        3.51417768027967800e-02,
        3.70923545503918164e-02,
        3.90426499551669928e-02,
        4.09926482452637811e-02,
        4.29423346623621707e-02,
        4.48916944623464972e-02,
        4.68407129159696539e-02,
        4.87893753095156174e-02,
        5.07376669454602178e-02,
        5.26855731431300420e-02,
        5.46330792393594777e-02,
        5.65801705891457105e-02,
        5.85268325663017702e-02,
        6.04730505641073168e-02,
        6.24188099959573500e-02,
        6.63088949198234884e-02,
        7.01969710718705203e-02,
        7.40829225490337306e-02,
        7.79666338315423008e-02,
        8.18479898030765457e-02,
        8.57268757707448092e-02,
        8.96031774848717461e-02,
        9.34767811585894698e-02,
        9.73475734872236709e-02,
        1.01215441667466668e-01,
        1.05080273416329528e-01,
        1.08941956989865793e-01,
        1.12800381201659389e-01,
        1.16655435441069349e-01,
        1.20507009691224562e-01,
        1.24354994546761438e-01,
        1.32039761614638762e-01,
        1.39708874289163648e-01,
        1.47361481088651630e-01,
        1.54996741923940973e-01,
        1.62613828597948568e-01,
        1.70211925285474408e-01,
        1.77790228992676075e-01,
        1.85347949995694761e-01,
        1.92884312257974672e-01,
        2.00398553825878512e-01,
        2.07889927202262986e-01,
        2.15357699697738048e-01,
        2.22801153759394521e-01,
        2.30219587276843718e-01,
        2.37612313865471242e-01,
        2.44978663126864143e-01,
        2.59629629408257512e-01,
        2.74167451119658789e-01,
        2.88587361894077410e-01,
        3.02884868374971417e-01,
        3.17055753209147029e-01,
        3.31096076704132103e-01,
        3.45002177207105132e-01,
        3.58770670270572245e-01,
        3.72398446676754202e-01,
        3.85882669398073752e-01,
        3.99220769575252543e-01,
        4.12410441597387323e-01,
        4.25449637370042266e-01,
        4.38336559857957830e-01,
        4.51069655988523499e-01,
        4.63647609000806094e-01,
        4.88333951056405535e-01,
        5.12389460310737732e-01,
```

```
 127         5.35811237960463704e-01,
 128         5.58599315343562441e-01,
 129         5.80756353567670414e-01,
 130         6.02287346134964152e-01,
 131         6.23199329934065904e-01,
 132         6.43501108793284371e-01,
 133         6.63202992706093286e-01,
 134         6.82316554874748071e-01,
 135         7.00854407884450192e-01,
 136         7.18829999621624527e-01,
 137         7.36257428981428097e-01,
 138         7.53151280962194414e-01,
 139         7.69526480405658297e-01,
 140         7.85398163397448279e-01,
 141         8.15691923316223422e-01,
 142         8.44153986113171051e-01,
 143         8.70903457075652976e-01,
 144         8.96055384571343927e-01,
 145         9.19719605350416858e-01,
 146         9.42000040379463610e-01,
 147         9.62994330680936206e-01,
 148         9.82793723247329054e-01,
 149         1.00148313569423464e+00,
 150         1.01914134426634972e+00,
 151         1.03584125300880014e+00,
 152         1.05165021254837376e+00,
 153         1.06663036531574362e+00,
 154         1.08083900054116833e+00,
 155         1.09432890732118993e+00,
 156         1.10714871779409041e+00,
 157         1.13095374397916038e+00,
 158         1.15257199721566761e+00,
 159         1.17227388112847630e+00,
 160         1.19028994968253166e+00,
 161         1.20681737028525249e+00,
 162         1.22202532321098967e+00,
 163         1.23605948947808186e+00,
 164         1.24904577239825443e+00,
 165         1.26109338225244039e+00,
 166         1.27229739520871732e+00,
 167         1.28274087974427076e+00,
 168         1.29249666778978534e+00,
 169         1.30162883400919616e+00,
 170         1.31019393504755555e+00,
 171         1.31824205101683711e+00,
 172         1.32581766366803255e+00,
 173         1.33970565959899957e+00,
 174         1.35212738092095464e+00,
 175         1.36330010035969384e+00,
 176         1.37340076694501589e+00,
 177         1.38257482149012589e+00,
 178         1.39094282700241845e+00,
 179         1.39860551227195762e+00,
 180         1.40564764938026987e+00,
 181         1.41214106460849531e+00,
 182         1.41814699839963154e+00,
 183         1.42371797140649403e+00,
 184         1.42889927219073276e+00,
 185         1.43373015248470903e+00,
 186         1.43824479449822262e+00,
 187         1.44247309910910193e+00,
 188         1.44644133224813509e+00,
 189         1.45368758222803240e+00,
 190         1.46013910562100091e+00,
 191         1.46591938806466282e+00,
 192         1.47112767430373470e+00,
```



```
 193         1.47584462045214027e+00,
 194         1.48013643959415142e+00,
 195         1.48405798811891154e+00,
 196         1.48765509490645531e+00,
 197         1.49096634108265924e+00,
 198         1.49402443552511865e+00,
 199         1.49685728913695626e+00,
 200         1.49948886200960629e+00,
 201         1.50193983749385196e+00,
 202         1.50422816301907281e+00,
 203         1.50636948736934317e+00,
 204         1.50837751679893928e+00,
 205         1.51204050407917401e+00,
 206         1.51529782154917969e+00,
 207         1.51821326518395483e+00,
 208         1.52083793107295384e+00,
 209         1.52321322351791322e+00,
 210         1.52537304737331958e+00,
 211         1.52734543140336587e+00,
 212         1.52915374769630819e+00,
 213         1.53081763967160667e+00,
 214         1.53235373677370856e+00,
 215         1.53377621092096650e+00,
 216         1.53509721411557254e+00,
 217         1.53632722579538861e+00,
 218         1.53747533091664934e+00,
 219         1.53854944435964280e+00,
 220         1.53955649336462841e+00,
 221         1.54139303859089161e+00,
 222         1.54302569020147562e+00,
 223         1.54448660954197448e+00,
 224         1.54580153317597646e+00,
 225         1.54699130060982659e+00,
 226         1.54807296595325550e+00,
 227         1.54906061995310385e+00,
 228         1.54996600675867957e+00,
 229         1.55079899282174605e+00,
 230         1.55156792769518947e+00,
 231         1.55227992472688747e+00,
 232         1.55294108165534417e+00,
 233         1.55355665560036682e+00,
 234         1.55413120308095598e+00,
 235         1.55466869295126031e+00,
 236         1.55517259817441977e+00,
 237 };
```

_____***unchanged_portion_omitted_***

****
1473 Tue Nov 4 19:00:39 2014
new/usr/src/lib/libm/common/R/atan2pif.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak atan2pif = __atan2pif

#include "libm.h"

static const double invpi = 0.31830988618379067153776752;

float
atan2pif(float y, float x) {
        int     ix, iy, hx, hy;

        ix = *(int *)&x;
        iy = *(int *)&y;
        hx = ix & ~0x80000000;
        hy = iy & ~0x80000000;
        if (hx > 0x7f800000 || hy > 0x7f800000) /* x or y is nan */
                return (x * y);
        if ((hx | hy) == 0) {
                /* x and y are both zero */
                if (ix == 0)
                        return (y);
                return ((iy == 0)? 1.0f : -1.0f);
        }
        return ((float)(invpi * atan2((double)y, (double)x)));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    5444 Tue Nov  4 19:00:40 2014
new/usr/src/lib/libm/common/R/atanf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanf = atanf
#pragma weak atanf = __atanf

/* INDENT OFF */
/*
 * float atanf(float x);
 * Table look-up algorithm
 * By K.C. Ng, March 9, 1989
 *
 * Algorithm.
 *
 * The algorithm is based on atan(x)=atan(y)+atan((x-y)/(1+x*y)).
 * We use poly1(x) to approximate atan(x) for x in [0,1/8] with
 * error (relative)
 *      |(atan(x)-poly1(x))/x|<= 2^-115.94      long double
 *      |(atan(x)-poly1(x))/x|<= 2^-58.85       double
 *      |(atan(x)-poly1(x))/x|<= 2^-25.53       float
 * and use poly2(x) to approximate atan(x) for x in [0,1/65] with
 * error (absolute)
 *      |atan(x)-poly2(x)|<= 2^-122.15          long double
 *      |atan(x)-poly2(x)|<= 2^-64.79           double
 *      |atan(x)-poly2(x)|<= 2^-35.36           float
 * and use poly3(x) to approximate atan(x) for x in [1/8,7/16] with
 * error (relative, on for single precision)
 *      |(atan(x)-poly1(x))/x|<= 2^-25.53       float
 *
 * Here poly1-3 are odd polynomial with the following form:
 *              x + x^3*(a1+x^2*(a2+...))
 *
 * (0). Purge off Inf and NaN and 0
 * (1). Reduce x to positive by atan(x) = -atan(-x).
 * (2). For x <= 1/8, use
```



```
 *      (2.1) if x < 2^(-prec/2-2), atan(x) =  x  with inexact
 *      (2.2) Otherwise
 *              atan(x) = poly1(x)
 * (3). For x >= 8 then
 *      (3.1) if x >= 2^(prec+2),   atan(x) = atan(inf) - pio2lo
 *      (3.2) if x >= 2^(prec/3+2), atan(x) = atan(inf) - 1/x
 *      (3.3) if x >  65,           atan(x) = atan(inf) - poly2(1/x)
 *      (3.4) Otherwise,            atan(x) = atan(inf) - poly1(1/x)
 *
 * (4). Now x is in (0.125, 8)
 *      Find y that match x to 4.5 bit after binary (easy).
 *      If iy is the high word of y, then
 *              single : j = (iy - 0x3e000000) >> 19
 *              (single is modified to (iy-0x3f000000)>>19)
 *              double : j = (iy - 0x3fc00000) >> 16
 *              quad   : j = (iy - 0x3ffc0000) >> 12
 *
 *      Let s = (x-y)/(1+x*y). Then
 *      atan(x) = atan(y) + poly1(s)
 *              = _TBL_r_atan_hi[j] + (_TBL_r_atan_lo[j] + poly2(s) )
 *
 *      Note. |s| <= 1.5384615385e-02 = 1/65. Maxium occurs at x = 1.03125
 *
 */

#include "libm.h"

extern const float _TBL_r_atan_hi[], _TBL_r_atan_lo[];
static const float
        big     =   1.0e37F,
        one     =   1.0F,
        p1      =  -3.33318595111168824722536849873354467217 2e-0001F,
        p2      =   1.96935289421345540521134198320318063602 1e-0001F,
        q1      =  -3.33292196409564681956341970411013293745 6e-0001F,
        a1      =  -3.33332346522389361406352335150933893459 2e-0001F,
        a2      =   1.99942562593527780549408227480817406240 3e-0001F,
        a3      =  -1.41754709050973778008576984629030178855 9e-0001F,
        a4      =   1.01625081387199198309727373322743268508 4e-0001F,
        a5      =  -5.13702369368835851575309381179175522180 5e-0002F,
        pio2hi  =   1.570796371e+0000F,
        pio2lo  =  -4.371139000e-0008F;
/* INDENT ON */

float
atanf(float xx) {
        float x, y, z, r, p, s;
        volatile double dummy;
        int ix, iy, sign, j;

        x = xx;
        ix = *(int *) &x;
        sign = ix & 0x80000000;
        ix ^= sign;

        /* for |x| < 1/8 */
        if (ix < 0x3e000000) {
                if (ix < 0x38800000) {  /* if |x| < 2**(-prec/2-2) */
                        dummy = big + x;        /* get inexact flag if x != 0 */
#ifdef lint
                        dummy = dummy;
#endif
                        return (x);
                }
                z = x * x;
                if (ix < 0x3c000000) {  /* if |x| < 2**(-prec/4-1) */
                        x = x + (x * z) * p1;
```

```
127                     return (x);
128             } else {
129                     x = x + (x * z) * (p1 + z * p2);
130                     return (x);
131             }
132         }

134         /* for |x| >= 8.0 */
135         if (ix >= 0x41000000) {
136                 *(int *) &x = ix;
137                 if (ix < 0x42820000) {  /* x <  65 */
138                         r = one / x;
139                         z = r * r;
140                         y = r * (one + z * (p1 + z * p2));      /* poly1 */
141                         y -= pio2lo;
142                 } else if (ix < 0x44800000) {   /* x <  2**(prec/3+2) */
143                         r = one / x;
144                         z = r * r;
145                         y = r * (one + z * q1); /* poly2 */
146                         y -= pio2lo;
147                 } else if (ix < 0x4c800000) {   /* x <  2**(prec+2) */
148                         y = one / x - pio2lo;
149                 } else if (ix < 0x7f800000) {   /* x <  inf */
150                         y = -pio2lo;
151                 } else {                        /* x is inf or NaN */
152                         if (ix > 0x7f800000) {
153                                 return (x * x); /* - -> * for Cheetah */
154                         }
155                         y = -pio2lo;
156                 }

158                 if (sign == 0)
159                         x = pio2hi - y;
160                 else
161                         x = y - pio2hi;
162                 return (x);
163         }


166         /* now x is between 1/8 and 8 */
167         if (ix < 0x3f000000) {  /* between 1/8 and 1/2 */
168                 z = x * x;
169                 x = x + (x * z) * (a1 + z * (a2 + z * (a3 + z * (a4 +
170                         z * a5))));
171                 return (x);
172         }
173         *(int *) &x = ix;
174         iy = (ix + 0x00040000) & 0x7ff80000;
175         *(int *) &y = iy;
176         j = (iy - 0x3f000000) >> 19;

178         if (ix == iy)
179                 p = x - y;      /* p=0.0 */
180         else {
181                 if (sign == 0)
182                         s = (x - y) / (one + x * y);
183                 else
184                         s = (y - x) / (one + x * y);
185                 z = s * s;
186                 p = s * (one + z * q1);
187         }
188         if (sign == 0) {
189                 r = p + _TBL_r_atan_lo[j];
190                 x = r + _TBL_r_atan_hi[j];
191         } else {
192                 r = p - _TBL_r_atan_lo[j];
```



```
193                 x = r - _TBL_r_atan_hi[j];
194         }
195         return (x);
196 }
```

_____***unchanged_portion_omitted_***

****
1322 Tue Nov 4 19:00:40 2014
new/usr/src/lib/libm/common/R/atanhf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __atanhf = atanhf
#pragma weak atanhf = __atanhf

#include "libm.h"

static const float zero = 0.0f;

float
atanhf(float x) {
        int     ix;

        ix = *((int *)&x) & ~0x80000000;
        if (ix > 0x3f800000)    /* |x| > 1 or x is nan */
                return ((x * zero) / zero);
        if (ix == 0x3f800000)   /* |x| == 1 */
                return (x / zero);
        return ((float)atanh((double)x));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   20641 Tue Nov  4 19:00:41 2014
new/usr/src/lib/libm/common/R/besself.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __j0f = j0f
#pragma weak __j1f = j1f
#pragma weak __jnf = jnf
#pragma weak __y0f = y0f
#pragma weak __y1f = y1f
#pragma weak __ynf = ynf
#pragma weak j0f = __j0f
#pragma weak j1f = __j1f
#pragma weak jnf = __jnf
#pragma weak y0f = __y0f
#pragma weak y1f = __y1f
#pragma weak ynf = __ynf

#include "libm.h"
#include <float.h>

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const float
        zerof   = 0.0f,
        onef    = 1.0f;

static const double C[] = {
        0.0,
        -0.125,
        0.25,
        0.375,
        0.5,
        1.0,
        2.0,
        8.0,
```



```
        0.564189583547756286948079451560772585844l,     /* 1/sqrt(pi) */
        0.636619772367581343075535053490057448, /* 2/pi */
        1.0e9,
};
______unchanged_portion_omitted_
```

****
1197 Tue Nov 4 19:00:41 2014
new/usr/src/lib/libm/common/R/cbrtf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cbrtf = cbrtf
#pragma weak cbrtf = __cbrtf

#include "libm.h"

float
cbrtf(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) cbrt((double) x));
}
```

*_____unchanged_portion_omitted_*

****
1189 Tue Nov 4 19:00:42 2014
new/usr/src/lib/libm/common/R/copysignf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __copysignf = copysignf
#pragma weak copysignf = __copysignf

#include "libm.h"

float
copysignf(float x, float y) {
        float w;

        *(int *) &w = (*(int *) &x & ~0x80000000) | (*(int *) &y & 0x80000000);
        return (w);
}
```

_____***unchanged_portion_omitted_***

****
3872 Tue Nov 4 19:00:42 2014
new/usr/src/lib/libm/common/R/cosf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cosf = cosf
#pragma weak cosf = __cosf

/*
 * See sincosf.c
 */

#include "libm.h"

extern const int _TBL_ipio2_inf[];
extern int __rem_pio2m(double *, double *, int, int, int, const int *);
#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const double C[] = {
        1.85735322054308378716204874632872525989806770558e-0003,
        -1.95035094218403635082921458859320791358115801259e-0004,
        5.38400550766074785970952495168558701485841707252e+0002,
        -3.31975110777873728964197739157371509422022905947e+0001,
        1.09349482127188401868272000389539985058873853699e-0003,
        -5.03324285989964979398034700054920226866107675091e-0004,
        2.43792880266971107750418061559602239831538067410e-0005,
        9.14499072605666582228127405245558035523741471271e+0002,
        -3.63151270591815439197122504991683846785293207730e+0001,
        0.63661977236758134307553 5,     /* 2^ -1  * 1.45F306DC9C883 */
        0.5,
        1.570796326734125614166,        /* 2^  0  * 1.921FB54400000 */
        6.077100506506192601475e-11,    /* 2^-34  * 1.0B4611A626331 */
};
```

_____***unchanged_portion_omitted_***

****
1354 Tue Nov 4 19:00:42 2014
new/usr/src/lib/libm/common/R/coshf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __coshf = coshf
#pragma weak coshf = __coshf

#include "libm.h"

float
coshf(float x) {
        double  c;
        float   w;
        int     ix;

        ix = *(int *)&x & ~0x80000000;
        if (ix >= 0x7f800000) {
                /* coshf(x) is |x| if x is +-Inf or NaN */
                return (x * x);
        }
        if (ix >= 0x43000000)   /* coshf(x) trivially overflows */
                c = 1.0e100;
        else
                c = cosh((double)x);
        w = (float)c;
        return (w);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1706 Tue Nov  4 19:00:43 2014
new/usr/src/lib/libm/common/R/erff.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __erff = erff
#pragma weak __erfcf = erfcf
#pragma weak erff = __erff
#pragma weak erfcf = __erfcf

#include "libm.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

float
erff(float x) {
        int     ix;

        ix = *(int *)&x & ~0x80000000;
        if (ix > 0x7f800000)    /* x is NaN */
                return (x * x);
        return ((float)erf((double)x));
}
```

*_____unchanged_portion_omitted_*

****
1199 Tue Nov 4 19:00:43 2014
new/usr/src/lib/libm/common/R/exp10f.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak exp10f = __exp10f

#include "libm.h"

extern double exp10(double);

float
exp10f(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) exp10((double) x));
}
```

_____unchanged_portion_omitted_

****
1197 Tue Nov 4 19:00:44 2014
new/usr/src/lib/libm/common/R/exp2f.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __exp2f = exp2f
#pragma weak exp2f = __exp2f

#include "libm.h"

float
exp2f(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) exp2((double) x));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
   16892 Tue Nov  4 19:00:45 2014
new/usr/src/lib/libm/common/R/expf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __expf = expf
#pragma weak expf = __expf

/* INDENT OFF */
/*
 * float expf(float x);
 * Code by K.C. Ng for SUN 5.0 libmopt
 * 11/5/99
 * Method :
 *      1. For |x| >= 2^7, either underflow/overflow.
 *         More precisely:
 *              x > 88.722839355...(0x42B17218) => overflow;
 *              x < -103.97207642..(0xc2CFF1B4) => underflow.
 *      2. For |x| <  2^-6, use polynomail
 *              exp(x) = 1 + x + p1*x^2 + p2*x^3
 *      3. Otherwise, write |x|=(1+r)*2^n, where 0<=r<1.
 *         Let t = 2^n * (1+r) .... x > 0;
 *             t = 2^n * (1-r) .... x < 0. (x= -2**(n+1)+t)
 *         Since -6 <= n <= 6, we may break t into
 *         six 6-bits chunks:
 *                    -5      -11     -17     -23     -29
 *         t=j *2+j *2  +j *2   +j *2   +j *2   +j *2   +j *2
 *            1     2       3       4       5       6
 *
 *         where 0 <= j  < 64 for i = 1,...,6.
 *                    i
 *         Note that since t has only 24 significant bits,
 *         either j  or j  must be 0.
 *                 1     6
 *                                                  7-6i
 *         One may define j  by   (int) ( t * 2     ) mod 64
 *                        i
 *         mathematically. In actual implementation, they can
```



```
 *         be obtained by manipulating the exponent and
 *         mantissa bits as follow:
 *              Let ix = (HEX(x)&0x007fffff)|0x00800000.
 *              If n>=0, let ix=ix<<n, then j =0 and
 *                                           6
 *                  j  = ix>>(30-6i)) mod 64  ...i=1,...,5
 *                   i
 *              Otherwise, let ix=ix<<(j+6), then j = 0 and
 *                                             1
 *                  j  = ix>>(36-6i)) mod 64  ...i=2,...,6
 *                   i
 *
 *      4. Compute exp(t) by table look-up method.
 *         Precompute ET[k] = exp(j*2^(7-6i)), k=j+64*(6-i).
 *         Then
 *         exp(t) = ET[j +320]*ET[j +256]*ET[j +192]*
 *                     1          2          3
 *
 *                  ET[j +128]*ET[j +64]*ET[j ]
 *                     4          5          6
 *
 *                                n+1
 *      5. If x < 0, return exp(-2   )* exp(t). Note that
 *         -6 <= n <= 6. Let k = n - 6, then we can
 *         precompute
 *                          k-5          n+1
 *         EN[k] = exp(-2   ) = exp(-2   ) for k=0,1,...,12.
 *
 *
 * Special cases:
 *      exp(INF) is INF, exp(NaN) is NaN;
 *      exp(-INF) = 0;
 *      for finite argument, only exp(0) = 1 is exact.
 *
 * Accuracy:
 *      All calculations are done in double precision except for
 *      the case |x| < 2^-6.  When |x| < 2^-6, the error is less
 *      than 0.55 ulp.  When |x| >= 2^-6 and the result is normal,
 *      the error is less than 0.51 ulp.  When FDTOS_TRAPS_... is
 *      defined and the result is subnormal, the error can be as
 *      large as 0.75 ulp.
 */
/* INDENT ON */

#include "libm.h"

/*
 * ET[k] = exp(j*2^(7-6i)) , where j = k mod 64, i = k/64
 */
static const double ET[] = {
        1.00000000000000000000e+00, 1.00000000186264514923e+00,
        1.00000000372529029846e+00, 1.00000000558793544769e+00,
        1.00000000745058059692e+00, 1.00000000931322574615e+00,
        1.00000001117587089539e+00, 1.00000001303851604462e+00,
        1.00000001490116119385e+00, 1.00000001676380656512e+00,
        1.00000001862645171435e+00, 1.00000002048909686359e+00,
        1.00000002235174201282e+00, 1.00000002421438716205e+00,
        1.00000002607703253332e+00, 1.00000002793967768255e+00,
        1.00000002980232283178e+00, 1.00000003166496798102e+00,
        1.00000003352761335229e+00, 1.00000003539025850152e+00,
        1.00000003725290365075e+00, 1.00000003911554879998e+00,
        1.00000004097819417126e+00, 1.00000004284083932049e+00,
        1.00000004470348446972e+00, 1.00000004656612984100e+00,
        1.00000004842877499023e+00, 1.00000005029142036150e+00,
        1.00000005215406551073e+00, 1.00000005401671088201e+00,
        1.00000005587935603124e+00, 1.00000005774200140252e+00,
```

```
 127         1.0000000596046465517 5e+00, 1.0000000614672919230 2e+00,
 128         1.0000000633299370722 5e+00, 1.0000000651925824435 3e+00,
 129         1.0000000670552275927 6e+00, 1.0000000689178729640 4e+00,
 130         1.0000000707805181132 7e+00, 1.0000000726431634845 4e+00,
 131         1.0000000745058086337 7e+00, 1.0000000763684540050 5e+00,
 132         1.0000000782310993763 2e+00, 1.0000000800937445255 6e+00,
 133         1.0000000819563898968 3e+00, 1.0000000838190352681 1e+00,
 134         1.0000000856816806393 8e+00, 1.0000000875443257886 1e+00,
 135         1.0000000894069711598 9e+00, 1.0000000912696165311 6e+00,
 136         1.0000000931322619024 4e+00, 1.0000000949949070516 7e+00,
 137         1.0000000968575524229 5e+00, 1.0000000987201977942 2e+00,
 138         1.0000001005828431655 0e+00, 1.0000001024454885367 7e+00,
 139         1.0000001043081336860 0e+00, 1.0000001061707790572 8e+00,
 140         1.0000001080334244285 6e+00, 1.0000001098960697998 3e+00,
 141         1.0000001117587151711 1e+00, 1.0000001136213605423 8e+00,
 142         1.0000001154840059136 6e+00, 1.0000001173466512849 3e+00,
 143         1.0000000000000000000 0e+00, 1.0000001192092966562 1e+00,
 144         1.0000002384186075232 7e+00, 1.0000003576279326011 9e+00,
 145         1.0000004768372718899 6e+00, 1.0000005960466253895 9e+00,
 146         1.0000007152559931000 7e+00, 1.0000008344653750214 1e+00,
 147         1.0000009536747711536 0e+00, 1.0000010728841814966 5e+00,
 148         1.0000011920936060505 5e+00, 1.0000013113030448153 0e+00,
 149         1.0000014305124977909 1e+00, 1.0000015497219649773 8e+00,
 150         1.0000016689314463747 0e+00, 1.0000017881409419828 7e+00,
 151         1.0000019073504518019 0e+00, 1.0000020265597583179 6e+00,
 152         1.0000021457695140725 3e+00, 1.0000022649790656241 2e+00,
 153         1.0000023841886331865 7e+00, 1.0000025033982140598 7e+00,
 154         1.0000026226078091440 3e+00, 1.0000027418174184390 4e+00,
 155         1.0000028610270419449 1e+00, 1.0000029802366796616 3e+00,
 156         1.0000030994463315892 1e+00, 1.0000032186559977276 4e+00,
 157         1.0000033378656780769 2e+00, 1.0000034570753726370 6e+00,
 158         1.0000035762850814080 6e+00, 1.0000036954948043899 1e+00,
 159         1.0000038147045415826 1e+00, 1.0000039339142929861 7e+00,
 160         1.0000040531240586005 9e+00, 1.0000041723338384258 6e+00,
 161         1.0000042915436324619 8e+00, 1.0000044107534407089 6e+00,
 162         1.0000045299632631667 9e+00, 1.0000046491730930954 8e+00,
 163         1.0000047683829507150 2e+00, 1.0000048875928158054 2e+00,
 164         1.0000050068026951066 7e+00, 1.0000051260125886187 8e+00,
 165         1.0000052452224963417 4e+00, 1.0000053644324182755 6e+00,
 166         1.0000054836423544202 3e+00, 1.0000056028523047757 5e+00,
 167         1.0000057220622693421 3e+00, 1.0000058412722481193 7e+00,
 168         1.0000059604822411074 6e+00, 1.0000060796922483064 0e+00,
 169         1.0000061989022697162 0e+00, 1.0000063181123053368 5e+00,
 170         1.0000064373223551683 6e+00, 1.0000065565324192107 3e+00,
 171         1.0000066757424974639 4e+00, 1.0000067949525899280 2e+00,
 172         1.0000069141626966029 4e+00, 1.0000070333728174887 3e+00,
 173         1.0000071525829525853 6e+00, 1.0000072717931018928 5e+00,
 174         1.0000073910032654112 0e+00, 1.0000075102134431404 0e+00,
 175         1.0000000000000000000 0e+00, 1.0000076294236350804 6e+00,
 176         1.0000152589054784879 6e+00, 1.0000228884453302225 1e+00,
 177         1.0000305180437909502 4e+00, 1.0000381477002613372 9e+00,
 178         1.0000457774149413836 5e+00, 1.0000534071878317554 6e+00,
 179         1.0000610370189331188 6e+00, 1.0000686669082454738 3e+00,
 180         1.0000762968557694865 3e+00, 1.0000839268615058230 7e+00,
 181         1.0000915569254544834 6e+00, 1.0000991870476161338 4e+00,
 182         1.0001068172279914403 3e+00, 1.0001144474665804029 5e+00,
 183         1.0001220777633836878 1e+00, 1.0001297081184019610 6e+00,
 184         1.0001373385316352226 9e+00, 1.0001449690030841388 5e+00,
 185         1.0001525995327493756 5e+00, 1.0001602301206309331 1e+00,
 186         1.0001678607667294773 6e+00, 1.0001754914710456745 3e+00,
 187         1.0001831222335795246 2e+00, 1.0001907530543319158 1e+00,
 188         1.0001983839333028480 9e+00, 1.0002060148704929876 1e+00,
 189         1.0002136458659030005 0e+00, 1.0002212769195328867 5e+00,
 190         1.0002289080313835345 5e+00, 1.0002365392014549438 9e+00,
 191         1.0002441704297477809 1e+00, 1.0002518017162627117 5e+00,
 192         1.0002594330609997364 0e+00, 1.0002670644639597430 4e+00,
```



```
 193         1.0002746959251427316 7e+00, 1.0002823274445495904 7e+00,
 194         1.0002899590221803194 4e+00, 1.0002975906580355847 1e+00,
 195         1.0003052223521160524 2e+00, 1.0003128541044217225 7e+00,
 196         1.0003204859149534833 3e+00, 1.0003281177837115567 5e+00,
 197         1.0003357497106961648 8e+00, 1.0003433816959081958 9e+00,
 198         1.0003510137393476497 9e+00, 1.0003586458410154147 5e+00,
 199         1.0003662780009114907 6e+00, 1.0003739102190367660 2e+00,
 200         1.0003815424953914625 7e+00, 1.0003891748299758024 4e+00,
 201         1.0003968072227906738 2e+00, 1.0004044396738362987 5e+00,
 202         1.0004120721831128992 8e+00, 1.0004197047506213635 9e+00,
 203         1.0004273373763619137 1e+00, 1.0004349700603349937 5e+00,
 204         1.0004426028025410477 8e+00, 1.0004502356029802978 6e+00,
 205         1.0004578684616536321 5e+00, 1.0004655013785612722 6e+00,
 206         1.0004731343537036636 3e+00, 1.0004807673870812490 0e+00,
 207         1.0000000000000000000 0e+00, 1.0004884004786944728 9e+00,
 208         1.0009770394924164538 3e+00, 1.0014659171576667517 9e+00,
 209         1.0019550335910027971 7e+00, 1.0024443889090390857 9e+00,
 210         1.0029339832284467348 7e+00, 1.0034238166659545932 2e+00,
 211         1.0039138893383474648 9e+00, 1.0044042013624685516 5e+00,
 212         1.0048947528552161656 4e+00, 1.0053855439335486199 3e+00,
 213         1.0058765747144782221 1e+00, 1.0063678453150763925 1e+00,
 214         1.0068593558524709941 1e+00, 1.0073511064438473942 0e+00,
 215         1.0078430972064480464 2e+00, 1.0083353282575724385 6e+00,
 216         1.0088277997145780329 2e+00, 1.0093205116948789079 6e+00,
 217         1.0098134643159468737 4e+00, 1.0103066576953110278 2e+00,
 218         1.0108000919505575332 4e+00, 1.0112937671993305066 6e+00,
 219         1.0117876835593315743 0e+00, 1.0122818411483189837 7e+00,
 220         1.0127762400841096024 4e+00, 1.0132708804845714109 6e+00,
 221         1.0137657624676528200 8e+00, 1.0142608861513262574 8e+00,
 222         1.0147562516536434706 9e+00, 1.0152518590927093189 4e+00,
 223         1.0157477085866857269 3e+00, 1.0162438002537923509 3e+00,
 224         1.0167401342123065788 3e+00, 1.0172367105805637521 6e+00,
 225         1.0177335294769569440 4e+00, 1.0182305910199367371 4e+00,
 226         1.0187278953280123339 2e+00, 1.0192254425197500022 9e+00,
 227         1.0197232327137748585 8e+00, 1.0202212660287675039 0e+00,
 228         1.0207195425834700852 6e+00, 1.0212180624966806785 6e+00,
 229         1.0217168258872555419 7e+00, 1.0222158328741091093 4e+00,
 230         1.0227150835762137681 7e+00, 1.0232145781126005257 3e+00,
 231         1.0237143166023578988 4e+00, 1.0242142991646328020 7e+00,
 232         1.0247145259186305477 1e+00, 1.0252149698361440167 e+00,
 233         1.0257157124789060276 3e+00, 1.0262166725238852649 2e+00,
 234         1.0267178772379901285 9e+00, 1.0272193267407172534 4e+00,
 235         1.0277210211516216720 2e+00, 1.0282229605903165925 4e+00,
 236         1.0287251451764733989 3e+00, 1.0292275750298227610 1e+00,
 237         1.0297302502701528581 5e+00, 1.0302331710173109335 9e+00,
 238         1.0307363373912026283 1e+00, 1.0312397495117924510 e+00,
 239         1.0000000000000000000 0e+00, 1.0317434074991027603 8e+00,
 240         1.0644944589178595428 8e+00, 1.0982851403078257579 4e+00,
 241         1.1331484530668263222 0e+00, 1.1691184461695043328 4e+00,
 242         1.2062302494209806713 6e+00, 1.2445201077660952293 5e+00,
 243         1.2840254166877413943 8e+00, 1.3247847587288656967 5e+00,
 244         1.3668379411737963113 9e+00, 1.4102260349257107474 6e+00,
 245         1.4549914146182012508 7e+00, 1.5011778000001227972 9e+00,
 246         1.5488302986341331291 0e+00, 1.5979954499506332510 4e+00,
 247         1.6487212707001281941 6e+00, 1.7010573018484007601 4e+00,
 248         1.7550546569602984890 9e+00, 1.8107660721193872266 4e+00,
 249         1.8682459574322223261 3e+00, 1.9275504501675446711 3e+00,
 250         1.9887374695822919168 4e+00, 2.0518667734879767472 5e+00,
 251         2.1170000166126747842 6e+00, 2.1842008108156178991 5e+00,
 252         2.2535347872132085456 1e+00, 2.3250696602771210308 4e+00,
 253         2.3988752939670980879 3e+00, 2.4750237699630250887 1e+00,
 254         2.5535894580629268091 3e+00, 2.6346490888156308685 1e+00,
 255         2.7182818284590455348 8e+00, 2.8045693562372266960 4e+00,
 256         2.8935959441717611362 3e+00, 2.9854485393653581340 e+00,
 257         3.0802168489180310473 3e+00, 3.1779934275388384001 8e+00,
 258         3.2788737679386734669 2e+00, 3.3829563940924689546 8e+00,
```

```
 259         3.490342957461841422l7e+00, 3.601138336272175610l3e+00,
 260         3.715450737941103920l9e+00, 3.833391804758410348l4e+00,
 261         3.955076722920577214l4e+00, 4.080624335026460158l2e+00,
 262         4.210157256143959969l6e+00, 4.343801993561042351l4e+00,
 263         4.481689070338064517l8e+00, 4.623953152782080522l4e+00,
 264         4.770733181967602654l8e+00, 4.922172509432290787l6e+00,
 265         5.078419037180081474l0e+00, 5.239625362128489172l6e+00,
 266         5.405948925141166760l7e+00, 5.577552164791259592l9e+00,
 267         5.754602676005730721l4e+00, 5.937273373745607152l3e+00,
 268         6.125742661881986350l4e+00, 6.320194607432743971l4e+00,
 269         6.520819120330112461l6e+00, 6.727812138894691429l1e+00,
 270         6.941375821197035556l5e+00, 7.161718742493711431l5e+00,
 271         1.000000000000000000l0e+00, 7.389056098930650406l4e+00,
 272         5.459815003314423620l4e+01, 4.034287934927351102l9e+02,
 273         2.980957987041728301l8e+03, 2.202646579480671789l0e+04,
 274         1.627547914190039155l7e+05, 1.202604284164776792l5e+06,
 275         8.886110520507872104l6e+06, 6.565996913733050972l2e+07,
 276         4.851651954097902774l8e+08, 3.584912846131591796l8e+09,
 277         2.648912212984347152l1e+10, 1.957296094288387756l5e+11,
 278         1.446257064291475097l6e+12, 1.068647458152446289l6e+13,
 279         7.896296018268068750l0e+13, 5.834617425274548750l0e+14,
 280         4.311231547115195000l0e+15, 3.185593175711375600l0e+16,
 281         2.353852668370200000l0e+17, 1.739274941520500992l0e+18,
 282         1.285160011435930828l0e+19, 9.496119420602448281l6e+19,
 283         7.016735912097631436l8e+20, 5.184705528587072045l6e+21,
 284         3.831008000716576915l6e+22, 2.830753303274693947l6e+23,
 285         2.091659496012996103l1e+24, 1.545538935590103918l6e+25,
 286         1.142007389815684234l4e+26, 8.438356668741453831l8e+26,
 287         6.235149080811616743l1e+27, 4.607186634331291780l4e+28,
 288         3.404276049931740758l7e+29, 2.515438670919166879l9e+30,
 289         1.858671745284127887l2e+31, 1.373382979540176107l5e+32,
 290         1.014800388113888746l5e+33, 7.498416996990120907l1e+33,
 291         5.540622384393509834l5e+34, 4.093996962127454511l8e+35,
 292         3.025077322011425622l3e+36, 2.235246603734714974l6e+37,
 293         1.651636254994001809l7e+38, 1.220403294317840834l8e+39,
 294         9.017628405034298519l5e+39, 6.663176216410896185l0e+40,
 295         4.923458286012058261l6e+41, 3.637970947608804749l8e+42,
 296         2.688117141816135609l3e+43, 1.986264836137654343l6e+44,
 297         1.467662230155442385l5e+45, 1.084463855290023132l7e+46,
 298         8.013164264000590698l0e+46, 5.920972027664669936l7e+47,
 299         4.375039447261340969l8e+48, 3.232741191084859474l0e+49,
 300         2.388690601424991270l3e+50, 1.765016885691765546l7e+51,
 301         1.304180878393632256l4e+52, 9.636665736032016641l6e+52,
 302         7.120586326889337931l7e+53, 5.261441182666385969l1e+54,
 303 };
```

_____*unchanged_portion_omitted_*

****
1201 Tue Nov 4 19:00:45 2014
new/usr/src/lib/libm/common/R/expm1f.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __expm1f = expm1f
#pragma weak expm1f = __expm1f

#include "libm.h"

float
expm1f(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) expm1((double) x));
}
```

_____***unchanged_portion_omitted_***

****
1113 Tue Nov 4 19:00:46 2014
new/usr/src/lib/libm/common/R/fabsf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fabsf = fabsf
#pragma weak fabsf = __fabsf

#include "libm.h"

float
fabsf(float x) {
        *(int *) &x &= ~0x80000000;
        return (x);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    2643 Tue Nov  4 19:00:46 2014
new/usr/src/lib/libm/common/R/floorf.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ceilf = ceilf
#pragma weak __floorf = floorf
#pragma weak ceilf = __ceilf
#pragma weak floorf = __floorf

/* INDENT OFF */
/*
 * ceilf(x)     return the biggest integral value (in float) below x
 * floorf(x)    return the least integral value (in float) above x
 *
 * NOTE: ceilf(x) and floorf(x) return result
 * with the same sign as x's,  including 0.0F.
 */

#include "libm.h"

static const float xf[] = {
/* ZEROF */     0.0f,
/* ONEF */      1.0f,
/* MONEF */     -1.0f,
/* HUGEF */     1.0e30f,
};
______unchanged_portion_omitted_
```

```
**********************************************************
    3443 Tue Nov  4 19:00:47 2014
new/usr/src/lib/libm/common/R/fmodf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fmodf = fmodf
#pragma weak fmodf = __fmodf

#include "libm.h"

/* INDENT OFF */
static const int
        is = (int)0x80000000,
        im = 0x007fffff,
        ii = 0x7f800000,
        iu = 0x00800000;
/* INDENT ON */

static const float zero = 0.0;

float
fmodf(float x, float y) {
        float   w;
        int     hx, ix, iy, iz, k, ny, nd;

        hx = *(int *)&x;
        ix = hx & 0x7fffffff;
        iy = *(int *)&y & 0x7fffffff;

        /* purge off exception values */
        if (ix >= ii || iy > ii || iy == 0) {
                w = x * y;
                w = w / w;
        } else if (ix <= iy) {
                if (ix < iy)
                        w = x;  /* return x if |x|<|y| */
                else
                        w = zero * x;   /* return sign(x)*0.0 */
```



```
        } else {
                /* INDENT OFF */
                /*
                 * scale x,y to "normal" with
                 *      ny = exponent of y
                 *      nd = exponent of x minus exponent of y
                 */
                /* INDENT ON */
                ny = iy >> 23;
                k = ix >> 23;

                /* special case for subnormal y or x */
                if (ny == 0) {
                        ny = 1;
                        while (iy < iu) {
                                ny -= 1;
                                iy += iy;
                        }
                        nd = k - ny;
                        if (k == 0) {
                                nd += 1;
                                while (ix < iu) {
                                        nd -= 1;
                                        ix += ix;
                                }
                        } else {
                                ix = iu | (ix & im);
                        }
                } else {
                        nd = k - ny;
                        ix = iu | (ix & im);
                        iy = iu | (iy & im);
                }

                /* fix point fmod for normalized ix and iy */
                /* INDENT OFF */
                /*
                 * while (nd--) {
                 *      iz = ix - iy;
                 * if (iz < 0)
                 *      ix = ix + ix;
                 * else if (iz == 0) {
                 *      *(int *) &w = is & hx;
                 *      return w;
                 * }
                 * else
                 *      ix = iz + iz;
                 * }
                 */
                /* INDENT ON */
                /* unroll the above loop 4 times to gain performance */
                k = nd >> 2;
                nd -= k << 2;
                while (k--) {
                        iz = ix - iy;
                        if (iz >= 0)
                                ix = iz + iz;
                        else
                                ix += ix;
                        iz = ix - iy;
                        if (iz >= 0)
                                ix = iz + iz;
                        else
                                ix += ix;
                        iz = ix - iy;
                        if (iz >= 0)
```

```
127                                     ix = iz + iz;
128                             else
129                                     ix += ix;
130                             iz = ix - iy;
131                             if (iz >= 0)
132                                     ix = iz + iz;
133                             else
134                                     ix += ix;
135                             if (iz == 0) {
136                                     *(int *)&w = is & hx;
137                                     return (w);
138                             }
139                     }
140                     while (nd--) {
141                             iz = ix - iy;
142                             if (iz >= 0)
143                                     ix = iz + iz;
144                             else
145                                     ix += ix;
146                     }
147                     /* end of unrolling */

149                     iz = ix - iy;
150                     if (iz >= 0)
151                             ix = iz;

153                     /* convert back to floating value and restore the sign */
154                     if (ix == 0) {
155                             *(int *)&w = is & hx;
156                             return (w);
157                     }
158                     while (ix < iu) {
159                             ix += ix;
160                             ny -= 1;
161                     }
162                     while (ix > (iu + iu)) {
163                             ny += 1;
164                             ix >>= 1;
165                     }
166                     if (ny > 0) {
167                             *(int *)&w = (is & hx) | (ix & im) | (ny << 23);
168                     } else {
169                             /* subnormal output */
170                             k = -ny + 1;
171                             ix >>= k;
172                             *(int *)&w = (is & hx) | ix;
173                     }
174             }
175             return (w);
176 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1095 Tue Nov  4 19:00:47 2014
new/usr/src/lib/libm/common/R/gammaf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __gammaf = gammaf
#pragma weak gammaf = __gammaf

#include "libm.h"

float
gammaf(float x) {
        return (lgammaf(x));
}
```

*_____unchanged_portion_omitted_*

****
1130 Tue Nov 4 19:00:47 2014
new/usr/src/lib/libm/common/R/gammaf_r.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29 #pragma weak __gammaf_r = gammaf_r
29 #pragma weak gammaf_r = __gammaf_r

31 #include "libm.h"

33 float
34 gammaf_r(float x, int *signgamfp) {
35         return (lgammaf_r(x, signgamfp));
36 }
```

*_____unchanged_portion_omitted_*

****
1821 Tue Nov 4 19:00:48 2014
new/usr/src/lib/libm/common/R/hypotf.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */

22 /*
23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
24  */
25 /*
26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
27  * Use is subject to license terms.
28  */

30 #pragma weak __hypotf = hypotf
30 #pragma weak hypotf = __hypotf

32 #include "libm.h"

34 float
35 hypotf(float x, float y) {
36         double dx, dy;
37         float w;
38         int ix, iy;

40         ix = (*(int *) &x) & 0x7fffffff;
41         iy = (*(int *) &y) & 0x7fffffff;
42         if (ix >= 0x7f800000) {
43                 if (ix == 0x7f800000)
44                         *(int *) &w = x == y ? iy : ix; /* w = |x| = inf */
45                 else if (iy == 0x7f800000)
46                         *(int *) &w = x == y ? ix : iy; /* w = |y| = inf */
47                 else
48                         w = fabsf(x) * fabsf(y);        /* + -> * for Cheetah */
49         } else if (iy >= 0x7f800000) {
50                 if (iy == 0x7f800000)
51                         *(int *) &w = x == y ? ix : iy; /* w = |y| = inf */
52                 else
53                         w = fabsf(x) * fabsf(y);        /* + -> * for Cheetah */
54         } else if (ix == 0)
55                 *(int *) &w = iy;       /* w = |y|  */
56         else if (iy == 0)
57                 *(int *) &w = ix;       /* w = |x|  */
58         else {
59                 dx = (double) x;
60                 dy = (double) y;
```



```
61                 w = (float) sqrt(dx * dx + dy * dy);
62         }
63         return (w);
64 }
```

_____***unchanged_portion_omitted_***

****
2102 Tue Nov 4 19:00:48 2014
new/usr/src/lib/libm/common/R/ilogbf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ilogbf = ilogbf
#pragma weak ilogbf = __ilogbf

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */

#if defined(__x86)
static const float two25 = 33554432.0F;
#else
/*
 * v: a non-zero subnormal |x|
 */
static int
ilogbf_subnormal(unsigned v) {
        int r = -126 - 23;

        if (v & 0xffff0000)
                r += 16, v >>= 16;
        if (v & 0xff00)
                r += 8, v >>= 8;
        if (v & 0xf0)
                r += 4, v >>= 4;
        v <<= 1;
        return (r + ((0xffffaa50 >> v) & 0x3));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1154 Tue Nov  4 19:00:49 2014
new/usr/src/lib/libm/common/R/isnanf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #pragma weak __isnanf = isnanf
  31 #pragma weak _isnanf = isnanf
  30 #pragma weak isnanf = __isnanf
  31 #pragma weak _isnanf = __isnanf

  33 #include "libm.h"

  35 int
  36 isnanf(float x) {
  37         return ((*(int *) &x & ~0x80000000) > 0x7f800000);
  38 }
```

*_____unchanged_portion_omitted_*

****
1268 Tue Nov 4 19:00:49 2014
new/usr/src/lib/libm/common/R/lgammaf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgammaf = lgammaf
#pragma weak lgammaf = __lgammaf

#include "libm.h"

extern int signgamf;

float
lgammaf(float x) {
        float    y;

        if (isnanf(x))
                return (x * x);
        y = (float)__k_lgamma((double)x, &signgamf);
        signgam = signgamf;     /* SUSv3 requires the setting of signgam */
        return (y);
}
```

_____unchanged_portion_omitted_

****
1183 Tue Nov 4 19:00:50 2014
new/usr/src/lib/libm/common/R/lgammaf_r.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lgammaf_r = lgammaf_r
#pragma weak lgammaf_r = __lgammaf_r

#include "libm.h"

float
lgammaf_r(float x, int *signgamfp) {
        if (isnanf(x))
                return (x * x);
        return ((float)__k_lgamma((double)x, signgamfp));
}
```

unchanged_portion_omitted_

```
**********************************************************
    1409 Tue Nov  4 19:00:50 2014
new/usr/src/lib/libm/common/R/log10f.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log10f = log10f
#pragma weak log10f = __log10f

#include "libm.h"

static const float zero = 0.0f, mone = -1.0f;

float
log10f(float x) {
        int     hx, ix;
        float   w;

        hx = *(int *)&x;
        ix = hx & ~0x80000000;
        if (ix > 0x7f800000)
                return (x * x);
        if (ix == 0x7f800000)
                return (x + x * x);
        if (ix == 0) {
                w = mone;
                return (w / zero);
        }
        if (hx < 0) {
                w = zero;
                return (w / zero);
        }
        return ((float)log10((double)x));
}
```

_____***unchanged_portion_omitted_***

****
1416 Tue Nov 4 19:00:51 2014
new/usr/src/lib/libm/common/R/log1pf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log1pf = log1pf
#pragma weak log1pf = __log1pf

#include "libm.h"

static const float zero = 0.0f;

float
log1pf(float x) {
        int     ix;

        ix = *(int *)&x;
        if (ix >= 0x7f800000) {
                /* x is +inf or nan */
                return (x * x);
        }
        if (ix < 0) {
                ix &= ~0x80000000;
                if (ix == 0x3f800000) /* x is -1 */
                        return (x / zero);
                if (ix > 0x3f800000) /* x is < -1 or nan */
                        return ((x * zero) / zero);
        }
        return ((float)log1p((double)x));
}
```

_____***unchanged_portion_omitted_***

****
1197 Tue Nov 4 19:00:51 2014
new/usr/src/lib/libm/common/R/log2f.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __log2f = log2f
#pragma weak log2f = __log2f

#include "libm.h"

float
log2f(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) log2((double) x));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    2080 Tue Nov  4 19:00:51 2014
new/usr/src/lib/libm/common/R/logbf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #pragma weak __logbf = logbf
  30 #pragma weak logbf = __logbf

  32 #include "libm.h"
  33 #include "xpg6.h"       /* __xpg6 */
  34 #define _C99SUSv3_logb  _C99SUSv3_logb_subnormal_is_like_ilogb

  36 #if defined(__x86)
  37 static const float two25 = 33554432.0F;
  38 #else
  39 /*
  40  * v: a non-zero subnormal |x|
  41  */
  42 static int
  43 ilogbf_subnormal(unsigned v) {
  44         int r = -126 - 23;

  46         if (v & 0xffff0000)
  47                 r += 16, v >>= 16;
  48         if (v & 0xff00)
  49                 r += 8, v >>= 8;
  50         if (v & 0xf0)
  51                 r += 4, v >>= 4;
  52         v <<= 1;
  53         return (r + ((0xffffaa50 >> v) & 0x3));
  54 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    4679 Tue Nov  4 19:00:52 2014
new/usr/src/lib/libm/common/R/logf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __logf = logf
#pragma weak logf = __logf

/*
 * Algorithm:
 *
 * Let y = x rounded to six significant bits.  Then for any choice
 * of e and z such that y = 2^e z, we have
 *
 * log(x) = e log(2) + log(z) + log(1+(x-y)/y)
 *
 * Note that (x-y)/y = (x'-y')/y' for any scaled x' = sx, y' = sy;
 * in particular, we can take s to be the power of two that makes
 * ulp(x') = 1.
 *
 * From a table, obtain l = log(z) and r = 1/y'.  For |s| <= 2^-6,
 * approximate log(1+s) by a polynomial p(s) where p(s) := s+s*s*
 * (K1+s*(K2+s*K3)).  Then we compute the expression above as
 * e*ln2 + l + p(r*(x'-y')) all evaluated in double precision.
 *
 * When x is subnormal, we first scale it to the normal range,
 * adjusting e accordingly.
 *
 * Accuracy:
 *
 * The largest error is less than 0.6 ulps.
 */

#include "libm.h"

/*
 * For i = 0, ..., 12,
 *   TBL[2i] = log(1 + i/32) and TBL[2i+1] = 2^-23 / (1 + i/32)
```



```
 *
 * For i = 13, ..., 32,
 *   TBL[2i] = log(1/2 + i/64) and TBL[2i+1] = 2^-23 / (1 + i/32)
 */
static const double TBL[] = {
        0.00000000000000000e+00, 1.92092895507812500e-07,
        3.07716586667536873e-02, 1.15596886837121215e-07,
        6.06246218164348399e-02, 1.12196978400735292e-07,
        8.96121586896871380e-02, 1.08991350446428568e-07,
        1.17783035656383456e-01, 1.05963812934027771e-07,
        1.45182009844497889e-01, 1.03099926097972978e-07,
        1.71850256926659228e-01, 1.00386770148026310e-07,
        1.97825743329919867e-01, 9.78127504006410225e-08,
        2.23143551314209764e-01, 9.53674316406250052e-08,
        2.47836163904581269e-01, 9.30413967225609788e-08,
        2.71933715483641758e-01, 9.08261253720238044e-08,
        2.95464212893835898e-01, 8.87138898982558127e-08,
        3.18453731118534588e-01, 8.66976651278409115e-08,
        -3.52220593589352093e-01, 8.47710503472222546e-08,
        -3.30241686870576867e-01, 8.29282014266304324e-08,
        -3.08735481649613285e-01, 8.11637716090425512e-08,
        -2.87682072451780901e-01, 7.94728597005208289e-08,
        -2.67062785249045253e-01, 7.78509646045918305e-08,
        -2.46860077931525784e-01, 7.62939453125000015e-08,
        -2.27057450635346075e-01, 7.47979856004901950e-08,
        -2.07639364778244489e-01, 7.33595628004807733e-08,
        -1.88591169807550029e-01, 7.19754201061320727e-08,
        -1.69899036795397473e-01, 7.06425419560185146e-08,
        -1.51549898127200932e-01, 6.93581321022727239e-08,
        -1.33531392624522626e-01, 6.81195940290178533e-08,
        -1.15831815525121700e-01, 6.69245134320175401e-08,
        -9.84400728132525243e-02, 6.57706425107758611e-08,
        -8.13456394539524008e-02, 6.46558585805084723e-08,
        -6.45385211375711781e-02, 6.35782877604166578e-08,
        -4.80092191863606063e-02, 6.25360207479508229e-08,
        -3.17486983145802981e-02, 6.15273752520161273e-08,
        -1.57483569681391676e-02, 6.05507502480158696e-08,
        0.00000000000000000e+00, 5.96046447753906250e-08,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1937 Tue Nov  4 19:00:52 2014
new/usr/src/lib/libm/common/R/nextafterf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __nextafterf = nextafterf
#pragma weak nextafterf = __nextafterf

#include "libm.h"

float
nextafterf(float x, float y) {
        float w;
        int *pw = (int *) &w;
        int *px = (int *) &x;
        int *py = (int *) &y;
        int ix, iy, iz;

        ix = px[0];
        iy = py[0];
        if ((ix & ~0x80000000) > 0x7f800000)
                return (x * y);         /* + -> * for Cheetah */
        if ((iy & ~0x80000000) > 0x7f800000)
                return (y * x);         /* + -> * for Cheetah */
        if (ix == iy || (ix | iy) == 0x80000000)
                return (y);             /* C99 requirement */
        if ((ix & ~0x80000000) == 0)
                iz = 1 | (iy & 0x80000000);
        else if (ix > 0) {
                if (ix > iy)
                        iz = ix - 1;
                else
                        iz = ix + 1;
        } else {
                if (iy < 0 && ix < iy)
                        iz = ix + 1;
                else
```



```
                        iz = ix - 1;
        }
        pw[0] = iz;
        ix = iz & 0x7f800000;
        if (ix == 0x7f800000) {
                /* raise overflow */
                volatile float t;

                *(int *) &t = 0x7f7fffff;
                t *= t;
        } else if (ix == 0) {
                /* raise underflow */
                volatile float t;

                *(int *) &t = 0x00800000;
                t *= t;
        }
        return (w);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    8104 Tue Nov  4 19:00:53 2014
new/usr/src/lib/libm/common/R/powf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __powf = powf
#pragma weak powf = __powf

#include "libm.h"
#include "xpg6.h"       /* __xpg6 */
#define _C99SUSv3_pow   _C99SUSv3_pow_treats_Inf_as_an_even_int

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

/* INDENT OFF */
static const double
        ln2 = 6.93147180559945286227e-01,       /* 0x3fe62e42, 0xfefa39ef */
        invln2 = 1.44269504088896338700e+00,    /* 0x3ff71547, 0x652b82fe */
        dtwo = 2.0,
        done = 1.0,
        dhalf = 0.5,
        d32 = 32.0,
        d1_32 = 0.03125,
        A0 = 1.99999999981372330364751114695966439250e+0000,
        A1 = 6.66691081793585853377013865713966560861 0e-0001,
        t0 = 2.00000000000477748926240531507320374694 3e+0000,
        t1 = 1.66666340834992637987311193299425072630 7e-0001;

static const double S[] = {
        1.00000000000000000000e+00,     /* 3FF0000000000000 */
        1.02189714865411662714e+00,     /* 3FF059B0D3158574 */
        1.04427378242741375480e+00,     /* 3FF0B5586CF9890F */
        1.06714040067682369717e+00,     /* 3FF11301D0125B51 */
        1.09050773266525768967e+00,     /* 3FF172B83C7D517B */
        1.11438674259589243221e+00,     /* 3FF1D4873168B9AA */
        1.13878863475669156458e+00,     /* 3FF2387A6E756238 */
```



```
        1.16372485877757747552e+00,     /* 3FF29E9DF51FDEE1 */
        1.18920711500272102690e+00,     /* 3FF306FE0A31B715 */
        1.21524735998046895524e+00,     /* 3FF371A7373AA9CB */
        1.24185781207348400201e+00,     /* 3FF3DEA64C123422 */
        1.26905095719173321989e+00,     /* 3FF44E086061892D */
        1.29683955465100964055e+00,     /* 3FF4BFDAD5362A27 */
        1.32523664315974132322e+00,     /* 3FF5342B569D4F82 */
        1.35425554693689265129e+00,     /* 3FF5AB07DD485429 */
        1.38390988196383202258e+00,     /* 3FF6247EB03A5585 */
        1.41421356237309514547e+00,     /* 3FF6A09E667F3BCD */
        1.44518080697704665027e+00,     /* 3FF71F75E8EC5F74 */
        1.47682614593949934623e+00,     /* 3FF7A11473EB0187 */
        1.50916442759342284141e+00,     /* 3FF82589994CCE13 */
        1.54221082540794074411e+00,     /* 3FF8ACE5422AA0DB */
        1.57598084510788649659e+00,     /* 3FF93737B0CDC5E5 */
        1.61049033194925428347e+00,     /* 3FF9C49182A3F090 */
        1.64575547815396494578e+00,     /* 3FFA5503B23E255D */
        1.68179283050742900407e+00,     /* 3FFAE89F995AD3AD */
        1.71861929812247793414e+00,     /* 3FFB7F76F2FB5E47 */
        1.75625216037329945351e+00,     /* 3FFC199BDD85529C */
        1.79470907500310716820e+00,     /* 3FFCB720DCEF9069 */
        1.83400808640934243066e+00,     /* 3FFD5818DCFBA487 */
        1.87416763411029996256e+00,     /* 3FFDFC97337B9B5F */
        1.91520656139714740007e+00,     /* 3FFEA4AFA2A490DA */
        1.95714412417540017941e+00,     /* 3FFF50765B6E4540 */
};
```

_____***unchanged_portion_omitted_***

****
1381 Tue Nov 4 19:00:53 2014
new/usr/src/lib/libm/common/R/remainderf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remainderf = remainderf
#pragma weak remainderf = __remainderf

#include "libm.h"

float
remainderf(float x, float y) {
        if (isnanf(x) || isnanf(y))
                return (x * y);
        if (y == 0.0f || (*(int *) &x & ~0x80000000) == 0x7f800000) {
                /* y is 0 or x is infinite; raise invalid and return NaN */
                y = 0.0f;
                *(int *) &x = 0x7f800000;
                return (x * y);
        }
        return ((float) remainder((double) x, (double) y));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3660 Tue Nov  4 19:00:54 2014
new/usr/src/lib/libm/common/R/rintf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __rintf = rintf
#pragma weak aintf = __aintf
#pragma weak anintf = __anintf
#pragma weak irintf = __irintf
#pragma weak nintf = __nintf
#pragma weak rintf = __rintf

/* INDENT OFF */
/*
 * aintf(x)     return x chopped to integral value
 * anintf(x)    return sign(x)*(|x|+0.5) chopped to integral value
 * irintf(x)    return rint(x) in integer format
 * nintf(x)     return anint(x) in integer format
 * rintf(x)     return x rounded to integral according to the rounding direction
 *
 * NOTE: rintf(x), aintf(x) and anintf(x) return results with the same sign as
 * x's,  including 0.0.
 */

#include "libm.h"

static const float xf[] = {
/* ZEROF */     0.0f,
/* TWO_23F */   8.3886080000e6f,
/* MTWO_23F */  -8.3886080000e6f,
/* ONEF */      1.0f,
/* MONEF */     -1.0f,
/* HALFF */     0.5f,
/* MHALFF */    -0.5f,
/* HUGEF */     1.0e30f,
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1725 Tue Nov  4 19:00:55 2014
new/usr/src/lib/libm/common/R/scalbf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbf = scalbf
#pragma weak scalbf = __scalbf

#include "libm.h"

float
scalbf(float x, float y) {
        int     ix, iy, hx, hy, n;

        ix = *(int *)&x;
        iy = *(int *)&y;
        hx = ix & ~0x80000000;
        hy = iy & ~0x80000000;

        if (hx > 0x7f800000 || hy >= 0x7f800000) {
                /* x is nan or y is inf or nan */
                return ((iy < 0)? x / -y : x * y);
        }

        /* see if y is an integer without raising inexact */
        if (hy >= 0x4b000000) {
                /* |y| >= 2^23, so it must be an integer */
                n = (iy < 0)? -65000 : 65000;
        } else if (hy < 0x3f800000) {
                /* |y| < 1, so it must be zero or non-integer */
                return ((hy == 0)? x : (x - x) / (x - x));
        } else {
                if (hy & ((1 << (0x96 - (hy >> 23))) - 1))
                        return ((y - y) / (y - y));
                n = (int)y;
        }
        return (scalbnf(x, n));
}
```

```
**********************************************************
    2388 Tue Nov  4 19:00:55 2014
new/usr/src/lib/libm/common/R/scalbnf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbnf = scalbnf
#pragma weak scalbnf = __scalbnf

#include "libm.h"
#include <float.h>              /* FLT_MAX, FLT_MIN */
#include <stdlib.h>             /* abs */

static const float twom25f = 2.98023223876953125e-8F;
#if defined(__x86)
static const float two23f = 8388608.0F;
#else
/*
 * v: a non-zero subnormal |x|; returns [-22, 0]
 */
static int
ilogbf_biased(unsigned v) {
        int r = -22;

        if (v & 0xffff0000)
                r += 16, v >>= 16;
        if (v & 0xff00)
                r += 8, v >>= 8;
        if (v & 0xf0)
                r += 4, v >>= 4;
        v <<= 1;
        return (r + ((0xffffaa50 >> v) & 0x3));
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    1070 Tue Nov  4 19:00:55 2014
new/usr/src/lib/libm/common/R/signgamf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __signgamf = signgamf
#pragma weak signgamf = __signgamf

#include "libm.h"

int signgamf = 0;
```

****
1337 Tue Nov 4 19:00:56 2014
new/usr/src/lib/libm/common/R/significandf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __significandf = significandf
#pragma weak significandf = __significandf

#include "libm.h"

float
significandf(float x) {
        int ix = *(int *) &x & ~0x80000000;

        if (ix == 0 || ix >= 0x7f800000)        /* 0/+-Inf/NaN */
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (ix > 0x7f800000 ? x * x : x);
#else
                return (x + x);
#endif
        else
                return (scalbnf(x, -ilogbf(x)));
}
```

_____***unchanged_portion_omitted_***

****
5085 Tue Nov 4 19:00:56 2014
new/usr/src/lib/libm/common/R/sincosf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sincosf = sincosf
#pragma weak sincosf = __sincosf

/* INDENT OFF */
/*
 * For |x| < pi/4, let z = x * x, and approximate sin(x) by
 *
 *      S(x) = x(S0 + S1*z)(S2 + S3*z + z*z)
 * where
 *      S0 =   1.85735322054308378716204874632872525989806770558e-0003,
 *      S1 =  -1.95035094218403635082921458859320791358115801259e-0004,
 *      S2 =   5.38400550766074785970952495168558701485841707252e+0002,
 *      S3 =  -3.31975110777873728964197739157371509422022905947e+0001,
 *
 * with error bounded by |(sin(x) - S(x))/x| < 2**(-28.2), and
 * cos(x) by
 *
 *      C(x) = (C0 + C1*z + C2*z*z) * (C3 + C4*z + z*z)
 * where
 *      C0 =   1.09349482127188401868272000389539985058873853699e-0003
 *      C1 =  -5.03324285989964979398034700054920226866107675091e-0004
 *      C2 =   2.43792880266971107750418061559602239831538067410e-0005
 *      C3 =   9.14499072605666582228127405245558035523741471271e+0002
 *      C4 =  -3.63151270591815439197122504991683846785293207730e+0001
 *
 * with error bounded by |cos(x) - C(x)| < 2**(-34.2).
 */
/* INDENT ON */

#include "libm.h"

extern const int _TBL_ipio2_inf[];
extern int __rem_pio2m(double *, double *, int, int, int, const int *);
```



```
#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const double C[] = {
        1.85735322054308378716204874632872525989806770558e-0003,
        -1.95035094218403635082921458859320791358115801259e-0004,
        5.38400550766074785970952495168558701485841707252e+0002,
        -3.31975110777873728964197739157371509422022905947e+0001,
        1.09349482127188401868272000389539985058873853699e-0003,
        -5.03324285989964979398034700054920226866107675091e-0004,
        2.43792880266971107750418061559602239831538067410e-0005,
        9.14499072605666582228127405245558035523741471271e+0002,
        -3.63151270591815439197122504991683846785293207730e+0001,
        0.636619772367581343075535,     /* 2^ -1  * 1.45F306DC9C883 */
        0.5,
        1.570796326734125614166,        /* 2^  0  * 1.921FB54400000 */
        6.077100506506192601475e-11,    /* 2^-34  * 1.0B4611A626331 */
};
```

______***unchanged_portion_omitted_***

****
1347 Tue Nov 4 19:00:57 2014
new/usr/src/lib/libm/common/R/sincospif.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak sincospif = __sincospif

#include "libm.h"

extern void sincospi(double, double *, double *);

void
sincospif(float x, float *s, float *c) {
        double ds, dc;

#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                *s = *c = x * x;
        else {
#endif
        sincospi((double) x, &ds, &dc);
        *s = (float) ds;
        *c = (float) dc;
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        }
#endif
}
```

_____unchanged_portion_omitted_

****
3911 Tue Nov 4 19:00:57 2014
new/usr/src/lib/libm/common/R/sinf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinf = sinf
#pragma weak sinf = __sinf

/*
 * See sincosf.c
 */

#include "libm.h"

extern const int _TBL_ipio2_inf[];
extern int __rem_pio2m(double *, double *, int, int, int, const int *);
#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const double C[] = {
        1.85735322054308378716204874632872525989806770558e-0003,
        -1.95035094218403635082921458859320791358115801259e-0004,
        5.38400550766074785970952495168558701485841707252e+0002,
        -3.31975110777873728964197739157371509422022905947e+0001,
        1.09349482127188401868272000389539985058873853699e-0003,
        -5.03324285989964979398034700054920226866107675091e-0004,
        2.43792880266971107750418061559602239831538067410e-0005,
        9.14499072605666582228127405245558035523741471271e+0002,
        -3.63151270591815439197122504991683846785293207730e+0001,
        0.63661977236758134307553 5,     /* 2^ -1  * 1.45F306DC9C883 */
        0.5,
        1.570796326734125614166,        /* 2^  0  * 1.921FB54400000 */
        6.077100506506192601475e-11,    /* 2^-34  * 1.0B4611A626331 */
};
______unchanged_portion_omitted_
```

****
1390 Tue Nov 4 19:00:57 2014
new/usr/src/lib/libm/common/R/sinhf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sinhf = sinhf
#pragma weak sinhf = __sinhf

#include "libm.h"

float
sinhf(float x) {
        double  s;
        float   w;
        int     hx, ix;

        hx = *(int *)&x;
        ix = hx & ~0x80000000;
        if (ix >= 0x7f800000) {
                /* sinhf(x) is x if x is +-Inf or NaN */
                return (x * 1.0f);
        }
        if (ix >= 0x43000000)   /* sinhf(x) trivially overflows */
                s = (hx < 0)? -1.0e100 : 1.0e100;
        else
                s = sinh((double)x);
        w = (float)s;
        return (w);
}
```

*_____unchanged_portion_omitted_*

****
1138 Tue Nov 4 19:00:58 2014
new/usr/src/lib/libm/common/R/sqrtf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __sqrtf = sqrtf
#pragma weak sqrtf = __sqrtf

#include "libm.h"


extern float __inline_sqrtf(float);

float
sqrtf(float x) {
        return (__inline_sqrtf(x));
}
```

*_____unchanged_portion_omitted_*

****
4309 Tue Nov 4 19:00:59 2014
new/usr/src/lib/libm/common/R/tanf.c
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29 #pragma weak __tanf = tanf
29 #pragma weak tanf = __tanf

31 #include "libm.h"

33 extern const int _TBL_ipio2_inf[];
34 extern int __rem_pio2m(double *, double *, int, int, int, const int *);
35 #if defined(__i386) && !defined(__amd64)
36 extern int __swapRP(int);
37 #endif

39 static const double C[] = {
40         1.0,
41         4.460669284289592306791405462718103080987930297 85e-0003,
42         4.921653163091890270663952833274379372596740722 66e+0000,
43         -7.114106481614734800444921347661875188350677490 23e-0001,
44         4.085498083740533914465231646317988634109497070 31e+0000,
45         2.504110703980509278210320189828053116798400878 91e+0000,
46         1.114920645602511584115745790768414735794067382 81e+0001,
47         -1.505655409684226508915116937714628875255584716 80e+0000,
48         -1.814843788783492950500431106775067746639251708 98e+0000,
49         3.333359975328356412974096117825108966 41e-0001,
50         2.999997598248363761541668282006867229939e+00,
51         0.6366197723675813430755 35,     /* 2^ -1  * 1.45F306DC9C883 */
52         0.5,
53         1.570796326734125614166,         /* 2^  0  * 1.921FB54400000 */
54         6.077100506506192601475e-11,     /* 2^-34  * 1.0B4611A626331 */
55 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1197 Tue Nov  4 19:00:59 2014
new/usr/src/lib/libm/common/R/tanhf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tanhf = tanhf
#pragma weak tanhf = __tanhf

#include "libm.h"

float
tanhf(float x) {
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
        if (isnanf(x))
                return (x * x);
        else
#endif
        return ((float) tanh((double) x));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    4644 Tue Nov  4 19:01:00 2014
new/usr/src/lib/libm/common/complex/cabs.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cabs = cabs
#pragma weak cabs = __cabs

#include "libm_synonyms.h"
#include <math.h>
#include "complex_wrapper.h"

/*
 * If C were the only standard we cared about, cabs could just call
 * hypot.  Unfortunately, various other standards say that hypot must
 * call matherr and/or set errno to ERANGE when the result overflows.
 * Since cabs should do neither of these things, we have to either
 * make hypot a wrapper on another internal function or duplicate
 * the hypot implementation here.  I've chosen to do the latter.
 */

static const double
        zero = 0.0,
        onep1u = 1.00000000000000022204e+00,    /* 0x3ff00000 1 = 1+2**-52 */
        twom53 = 1.11022302462515654042e-16,    /* 0x3ca00000 0 = 2**-53 */
        twom768 = 6.44114876959713308e-232,    /* 2^-768 */
        two768  = 1.55251809230070893 5e+231;    /* 2^768 */

double
cabs(dcomplex z)
{
        double          x, y, xh, yh, w, ax, ay;
        int             i, j, nx, ny, ix, iy, iscale = 0;
        unsigned        lx, ly;

        x = D_RE(z);
        y = D_IM(z);
```



```
        ix = ((int *)&x)[HIWORD] & ~0x80000000;
        lx = ((int *)&x)[LOWORD];
        iy = ((int *)&y)[HIWORD] & ~0x80000000;
        ly = ((int *)&y)[LOWORD];

        /* force ax = |x| ~>~ ay = |y| */
        if (iy > ix) {
                ax = fabs(y);
                ay = fabs(x);
                i = ix;
                ix = iy;
                iy = i;
                i = lx;
                lx = ly;
                ly = i;
        } else {
                ax = fabs(x);
                ay = fabs(y);
        }
        nx = ix >> 20;
        ny = iy >> 20;
        j  = nx - ny;

        if (nx >= 0x5f3) {
                /* x >= 2^500 (x*x or y*y may overflow) */
                if (nx == 0x7ff) {
                        /* inf or NaN, signal of sNaN */
                        if (((ix - 0x7ff00000) | lx) == 0)
                                return ((ax == ay)? ay : ax);
                        else if (((iy - 0x7ff00000) | ly) == 0)
                                return ((ay == ax)? ax : ay);
                        else
                                return (ax * ay);
                } else if (j > 32) {
                        /* x >> y */
                        if (j <= 53)
                                ay *= twom53;
                        ax += ay;
                        return (ax);
                }
                ax *= twom768;
                ay *= twom768;
                iscale = 2;
                ix -= 768 << 20;
                iy -= 768 << 20;
        } else if (ny < 0x23d) {
                /* y < 2^-450 (x*x or y*y may underflow) */
                if ((ix | lx) == 0)
                        return (ay);
                if ((iy | ly) == 0)
                        return (ax);
                if (j > 53)             /* x >> y */
                        return (ax + ay);
                iscale = 1;
                ax *= two768;
                ay *= two768;
                if (nx == 0) {
                        if (ax == zero) /* guard subnormal flush to zero */
                                return (ax);
                        ix = ((int *)&ax)[HIWORD];
                } else {
                        ix += 768 << 20;
                }
                if (ny == 0) {
                        if (ay == zero) /* guard subnormal flush to zero */
                                return (ax * twom768);
```

```
126                         iy = ((int *)&ay)[HIWORD];
127                 } else {
128                         iy += 768 << 20;
129                 }
130                 j = (ix >> 20) - (iy >> 20);
131                 if (j > 32) {
132                         /* x >> y */
133                         if (j <= 53)
134                                 ay *= twom53;
135                         return ((ax + ay) * twom768);
136                 }
137         } else if (j > 32) {
138                 /* x >> y */
139                 if (j <= 53)
140                         ay *= twom53;
141                 return (ax + ay);
142         }

144         /*
145          * Medium range ax and ay with max{|ax/ay|,|ay/ax|} bounded by 2^32.
146          * First check rounding mode by comparing onep1u*onep1u with onep1u
147          * + twom53.  Make sure the computation is done at run-time.
148          */
149         if (((lx | ly) << 5) == 0) {
150                 ay = ay * ay;
151                 ax += ay / (ax + sqrt(ax * ax + ay));
152         } else if (onep1u * onep1u != onep1u + twom53) {
153                 /* round-to-zero, positive, negative mode */
154                 /* magic formula with less than an ulp error */
155                 w = sqrt(ax * ax + ay * ay);
156                 ax += ay / ((ax + w) / ay);
157         } else {
158                 /* round-to-nearest mode */
159                 w = ax - ay;
160                 if (w > ay) {
161                         ((int *)&xh)[HIWORD] = ix;
162                         ((int *)&xh)[LOWORD] = 0;
163                         ay = ay * ay + (ax - xh) * (ax + xh);
164                         ax = sqrt(xh * xh + ay);
165                 } else {
166                         ax = ax + ax;
167                         ((int *)&xh)[HIWORD] = ix + 0x00100000;
168                         ((int *)&xh)[LOWORD] = 0;
169                         ((int *)&yh)[HIWORD] = iy;
170                         ((int *)&yh)[LOWORD] = 0;
171                         ay = w * w + ((ax - xh) * yh + (ay - yh) * ax);
172                         ax = sqrt(xh * yh + ay);
173                 }
174         }
175         if (iscale > 0) {
176                 if (iscale == 1)
177                         ax *= twom768;
178                 else
179                         ax *= two768;   /* must generate side effect here */
180         }
181         return (ax);
182 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1139 Tue Nov  4 19:01:00 2014
new/usr/src/lib/libm/common/complex/cabsf.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cabsf = cabsf
#pragma weak cabsf = __cabsf

#include "libm.h"
#include "complex_wrapper.h"

float
cabsf(fcomplex z) {
        return (hypotf(F_RE(z), F_IM(z)));
}
______unchanged_portion_omitted_
```

```
**********************************************************
    1148 Tue Nov  4 19:01:00 2014
new/usr/src/lib/libm/common/complex/cabsl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cabsl = cabsl
#pragma weak cabsl = __cabsl

#include "libm.h"
#include "complex_wrapper.h"

long double
cabsl(ldcomplex z) {
        return (hypotl(LD_RE(z), LD_IM(z)));
}
```

_____*unchanged_portion_omitted_*

****
12776 Tue Nov 4 19:01:01 2014
new/usr/src/lib/libm/common/complex/cacos.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacos = cacos
#pragma weak cacos = __cacos

/* INDENT OFF */
/*
 * dcomplex cacos(dcomplex z);
 *
 * Alogrithm
 * (based on T.E.Hull, Thomas F. Fairgrieve and Ping Tak Peter Tang's
 * paper "Implementing the Complex Arcsine and Arccosine Functins Using
 * Exception Handling", ACM TOMS, Vol 23, pp 299-335)
 *
 * The principal value of complex inverse cosine function cacos(z),
 * where z = x+iy, can be defined by
 *
 *      cacos(z) = acos(B) - i sign(y) log (A + sqrt(A*A-1)),
 *
 * where the log function is the natural log, and
 *                ____________          ____________
 *       1    /      2    2       1    /      2    2
 *  A = ---  / (x+1)  + y   +  ---  / (x-1)  + y
 *       2 \/                   2 \/
 *                ____________          ____________
 *       1    /      2    2       1    /      2    2
 *  B = ---  / (x+1)  + y   -  ---  / (x-1)  + y   .
 *       2 \/                   2 \/
 *
 * The Branch cuts are on the real line from -inf to -1 and from 1 to inf.
 * The real and imaginary parts are based on Abramowitz and Stegun
 * [Handbook of Mathematic Functions, 1972].  The sign of the imaginary
 * part is chosen to be the generally considered the principal value of
 * this function.
```



```
 *
 * Notes:1. A is the average of the distances from z to the points (1,0)
 *          and (-1,0) in the complex z-plane, and in particular A>=1.
 *       2. B is in [-1,1], and A*B = x
 *
 * Basic relations
 *    cacos(conj(z)) = conj(cacos(z))
 *    cacos(-z)      = pi   - cacos(z)
 *    cacos( z)      = pi/2 - casin(z)
 *
 * Special cases (conform to ISO/IEC 9899:1999(E)):
 *    cacos(+-0  + i y  ) = pi/2 - i y for y is +-0, +-inf, NaN
 *    cacos( x   + i inf) = pi/2 - i inf for all x
 *    cacos( x   + i NaN) = NaN  + i NaN with invalid for non-zero finite x
 *    cacos(-inf + i y  ) = pi   - i inf for finite +y
 *    cacos( inf + i y  ) = 0    - i inf for finite +y
 *    cacos(-inf + i inf) = 3pi/4- i inf
 *    cacos( inf + i inf) = pi/4 - i inf
 *    cacos(+-inf+ i NaN) = NaN  - i inf (sign of imaginary is unspecified)
 *    cacos(NaN  + i y  ) = NaN  + i NaN with invalid for finite y
 *    cacos(NaN  + i inf) = NaN  - i inf
 *    cacos(NaN  + i NaN) = NaN  + i NaN
 *
 * Special Regions (better formula for accuracy and for avoiding spurious
 * overflow or underflow) (all x and y are assumed nonnegative):
 *  case 1: y = 0
 *  case 2: tiny y relative to x-1: y <= ulp(0.5)*|x-1|
 *  case 3: tiny y: y < 4 sqrt(u), where u = minimum normal number
 *  case 4: huge y relative to x+1: y >= (1+x)/ulp(0.5)
 *  case 5: huge x and y: x and y >= sqrt(M)/8, where M = maximum normal number
 *  case 6: tiny x: x < 4 sqrt(u)
 *  --------
 *  case        1 & 2. y=0 or y/|x-1| is tiny. We have
 *                 ____________              _____________
 *            /      2    2              /       y    2
 *           / (x+-1)  + y   =  |x+-1| / 1 + (------)
 *         \/                         \/       |x+-1|
 *
 *                                             1     y    2
 *                           ~  |x+-1| ( 1 + --- (------)  )
 *                                             2   |x+-1|
 *
 *                                           2
 *                                          y
 *                           = |x+-1| + --------.
 *                                      2|x+-1|
 *
 *      Consequently, it is not difficult to see that
 *                                  2
 *                                 y
 *                    [ 1 + ------------ ,     if x < 1,
 *                    [      2(1+x)(1-x)
 *                    [
 *                    [
 *                    [ x,                     if x = 1 (y = 0),
 *                    [
 *              A ~=  [             2
 *                    [        x * y
 *                    [ x + ------------ ~ x,  if x > 1
 *                    [      2(x+1)(x-1)
 *
 *      and hence
 *                      ______                                 2
 *                     / 2                    y               y
 *               A + \/ A  - 1  ~  1 + ---------------- + -----------, if x < 1,
 *                                     sqrt((x+1)(1-x))   2(x+1)(1-x)
```

```
 127  *
 128  *
 129  *                             ~  x + sqrt((x-1)*(x+1)),            if x >= 1.
 130  *
 131  *                                      2
 132  *                                     y
 133  *                           [ x(1 - -----------) ~ x,  if x < 1,
 134  *                           [       2(1+x)(1-x)
 135  *              B = x/A  ~   [
 136  *                           [ 1,                       if x = 1,
 137  *                           [
 138  *                           [          2
 139  *                           [         y
 140  *                           [ 1 - ------------ ,       if x > 1,
 141  *                           [      2(x+1)(x-1)
 142  *      Thus
 143  *                             [ acos(x) - i y/sqrt((x-1)*(x+1)),      if x < 1,
 144  *                             [
 145  *              cacos(x+i*y)~  [ 0 - i 0,                              if x = 1,
 146  *                             [
 147  *                             [ y/sqrt(x*x-1) - i log(x+sqrt(x*x-1)), if x > 1.
 148  *
 149  *      Note: y/sqrt(x*x-1) ~ y/x when x >= 2**26.
 150  *  case 3. y < 4 sqrt(u), where u = minimum normal x.
 151  *      After case 1 and 2, this will only occurs when x=1. When x=1, we have
 152  *         A = (sqrt(4+y*y)+y)/2 ~ 1 + y/2 + y^2/8 + ...
 153  *      and
 154  *         B = 1/A = 1 - y/2 + y^2/8 + ...
 155  *      Since
 156  *         cos(sqrt(y)) ~ 1 - y/2 + ...
 157  *      we have, for the real part,
 158  *         acos(B) ~ acos(1 - y/2) ~ sqrt(y)
 159  *      For the imaginary part,
 160  *         log(A+sqrt(A*A-1)) ~ log(1+y/2+sqrt(2*y/2))
 161  *                            = log(1+y/2+sqrt(y))
 162  *                            = (y/2+sqrt(y)) - (y/2+sqrt(y))^2/2 + ...
 163  *                            ~ sqrt(y) - y*(sqrt(y)+y/2)/2
 164  *                            ~ sqrt(y)
 165  *
 166  *  case 4. y >= (x+1)/ulp(0.5). In this case, A ~ y and B ~ x/y. Thus
 167  *         real part = acos(B) ~ pi/2
 168  *      and
 169  *         imag part = log(y+sqrt(y*y-one))
 170  *
 171  *  case 5. Both x and y are large: x and y > sqrt(M)/8, where M = maximum x
 172  *      In this case,
 173  *         A ~ sqrt(x*x+y*y)
 174  *         B ~ x/sqrt(x*x+y*y).
 175  *      Thus
 176  *         real part = acos(B) = atan(y/x),
 177  *         imag part = log(A+sqrt(A*A-1)) ~ log(2A)
 178  *                   = log(2) + 0.5*log(x*x+y*y)
 179  *                   = log(2) + log(y) + 0.5*log(1+(x/y)^2)
 180  *
 181  *  case 6. x < 4 sqrt(u). In this case, we have
 182  *          A ~ sqrt(1+y*y), B = x/sqrt(1+y*y).
 183  *      Since B is tiny, we have
 184  *          real part = acos(B) ~ pi/2
 185  *          imag part = log(A+sqrt(A*A-1)) = log (A+sqrt(y*y))
 186  *                    = log(y+sqrt(1+y*y))
 187  *                    = 0.5*log(y^2+2ysqrt(1+y^2)+1+y^2)
 188  *                    = 0.5*log(1+2y(y+sqrt(1+y^2)));
 189  *                    = 0.5*log1p(2y(y+A));
 190  *
 191  *      cacos(z) = acos(B) - i sign(y) log (A + sqrt(A*A-1)),
 192  */
```



```
 193 /* INDENT ON */

 195 #include "libm.h"
 196 #include "complex_wrapper.h"

 198 /* INDENT OFF */
 199 static const double
 200         zero = 0.0,
 201         one = 1.0,
 202         E = 1.1102230246251565404236e-16,                /* 2**-53 */
 203         ln2 = 6.93147180559945286227e-01,
 204         pi = 3.14159265358979311599796346854,
 205         pi_l = 1.2246467991473531772e-16,
 206         pi_2 = 1.57079632679489656e+00,
 207         pi_2_l = 6.123233995736765886e-17,
 208         pi_4 = 0.78539816339744827899949,
 209         pi_4_l = 3.061616997868382943e-17,
 210         pi3_4 = 2.35619449019234836998,
 211         pi3_4_l = 9.1848509936051488291956e-17,
 212         Foursqrtu = 5.9666725849601653946e-154,        /* 2**(-509) */
 213         Acrossover = 1.5,
 214         Bcrossover = 0.6417,
 215         half = 0.5;
 216 /* INDENT ON */

 218 dcomplex
 219 cacos(dcomplex z) {
 220         double x, y, t, R, S, A, Am1, B, y2, xm1, xp1, Apx;
 221         int ix, iy, hx, hy;
 222         unsigned lx, ly;
 223         dcomplex ans;

 225         x = D_RE(z);
 226         y = D_IM(z);
 227         hx = HI_WORD(x);
 228         lx = LO_WORD(x);
 229         hy = HI_WORD(y);
 230         ly = LO_WORD(y);
 231         ix = hx & 0x7fffffff;
 232         iy = hy & 0x7fffffff;

 234         /* x is 0 */
 235         if ((ix | lx) == 0) {
 236                 if (((iy | ly) == 0) || (iy >= 0x7ff00000)) {
 237                         D_RE(ans) = pi_2;
 238                         D_IM(ans) = -y;
 239                         return (ans);
 240                 }
 241         }

 243         /* |y| is inf or NaN */
 244         if (iy >= 0x7ff00000) {
 245                 if (ISINF(iy, ly)) {    /* cacos(x + i inf) = pi/2  - i inf */
 246                         D_IM(ans) = -y;
 247                         if (ix < 0x7ff00000) {
 248                                 D_RE(ans) = pi_2 + pi_2_l;
 249                         } else if (ISINF(ix, lx)) {
 250                                 if (hx >= 0)
 251                                         D_RE(ans) = pi_4 + pi_4_l;
 252                                 else
 253                                         D_RE(ans) = pi3_4 + pi3_4_l;
 254                         } else {
 255                                 D_RE(ans) = x;
 256                         }
 257                 } else {                /* cacos(x + i NaN) = NaN  + i NaN */
 258                         D_RE(ans) = y + x;
```

```
259                         if (ISINF(ix, lx))
260                                 D_IM(ans) = -fabs(x);
261                         else
262                                 D_IM(ans) = y;
263                 }
264                 return (ans);
265         }

267         x = fabs(x);
268         y = fabs(y);

270         /* x is inf or NaN */
271         if (ix >= 0x7ff00000) { /* x is inf or NaN */
272                 if (ISINF(ix, lx)) {    /* x is INF */
273                         D_IM(ans) = -x;
274                         if (iy >= 0x7ff00000) {
275                                 if (ISINF(iy, ly)) {
276                                         /* INDENT OFF */
277                                         /* cacos(inf + i inf) = pi/4 - i inf */
278                                         /* cacos(-inf+ i inf) =3pi/4 - i inf */
279                                         /* INDENT ON */
280                                         if (hx >= 0)
281                                                 D_RE(ans) = pi_4 + pi_4_l;
282                                         else
283                                                 D_RE(ans) = pi3_4 + pi3_4_l;
284                                 } else
285                                         /* INDENT OFF */
286                                         /* cacos(inf + i NaN) = NaN  - i inf  */
287                                         /* INDENT ON */
288                                         D_RE(ans) = y + y;
289                         } else
290                                 /* INDENT OFF */
291                                 /* cacos(inf + iy ) = 0  - i inf */
292                                 /* cacos(-inf+ iy  ) = pi - i inf */
293                                 /* INDENT ON */
294                         if (hx >= 0)
295                                 D_RE(ans) = zero;
296                         else
297                                 D_RE(ans) = pi + pi_l;
298                 } else {                        /* x is NaN */
299                         /* INDENT OFF */
300                         /*
301                          * cacos(NaN + i inf) = NaN  - i inf
302                          * cacos(NaN + i y  ) = NaN  + i NaN
303                          * cacos(NaN + i NaN) = NaN  + i NaN
304                          */
305                         /* INDENT ON */
306                         D_RE(ans) = x + y;
307                         if (iy >= 0x7ff00000) {
308                                 D_IM(ans) = -y;
309                         } else {
310                                 D_IM(ans) = x;
311                         }
312                 }
313                 if (hy < 0)
314                         D_IM(ans) = -D_IM(ans);
315                 return (ans);
316         }

318         if ((iy | ly) == 0) {   /* region 1: y=0 */
319                 if (ix < 0x3ff00000) {  /* |x| < 1 */
320                         D_RE(ans) = acos(x);
321                         D_IM(ans) = zero;
322                 } else {
323                         D_RE(ans) = zero;
324                         if (ix >= 0x43500000)   /* |x| >= 2**54 */
```



```
325                                 D_IM(ans) = ln2 + log(x);
326                         else if (ix >= 0x3ff80000)      /* x > Acrossover */
327                                 D_IM(ans) = log(x + sqrt((x - one) * (x +
328                                         one)));
329                         else {
330                                 xm1 = x - one;
331                                 D_IM(ans) = log1p(xm1 + sqrt(xm1 * (x + one)));
332                         }
333                 }
334         } else if (y <= E * fabs(x - one)) {    /* region 2: y < tiny*|x-1| */
335                 if (ix < 0x3ff00000) {  /* x < 1 */
336                         D_RE(ans) = acos(x);
337                         D_IM(ans) = y / sqrt((one + x) * (one - x));
338                 } else if (ix >= 0x43500000) {  /* |x| >= 2**54 */
339                         D_RE(ans) = y / x;
340                         D_IM(ans) = ln2 + log(x);
341                 } else {
342                         t = sqrt((x - one) * (x + one));
343                         D_RE(ans) = y / t;
344                         if (ix >= 0x3ff80000)   /* x > Acrossover */
345                                 D_IM(ans) = log(x + t);
346                         else
347                                 D_IM(ans) = log1p((x - one) + t);
348                 }
349         } else if (y < Foursqrtu) {     /* region 3 */
350                 t = sqrt(y);
351                 D_RE(ans) = t;
352                 D_IM(ans) = t;
353         } else if (E * y - one >= x) {  /* region 4 */
354                 D_RE(ans) = pi_2;
355                 D_IM(ans) = ln2 + log(y);
356         } else if (ix >= 0x5fc00000 || iy >= 0x5fc00000) {      /* x,y>2**509 */
357                 /* region 5: x+1 or y is very large (>= sqrt(max)/8) */
358                 t = x / y;
359                 D_RE(ans) = atan(y / x);
360                 D_IM(ans) = ln2 + log(y) + half * log1p(t * t);
361         } else if (x < Foursqrtu) {
362                 /* region 6: x is very small, < 4sqrt(min) */
363                 D_RE(ans) = pi_2;
364                 A = sqrt(one + y * y);
365                 if (iy >= 0x3ff80000)   /* if y > Acrossover */
366                         D_IM(ans) = log(y + A);
367                 else
368                         D_IM(ans) = half * log1p((y + y) * (y + A));
369         } else {        /* safe region */
370                 y2 = y * y;
371                 xp1 = x + one;
372                 xm1 = x - one;
373                 R = sqrt(xp1 * xp1 + y2);
374                 S = sqrt(xm1 * xm1 + y2);
375                 A = half * (R + S);
376                 B = x / A;
377                 if (B <= Bcrossover)
378                         D_RE(ans) = acos(B);
379                 else {          /* use atan and an accurate approx to a-x */
380                         Apx = A + x;
381                         if (x <= one)
382                                 D_RE(ans) = atan(sqrt(half * Apx * (y2 / (R +
383                                         xp1) + (S - xm1))) / x);
384                         else
385                                 D_RE(ans) = atan((y * sqrt(half * (Apx / (R +
386                                         xp1) + Apx / (S + xm1)))) / x);
387                 }
388                 if (A <= Acrossover) {
389                         /* use log1p and an accurate approx to A-1 */
390                         if (x < one)
```

```
391                                 Am1 = half * (y2 / (R + xp1) + y2 / (S - xm1));
392                         else
393                                 Am1 = half * (y2 / (R + xp1) + (S + xm1));
394                         D_IM(ans) = log1p(Am1 + sqrt(Am1 * (A + one)));
395                 } else {
396                         D_IM(ans) = log(A + sqrt(A * A - one));
397                 }
398         }
399         if (hx < 0)
400                 D_RE(ans) = pi - D_RE(ans);
401         if (hy >= 0)
402                 D_IM(ans) = -D_IM(ans);
403         return (ans);
404 }
```

*unchanged_portion_omitted_*

****
1313 Tue Nov 4 19:01:01 2014
new/usr/src/lib/libm/common/complex/cacosf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacosf = cacosf
#pragma weak cacosf = __cacosf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
cacosf(fcomplex z) {
        dcomplex dz, dans;
        fcomplex ans;

        D_RE(dz) = (double) (F_RE(z));
        D_IM(dz) = (double) (F_IM(z));
        dans = cacos(dz);
        F_RE(ans) = (float) (D_RE(dans));
        F_IM(ans) = (float) (D_IM(dans));
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1735 Tue Nov  4 19:01:02 2014
new/usr/src/lib/libm/common/complex/cacosh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacosh = cacosh
#pragma weak cacosh = __cacosh

/* INDENT OFF */
/*
 * dcomplex cacosh(dcomplex z);
 *      cacosh z = +-i cacos z .
 * In order to make conj(cacosh(z))=cacosh(conj(z)),
 * we define
 *      cacosh z = sign(Im(z))*i cacos z .
 *
 */
/* INDENT ON */

#include "libm.h"       /* fabs/isnan/isinf/signbit */
#include "complex_wrapper.h"

/* need to work on special cases according to spec */

dcomplex
cacosh(dcomplex z) {
        dcomplex w, ans;
        double x, y;

        w = cacos(z);
        x = D_RE(z);
        y = D_IM(z);
        if (isnan(y)) {
                D_IM(ans) = y + y;
                if (isinf(x))
                        D_RE(ans) = fabs(x);
                else
```



```
                        D_RE(ans) = y;
        } else if (signbit(y) == 0) {
                D_RE(ans) = -D_IM(w);
                D_IM(ans) = D_RE(w);
        } else {
                D_RE(ans) = D_IM(w);
                D_IM(ans) = -D_RE(w);
        }
        return (ans);
}
```

_____***unchanged_portion_omitted_***

****
1372 Tue Nov 4 19:01:02 2014
new/usr/src/lib/libm/common/complex/cacoshf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacoshf = cacoshf
#pragma weak cacoshf = __cacoshf

#include "libm.h"
#include "complex_wrapper.h"

/* need to work on special cases according to spec */

fcomplex
cacoshf(fcomplex z) {
        dcomplex dz, dans;
        fcomplex ans;

        D_RE(dz) = (double) (F_RE(z));
        D_IM(dz) = (double) (F_IM(z));
        dans = cacosh(dz);
        F_RE(ans) = (float) (D_RE(dans));
        F_IM(ans) = (float) (D_IM(dans));
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1740 Tue Nov  4 19:01:03 2014
new/usr/src/lib/libm/common/complex/cacoshl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacoshl = cacoshl
#pragma weak cacoshl = __cacoshl

#include "libm.h"       /* fabsl/isnanl/isinfl/signbitl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
/*
 * ldcomplex cacoshl(ldcomplex z);
 *      cacosh z = +-i cacos z .
 * In order to make conj(cacosh(z))=cacosh(conj(z)),
 * we define
 *      cacosh z = sign(Im(z))*i cacos z .
 *
 */
/* INDENT ON */

ldcomplex
cacoshl(ldcomplex z) {
        ldcomplex w, ans;
        long double x, y;

        w = cacosl(z);
        x = LD_RE(z);
        y = LD_IM(z);
        if (isnanl(y)) {
                LD_IM(ans) = y + y;
                if (isinfl(x))
                        LD_RE(ans) = fabsl(x);
                else
                        LD_RE(ans) = y;
```



```
        } else if (signbitl(y) == 0) {
                LD_RE(ans) = -LD_IM(w);
                LD_IM(ans) = LD_RE(w);
        } else {
                LD_RE(ans) = LD_IM(w);
                LD_IM(ans) = -LD_RE(w);
        }
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7682 Tue Nov  4 19:01:03 2014
new/usr/src/lib/libm/common/complex/cacosl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cacosl = cacosl
#pragma weak cacosl = __cacosl

#include "libm.h"              /* acosl/atanl/fabsl/isinfl/log1pl/logl/sqrtl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double
zero = 0.0L,
one = 1.0L,
Acrossover = 1.5L,
Bcrossover = 0.6417L,
half = 0.5L,
ln2 = 6.93147180559945309417232121458176568075 5e-0001L,
Foursqrtu = 7.33441547021938866248564956819393266382 55e-2466L,  /* 2**-8189 */
#if defined(__x86)
E = 5.42101086242752217003726400434970855712 89e-20L,            /* 2**-64 */
pi = 3.14159265358979323829596852490908531763125211000442504882 8125L,
pi_l = 1.66674858370417566565917289370680772146819592307 8e-19L,
pi_2 = 1.57079632679489661914798426245454265881562 60L,
pi_2_l = 8.33374291852087832829586446853403860736073409796e-20L,
pi_4 = 0.78539816339744830957399213122727132940781302750110626220703125L,
pi_4_l = 4.16687145926043916414793223426701930367048980769 5410e-20L,
pi3_4 = 2.35619449019234492872197639368181398822343908250331878662109375L,
pi3_4_l = 1.25006143777813174924437967028010579110114694230 8e-19L;
#else
E = 9.62964972193617926527988971292463659269 05e-35L,            /* 2**-113 */
pi = 3.14159265358979323846264338327950279747906809813729557 30045043318L,
pi_l = 8.67181013012378102479704402604335196876232334625653034177593568 62e-35L,
pi_2 = 1.57079632679489661923132169163975139873953 40L,
pi_2_l = 4.33590506506189051239852201302167595843811616e-35L,
```



```
pi_4 = 0.78539816339744830961566084581987569936976702453432389325 1126L,
pi_4_l = 2.16795253253094525619926100651083799219058083656413258544 3e-35L,
pi3_4 = 2.35619449019234492884698253745962709810930107360297167975337824L,
pi3_4_l = 6.50385759759283576859778301953251397657174250969239775633 1e-35L;
#endif
/* INDENT ON */

#if defined(__x86)
static const int ip1 = 0x40400000;      /* 2**65 */
#else
static const int ip1 = 0x40710000;      /* 2**114 */
#endif

ldcomplex
cacosl(ldcomplex z) {
        long double x, y, t, R, S, A, Am1, B, y2, xm1, xp1, Apx;
        int ix, iy, hx, hy;
        ldcomplex ans;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        hy = HI_XWORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;

        /* x is 0 */
        if (x == zero) {
                if (y == zero || (iy >= 0x7fff0000)) {
                        LD_RE(ans) = pi_2 + pi_2_l;
                        LD_IM(ans) = -y;
                        return (ans);
                }
        }

        /* |y| is inf or NaN */
        if (iy >= 0x7fff0000) {
                if (isinfl(y)) {        /* cacos(x + i inf) =  pi/2 - i inf */
                        LD_IM(ans) = -y;
                        if (ix < 0x7fff0000) {
                                LD_RE(ans) = pi_2 + pi_2_l;
                        } else if (isinfl(x)) {
                                if (hx >= 0)
                                        LD_RE(ans) = pi_4 + pi_4_l;
                                else
                                        LD_RE(ans) = pi3_4 + pi3_4_l;
                        } else {
                                LD_RE(ans) = x;
                        }
                } else {                /* cacos(x + i NaN) = NaN  + i NaN */
                        LD_RE(ans) = y + x;
                        if (isinfl(x))
                                LD_IM(ans) = -fabsl(x);
                        else
                                LD_IM(ans) = y;
                }
                return (ans);
        }

        y = fabsl(y);

        if (ix >= 0x7fff0000) { /* x is inf or NaN */
                if (isinfl(x)) {        /* x is INF */
                        LD_IM(ans) = -fabsl(x);
                        if (iy >= 0x7fff0000) {
                                if (isinfl(y)) {
```

```
127                                     /* INDENT OFF */
128                                     /* cacos(inf + i inf) = pi/4 - i inf */
129                                     /* cacos(-inf+ i inf) =3pi/4 - i inf */
130                                     /* INDENT ON */
131                                     if (hx >= 0)
132                                             LD_RE(ans) = pi_4 + pi_4_l;
133                                     else
134                                             LD_RE(ans) = pi3_4 + pi3_4_l;
135                                 } else
136                                     /* INDENT OFF */
137                                     /* cacos(inf + i NaN) = NaN  - i inf  */
138                                     /* INDENT ON */
139                                     LD_RE(ans) = y + y;
140                             } else {
141                                     /* INDENT OFF */
142                                     /* cacos(inf + iy ) = 0  - i inf */
143                                     /* cacos(-inf+ iy  ) = pi - i inf */
144                                     /* INDENT ON */
145                                     if (hx >= 0)
146                                             LD_RE(ans) = zero;
147                                     else
148                                             LD_RE(ans) = pi + pi_l;
149                             }
150                     } else {                /* x is NaN */
151                             /* INDENT OFF */
152                             /*
153                              * cacos(NaN + i inf) = NaN  - i inf
154                              * cacos(NaN + i y  ) = NaN  + i NaN
155                              * cacos(NaN + i NaN) = NaN  + i NaN
156                              */
157                             /* INDENT ON */
158                             LD_RE(ans) = x + y;
159                             if (iy >= 0x7fff0000) {
160                                     LD_IM(ans) = -y;
161                             } else {
162                                     LD_IM(ans) = x;
163                             }
164                     }
165                     if (hy < 0)
166                             LD_IM(ans) = -LD_IM(ans);
167                     return (ans);
168             }

170             if (y == zero) {        /* region 1: y=0 */
171                     if (ix < 0x3fff0000) {  /* |x| < 1 */
172                             LD_RE(ans) = acosl(x);
173                             LD_IM(ans) = zero;
174                     } else {
175                             LD_RE(ans) = zero;
176                             x = fabsl(x);
177                             if (ix >= ip1)  /* i386 ? 2**65 : 2**114 */
178                                     LD_IM(ans) = ln2 + logl(x);
179                             else if (ix >= 0x3fff8000)      /* x > Acrossover */
180                                     LD_IM(ans) = logl(x + sqrtl((x - one) * (x +
181                                         one)));
182                             else {
183                                     xm1 = x - one;
184                                     LD_IM(ans) = log1pl(xm1 + sqrtl(xm1 * (x +
185                                         one)));
186                             }
187                     }
188             } else if (y <= E * fabsl(fabsl(x) - one)) {
189                     /* region 2: y < tiny*||x|-1| */
190                     if (ix < 0x3fff0000) {  /* x < 1 */
191                             LD_RE(ans) = acosl(x);
192                             x = fabsl(x);
```



```
193                             LD_IM(ans) = y / sqrtl((one + x) * (one - x));
194                     } else if (ix >= ip1) { /* i386 ? 2**65 : 2**114 */
195                             if (hx >= 0)
196                                     LD_RE(ans) = y / x;
197                             else {
198                                     if (ix >= ip1 + 0x00040000)
199                                             LD_RE(ans) = pi + pi_l;
200                                     else {
201                                             t = pi_l + y / x;
202                                             LD_RE(ans) = pi + t;
203                                     }
204                             }
205                             LD_IM(ans) = ln2 + logl(fabsl(x));
206                     } else {
207                             x = fabsl(x);
208                             t = sqrtl((x - one) * (x + one));
209                             LD_RE(ans) = (hx >= 0)? y / t : pi - (y / t - pi_l);
210                             if (ix >= 0x3fff8000)   /* x > Acrossover */
211                                     LD_IM(ans) = logl(x + t);
212                             else
213                                     LD_IM(ans) = log1pl(t - (one - x));
214                     }
215             } else if (y < Foursqrtu) {     /* region 3 */
216                     t = sqrtl(y);
217                     LD_RE(ans) = (hx >= 0)? t : pi + pi_l;
218                     LD_IM(ans) = t;
219             } else if (E * y - one >= fabsl(x)) {   /* region 4 */
220                     LD_RE(ans) = pi_2 + pi_2_l;
221                     LD_IM(ans) = ln2 + logl(y);
222             } else if (ix >= 0x5ffb0000 || iy >= 0x5ffb0000) {
223                     /* region 5: x+1 and y are both (>= sqrt(max)/8) i.e. 2**8188 */
224                     t = x / y;
225                     LD_RE(ans) = atan2l(y, x);
226                     LD_IM(ans) = ln2 + logl(y) + half * log1pl(t * t);
227             } else if (fabsl(x) < Foursqrtu) {
228                     /* region 6: x is very small, < 4sqrt(min) */
229                     LD_RE(ans) = pi_2 + pi_2_l;
230                     A = sqrtl(one + y * y);
231                     if (iy >= 0x3fff8000)   /* if y > Acrossover */
232                             LD_IM(ans) = logl(y + A);
233                     else
234                             LD_IM(ans) = half * log1pl((y + y) * (y + A));
235             } else {                /* safe region */
236                     t = fabsl(x);
237                     y2 = y * y;
238                     xp1 = t + one;
239                     xm1 = t - one;
240                     R = sqrtl(xp1 * xp1 + y2);
241                     S = sqrtl(xm1 * xm1 + y2);
242                     A = half * (R + S);
243                     B = t / A;

245                     if (B <= Bcrossover)
246                             LD_RE(ans) = (hx >= 0)? acosl(B) : acosl(-B);
247                     else {          /* use atan and an accurate approx to a-x */
248                             Apx = A + t;
249                             if (t <= one)
250                                     LD_RE(ans) = atan2l(sqrtl(half * Apx * (y2 /
251                                             (R + xp1) + (S - xm1))), x);
252                             else
253                                     LD_RE(ans) = atan2l((y * sqrtl(half * (Apx /
254                                             (R + xp1) + Apx / (S + xm1)))), x);
255                     }
256                     if (A <= Acrossover) {
257                             /* use log1p and an accurate approx to A-1 */
258                             if (ix < 0x3fff0000)
```

```
259                                 Am1 = half * (y2 / (R + xp1) + y2 / (S - xm1));
260                         else
261                                 Am1 = half * (y2 / (R + xp1) + (S + xm1));
262                         LD_IM(ans) = log1pl(Am1 + sqrtl(Am1 * (A + one)));
263                 } else {
264                         LD_IM(ans) = logl(A + sqrtl(A * A - one));
265                 }
266         }

268         if (hy >= 0)
269                 LD_IM(ans) = -LD_IM(ans);

271         return (ans);
272 }
```

_____***unchanged_portion_omitted_***

****
1547 Tue Nov 4 19:01:04 2014
new/usr/src/lib/libm/common/complex/carg.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __carg = carg
#pragma weak carg = __carg

#include "libm_synonyms.h"
#include <math.h>               /* atan2 */
#include "complex_wrapper.h"

static const double
        pi      = 3.14159265358979311600e+00,
        pi_lo   = 1.22464679914735320717e-16;

double
carg(dcomplex z) {
        int     ix, iy;

        ix = ((int *)&(D_RE(z)))[HIWORD];
        iy = ((int *)&(D_IM(z)))[HIWORD];
        if ((((ix | iy) & ~0x80000000) | ((int *)&(D_RE(z)))[LOWORD] |
            ((int *)&(D_IM(z)))[LOWORD]) == 0) {
                /* x and y are both zero */
                if (ix == 0)
                        return (D_IM(z));
                return ((iy == 0)? pi + pi_lo : -pi - pi_lo);
        }
        return (atan2(D_IM(z), D_RE(z)));
}
```

_____unchanged_portion_omitted_

****
1154 Tue Nov 4 19:01:04 2014
new/usr/src/lib/libm/common/complex/cargf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cargf = cargf
#pragma weak cargf = __cargf

#include "libm.h"                       /* atan2f */
#include "complex_wrapper.h"

float
cargf(fcomplex z) {
        return (atan2f(F_IM(z), F_RE(z)));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1148 Tue Nov  4 19:01:05 2014
new/usr/src/lib/libm/common/complex/cargl.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cargl = cargl
#pragma weak cargl = __cargl

#include "libm.h"
#include "complex_wrapper.h"

long double
cargl(ldcomplex z) {
        return (atan2l(LD_IM(z), LD_RE(z)));
}
______unchanged_portion_omitted_
```

```
**********************************************************
   12214 Tue Nov  4 19:01:05 2014
new/usr/src/lib/libm/common/complex/casin.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casin = casin
#pragma weak casin = __casin

/* INDENT OFF */
/*
 * dcomplex casin(dcomplex z);
 *
 * Alogrithm
 * (based on T.E.Hull, Thomas F. Fairgrieve and Ping Tak Peter Tang's
 * paper "Implementing the Complex Arcsine and Arccosine Functins Using
 * Exception Handling", ACM TOMS, Vol 23, pp 299-335)
 *
 * The principal value of complex inverse sine function casin(z),
 * where z = x+iy, can be defined by
 *
 *      casin(z) = asin(B) + i sign(y) log (A + sqrt(A*A-1)),
 *
 * where the log function is the natural log, and
 *                 ____________             ____________
 *       1   /     2    2      1   /     2    2
 *  A = ---  / (x+1)  + y   +  ---  / (x-1)  + y
 *       2 \/                  2 \/
 *                 ____________             ____________
 *       1   /     2    2      1   /     2    2
 *  B = ---  / (x+1)  + y   -  ---  / (x-1)  + y   .
 *       2 \/                  2 \/
 *
 * The Branch cuts are on the real line from -inf to -1 and from 1 to inf.
 * The real and imaginary parts are based on Abramowitz and Stegun
 * [Handbook of Mathematic Functions, 1972].  The sign of the imaginary
 * part is chosen to be the generally considered the principal value of
 * this function.
```



```
 *
 * Notes:1. A is the average of the distances from z to the points (1,0)
 *          and (-1,0) in the complex z-plane, and in particular A>=1.
 *       2. B is in [-1,1], and A*B = x.
 *
 * Special notes: if casin( x, y) = ( u, v), then
 *                  casin(-x, y) = (-u, v),
 *                  casin( x,-y) = ( u,-v),
 *    in general, we have casin(conj(z))     =  conj(casin(z))
 *                       casin(-z)          = -casin(z)
 *                       casin(z)           =  pi/2 - cacos(z)
 *
 * EXCEPTION CASES (conform to ISO/IEC 9899:1999(E)):
 *    casin( 0 + i 0   ) =  0    + i 0
 *    casin( 0 + i NaN ) =  0    + i NaN
 *    casin( x + i inf ) =  0    + i inf for finite x
 *    casin( x + i NaN ) =  NaN  + i NaN with invalid for finite x != 0
 *    casin(inf + iy   ) =  pi/2 + i inf finite y
 *    casin(inf + i inf) =  pi/4 + i inf
 *    casin(inf + i NaN) =  NaN  + i inf
 *    casin(NaN + i y  ) =  NaN  + i NaN for finite y
 *    casin(NaN + i inf) =  NaN  + i inf
 *    casin(NaN + i NaN) =  NaN  + i NaN
 *
 * Special Regions (better formula for accuracy and for avoiding spurious
 * overflow or underflow) (all x and y are assumed nonnegative):
 *  case 1: y = 0
 *  case 2: tiny y relative to x-1: y <= ulp(0.5)*|x-1|
 *  case 3: tiny y: y < 4 sqrt(u), where u = minimum normal number
 *  case 4: huge y relative to x+1: y >= (1+x)/ulp(0.5)
 *  case 5: huge x and y: x and y >= sqrt(M)/8, where M = maximum normal number
 *  case 6: tiny x: x < 4 sqrt(u)
 *  --------
 *  case       1 & 2. y=0 or y/|x-1| is tiny. We have
 *                ____________              _____________
 *              /      2    2             /      y    2
 *             / (x+-1)  + y   =  |x+-1| / 1 + (------)
 *           \/                        \/       |x+-1|
 *
 *                                           1      y   2
 *                          ~  |x+-1| ( 1 + --- (------)  )
 *                                           2   |x+-1|
 *
 *                                          2
 *                                         y
 *                          =  |x+-1| + --------.
 *                                      2|x+-1|
 *
 *      Consequently, it is not difficult to see that
 *                                 2
 *                                y
 *                    [ 1 + ------------ ,  if x < 1,
 *                    [      2(1+x)(1-x)
 *                    [
 *                    [
 *                    [ x,                if x = 1 (y = 0),
 *                    [
 *             A ~=   [                2
 *                    [            x * y
 *                    [ x + ------------ ,  if x > 1
 *                    [      2(1+x)(x-1)
 *
 *      and hence
 *                      ______                                 2
 *                     / 2                    y               y
 *              A + \/ A  - 1  ~  1 + ---------------- + -----------, if x < 1,
```

```
 127  *                                   sqrt((x+1)(1-x))   2(x+1)(1-x)
 128  *
 129  *
 130  *                             ~  x + sqrt((x-1)*(x+1)),             if x >= 1.
 131  *
 132  *                                      2
 133  *                                     y
 134  *                           [ x(1 - ------------), if x < 1,
 135  *                           [       2(1+x)(1-x)
 136  *              B = x/A  ~   [
 137  *                           [ 1,                  if x = 1,
 138  *                           [
 139  *                           [          2
 140  *                           [         y
 141  *                           [ 1 - ------------ ,   if x > 1,
 142  *                           [      2(1+x)(1-x)
 143  *      Thus
 144  *                             [ asin(x) + i y/sqrt((x-1)*(x+1)), if x <  1
 145  *              casin(x+i*y)=[
 146  *                             [ pi/2    + i log(x+sqrt(x*x-1)),  if x >= 1
 147  *
 148  *  case 3. y < 4 sqrt(u), where u = minimum normal x.
 149  *      After case 1 and 2, this will only occurs when x=1. When x=1, we have
 150  *         A = (sqrt(4+y*y)+y)/2 ~ 1 + y/2 + y^2/8 + ...
 151  *      and
 152  *         B = 1/A = 1 - y/2 + y^2/8 + ...
 153  *      Since
 154  *         asin(x) = pi/2-2*asin(sqrt((1-x)/2))
 155  *         asin(x) = x + x^3/6 + x^5*3/40 + x^7*15/336 + ...
 156  *      we have, for the real part asin(B),
 157  *         asin(1-y/2) ~ pi/2 - 2 asin(sqrt(y/4))
 158  *                     ~ pi/2 - sqrt(y)
 159  *      For the imaginary part,
 160  *         log(A+sqrt(A*A-1)) ~ log(1+y/2+sqrt(2*y/2))
 161  *                            = log(1+y/2+sqrt(y))
 162  *                            = (y/2+sqrt(y)) - (y/2+sqrt(y))^2/2 + ...
 163  *                            ~ sqrt(y) - y*(sqrt(y)+y/2)/2
 164  *                            ~ sqrt(y)
 165  *
 166  *  case 4. y >= (x+1)ulp(0.5). In this case, A ~ y and B ~ x/y. Thus
 167  *         real part = asin(B) ~ x/y (be careful, x/y may underflow)
 168  *      and
 169  *         imag part = log(y+sqrt(y*y-one))
 170  *
 171  *
 172  *  case 5. Both x and y are large: x and y > sqrt(M)/8, where M = maximum x
 173  *      In this case,
 174  *         A ~ sqrt(x*x+y*y)
 175  *         B ~ x/sqrt(x*x+y*y).
 176  *      Thus
 177  *         real part = asin(B) = atan(x/y),
 178  *         imag part = log(A+sqrt(A*A-1)) ~ log(2A)
 179  *                   = log(2) + 0.5*log(x*x+y*y)
 180  *                   = log(2) + log(y) + 0.5*log(1+(x/y)^2)
 181  *
 182  *  case 6. x < 4 sqrt(u). In this case, we have
 183  *         A ~ sqrt(1+y*y), B = x/sqrt(1+y*y).
 184  *      Since B is tiny, we have
 185  *          real part = asin(B) ~ B = x/sqrt(1+y*y)
 186  *          imag part = log(A+sqrt(A*A-1)) = log (A+sqrt(y*y))
 187  *                    = log(y+sqrt(1+y*y))
 188  *                    = 0.5*log(y^2+2ysqrt(1+y^2)+1+y^2)
 189  *                    = 0.5*log(1+2y(y+sqrt(1+y^2)));
 190  *                    = 0.5*log1p(2y(y+A));
 191  *
 192  *      casin(z) = asin(B) + i sign(y) log (A + sqrt(A*A-1)),
```



```
 193  */
 194 /* INDENT ON */

 196 #include "libm.h"               /* asin/atan/fabs/log/log1p/sqrt */
 197 #include "complex_wrapper.h"

 199 /* INDENT OFF */
 200 static const double
 201         zero = 0.0,
 202         one = 1.0,
 203         E = 1.1102230246251654042e-16,                  /* 2**-53 */
 204         ln2 = 6.93147180559945286227e-01,
 205         pi_2 = 1.57079632679489656e+00,
 206         pi_2_l = 6.123233995736765886e-17,
 207         pi_4 = 7.85398163397448278999e-01,
 208         Foursqrtu = 5.96667258496016539463e-154,        /* 2**(-509) */
 209         Acrossover = 1.5,
 210         Bcrossover = 0.6417,
 211         half = 0.5;
 212 /* INDENT ON */

 214 dcomplex
 215 casin(dcomplex z) {
 216         double x, y, t, R, S, A, Am1, B, y2, xm1, xp1, Apx;
 217         int ix, iy, hx, hy;
 218         unsigned lx, ly;
 219         dcomplex ans;

 221         x = D_RE(z);
 222         y = D_IM(z);
 223         hx = HI_WORD(x);
 224         lx = LO_WORD(x);
 225         hy = HI_WORD(y);
 226         ly = LO_WORD(y);
 227         ix = hx & 0x7fffffff;
 228         iy = hy & 0x7fffffff;
 229         x = fabs(x);
 230         y = fabs(y);

 232         /* special cases */

 234         /* x is inf or NaN */
 235         if (ix >= 0x7ff00000) { /* x is inf or NaN */
 236                 if (ISINF(ix, lx)) {    /* x is INF */
 237                         D_IM(ans) = x;
 238                         if (iy >= 0x7ff00000) {
 239                                 if (ISINF(iy, ly))
 240                                         /* casin(inf + i inf) = pi/4 + i inf */
 241                                         D_RE(ans) = pi_4;
 242                                 else    /* casin(inf + i NaN) = NaN  + i inf  */
 243                                         D_RE(ans) = y + y;
 244                         } else  /* casin(inf + iy) = pi/2 + i inf */
 245                                 D_RE(ans) = pi_2;
 246                 } else {                /* x is NaN */
 247                         if (iy >= 0x7ff00000) {
 248                                 /* INDENT OFF */
 249                                 /*
 250                                  * casin(NaN + i inf) = NaN + i inf
 251                                  * casin(NaN + i NaN) = NaN + i NaN
 252                                  */
 253                                 /* INDENT ON */
 254                                 D_IM(ans) = y + y;
 255                                 D_RE(ans) = x + x;
 256                         } else {
 257                                 /* casin(NaN + i y ) = NaN  + i NaN */
 258                                 D_IM(ans) = D_RE(ans) = x + y;
```

```
259                         }
260                 }
261                 if (hx < 0)
262                         D_RE(ans) = -D_RE(ans);
263                 if (hy < 0)
264                         D_IM(ans) = -D_IM(ans);
265                 return (ans);
266         }

268         /* casin(+0 + i 0  ) =  0   + i 0. */
269         if ((ix | lx | iy | ly) == 0)
270                 return (z);

272         if (iy >= 0x7ff00000) { /* y is inf or NaN */
273                 if (ISINF(iy, ly)) {    /* casin(x + i inf) =  0   + i inf */
274                         D_IM(ans) = y;
275                         D_RE(ans) = zero;
276                 } else {                /* casin(x + i NaN) = NaN  + i NaN */
277                         D_IM(ans) = x + y;
278                         if ((ix | lx) == 0)
279                                 D_RE(ans) = x;
280                         else
281                                 D_RE(ans) = y;
282                 }
283                 if (hx < 0)
284                         D_RE(ans) = -D_RE(ans);
285                 if (hy < 0)
286                         D_IM(ans) = -D_IM(ans);
287                 return (ans);
288         }

290         if ((iy | ly) == 0) {   /* region 1: y=0 */
291                 if (ix < 0x3ff00000) {  /* |x| < 1 */
292                         D_RE(ans) = asin(x);
293                         D_IM(ans) = zero;
294                 } else {
295                         D_RE(ans) = pi_2;
296                         if (ix >= 0x43500000)   /* |x| >= 2**54 */
297                                 D_IM(ans) = ln2 + log(x);
298                         else if (ix >= 0x3ff80000)      /* x > Acrossover */
299                                 D_IM(ans) = log(x + sqrt((x - one) * (x +
300                                         one)));
301                         else {
302                                 xm1 = x - one;
303                                 D_IM(ans) = log1p(xm1 + sqrt(xm1 * (x + one)));
304                         }
305                 }
306         } else if (y <= E * fabs(x - one)) {    /* region 2: y < tiny*|x-1| */
307                 if (ix < 0x3ff00000) {  /* x < 1 */
308                         D_RE(ans) = asin(x);
309                         D_IM(ans) = y / sqrt((one + x) * (one - x));
310                 } else {
311                         D_RE(ans) = pi_2;
312                         if (ix >= 0x43500000) { /* |x| >= 2**54 */
313                                 D_IM(ans) = ln2 + log(x);
314                         } else if (ix >= 0x3ff80000)    /* x > Acrossover */
315                                 D_IM(ans) = log(x + sqrt((x - one) * (x +
316                                         one)));
317                         else
318                                 D_IM(ans) = log1p((x - one) + sqrt((x - one) *
319                                         (x + one)));
320                 }
321         } else if (y < Foursqrtu) {     /* region 3 */
322                 t = sqrt(y);
323                 D_RE(ans) = pi_2 - (t - pi_2_l);
324                 D_IM(ans) = t;
```



```
325         } else if (E * y - one >= x) {  /* region 4 */
326                 D_RE(ans) = x / y;      /* need to fix underflow cases */
327                 D_IM(ans) = ln2 + log(y);
328         } else if (ix >= 0x5fc00000 || iy >= 0x5fc00000) {      /* x,y>2**509 */
329                 /* region 5: x+1 or y is very large (>= sqrt(max)/8) */
330                 t = x / y;
331                 D_RE(ans) = atan(t);
332                 D_IM(ans) = ln2 + log(y) + half * log1p(t * t);
333         } else if (x < Foursqrtu) {
334                 /* region 6: x is very small, < 4sqrt(min) */
335                 A = sqrt(one + y * y);
336                 D_RE(ans) = x / A;      /* may underflow */
337                 if (iy >= 0x3ff80000)   /* if y > Acrossover */
338                         D_IM(ans) = log(y + A);
339                 else
340                         D_IM(ans) = half * log1p((y + y) * (y + A));
341         } else {        /* safe region */
342                 y2 = y * y;
343                 xp1 = x + one;
344                 xm1 = x - one;
345                 R = sqrt(xp1 * xp1 + y2);
346                 S = sqrt(xm1 * xm1 + y2);
347                 A = half * (R + S);
348                 B = x / A;

350                 if (B <= Bcrossover)
351                         D_RE(ans) = asin(B);
352                 else {          /* use atan and an accurate approx to a-x */
353                         Apx = A + x;
354                         if (x <= one)
355                                 D_RE(ans) = atan(x / sqrt(half * Apx * (y2 /
356                                         (R + xp1) + (S - xm1))));
357                         else
358                                 D_RE(ans) = atan(x / (y * sqrt(half * (Apx /
359                                         (R + xp1) + Apx / (S + xm1)))));
360                 }
361                 if (A <= Acrossover) {
362                         /* use log1p and an accurate approx to A-1 */
363                         if (x < one)
364                                 Am1 = half * (y2 / (R + xp1) + y2 / (S - xm1));
365                         else
366                                 Am1 = half * (y2 / (R + xp1) + (S + xm1));
367                         D_IM(ans) = log1p(Am1 + sqrt(Am1 * (A + one)));
368                 } else {
369                         D_IM(ans) = log(A + sqrt(A * A - one));
370                 }
371         }

373         if (hx < 0)
374                 D_RE(ans) = -D_RE(ans);
375         if (hy < 0)
376                 D_IM(ans) = -D_IM(ans);

378         return (ans);
379 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1313 Tue Nov  4 19:01:06 2014
new/usr/src/lib/libm/common/complex/casinf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casinf = casinf
#pragma weak casinf = __casinf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
casinf(fcomplex z) {
        dcomplex dz, dans;
        fcomplex ans;

        D_RE(dz) = (double) (F_RE(z));
        D_IM(dz) = (double) (F_IM(z));
        dans = casin(dz);
        F_RE(ans) = (float) (D_RE(dans));
        F_IM(ans) = (float) (D_IM(dans));
        return (ans);
}
```

_____unchanged_portion_omitted_

****
1348 Tue Nov 4 19:01:06 2014
new/usr/src/lib/libm/common/complex/casinh.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casinh = casinh
#pragma weak casinh = __casinh

/* INDENT OFF */
/*
 * dcomplex casinh(dcomplex z);
 * casinh z = -i casin iz .
 */
/* INDENT ON */

#include "libm.h"
#include "complex_wrapper.h"

dcomplex
casinh(dcomplex z) {
        dcomplex w, r, ans;

        D_RE(w) = -D_IM(z);
        D_IM(w) = D_RE(z);
        r = casin(w);
        D_RE(ans) = D_IM(r);
        D_IM(ans) = -D_RE(r);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

****
1251 Tue Nov 4 19:01:06 2014
new/usr/src/lib/libm/common/complex/casinhf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casinhf = casinhf
#pragma weak casinhf = __casinhf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
casinhf(fcomplex z) {
        fcomplex w, r, ans;

        F_RE(w) = -F_IM(z);
        F_IM(w) = F_RE(z);
        r = casinf(w);
        F_RE(ans) = F_IM(r);
        F_IM(ans) = -F_RE(r);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

****
1262 Tue Nov 4 19:01:07 2014
new/usr/src/lib/libm/common/complex/casinhl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casinhl = casinhl
#pragma weak casinhl = __casinhl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
casinhl(ldcomplex z) {
        ldcomplex w, r, ans;

        LD_RE(w) = -LD_IM(z);
        LD_IM(w) = LD_RE(z);
        r = casinl(w);
        LD_RE(ans) = LD_IM(r);
        LD_IM(ans) = -LD_RE(r);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6438 Tue Nov  4 19:01:07 2014
new/usr/src/lib/libm/common/complex/casinl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __casinl = casinl
#pragma weak casinl = __casinl

#include "libm.h"               /* asinl/atanl/fabsl/isinfl/log1pl/logl/sqrtl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double
zero = 0.0L,
one = 1.0L,
Acrossover = 1.5L,
Bcrossover = 0.6417L,
half = 0.5L,
ln2 = 6.93147180559945309417232121458176568075e-0001L,
Foursqrtu = 7.33441547021938866248564956819393266382555e-2466L,  /* 2**-8189 */
#if defined(__x86)
E = 5.42101086242752217003726400434970855712890e-20L,    /* 2**-64 */
pi_4 = 0.78539816339744830957395921312272713294078130L,
pi_4_l = 4.16687145926043916414793223426701930367048980e-20L,
pi_2 = 1.57079632679489661914798426245454265881562600L,
pi_2_l = 8.33374291852087832829586446853403860734097960e-20L;

#else
E = 9.62964972193617926527988971292463659269050e-35L,    /* 2**-113 */
pi_4 = 0.78539816339744830961566084581987569936976700L,
pi_4_l = 2.16795253253094525619926100651083799219058080e-35L,
pi_2 = 1.57079632679489661923132169163975139873953400L,
pi_2_l = 4.33590506506189051239852201302167598438116160e-35L;

#endif
/* INDENT ON */
```



```
#if defined(__x86)
static const int ip1 = 0x40400000;      /* 2**65 */
#else
static const int ip1 = 0x40710000;      /* 2**114 */
#endif

ldcomplex
casinl(ldcomplex z) {
        long double x, y, t, R, S, A, Am1, B, y2, xm1, xp1, Apx;
        int ix, iy, hx, hy;
        ldcomplex ans;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        hy = HI_XWORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        x = fabsl(x);
        y = fabsl(y);

        /* special cases */

        /* x is inf or NaN */
        if (ix >= 0x7fff0000) { /* x is inf or NaN */
                if (isinfl(x)) {        /* x is INF */
                        LD_IM(ans) = x;
                        if (iy >= 0x7fff0000) {
                                if (isinfl(y))
                                        /* casin(inf + i inf) = pi/4 + i inf */
                                        LD_RE(ans) = pi_4 + pi_4_l;
                                else    /* casin(inf + i NaN) = NaN + i inf */
                                        LD_RE(ans) = y + y;
                        } else  /* casin(inf + iy) = pi/2 + i inf */
                                LD_RE(ans) = pi_2 + pi_2_l;
                } else {                /* x is NaN */
                        if (iy >= 0x7fff0000) {
                                /* INDENT OFF */
                                /*
                                 * casin(NaN + i inf) = NaN  + i inf
                                 * casin(NaN + i NaN) = NaN  + i NaN
                                 */
                                /* INDENT ON */
                                LD_IM(ans) = y + y;
                                LD_RE(ans) = x + x;
                        } else {
                                /* INDENT OFF */
                                /* casin(NaN + i y ) = NaN  + i NaN */
                                /* INDENT ON */
                                LD_IM(ans) = LD_RE(ans) = x + y;
                        }
                }
                if (hx < 0)
                        LD_RE(ans) = -LD_RE(ans);
                if (hy < 0)
                        LD_IM(ans) = -LD_IM(ans);
                return (ans);
        }

        /* casin(+0 + i 0) = 0 + i 0. */
        if (x == zero && y == zero)
                return (z);

        if (iy >= 0x7fff0000) { /* y is inf or NaN */
                if (isinfl(y)) {        /* casin(x + i inf) = 0 + i inf */
```

```
127                         LD_IM(ans) = y;
128                         LD_RE(ans) = zero;
129                 } else {                /* casin(x + i NaN) = NaN + i NaN */
130                         LD_IM(ans) = x + y;
131                         if (x == zero)
132                                 LD_RE(ans) = x;
133                         else
134                                 LD_RE(ans) = y;
135                 }
136                 if (hx < 0)
137                         LD_RE(ans) = -LD_RE(ans);
138                 if (hy < 0)
139                         LD_IM(ans) = -LD_IM(ans);
140                 return (ans);
141         }

143         if (y == zero) {        /* region 1: y=0 */
144                 if (ix < 0x3fff0000) {  /* |x| < 1 */
145                         LD_RE(ans) = asinl(x);
146                         LD_IM(ans) = zero;
147                 } else {
148                         LD_RE(ans) = pi_2 + pi_2_l;
149                         if (ix >= ip1)  /* |x| >= i386 ? 2**65 : 2**114 */
150                                 LD_IM(ans) = ln2 + logl(x);
151                         else if (ix >= 0x3fff8000)      /* x > Acrossover */
152                                 LD_IM(ans) = logl(x + sqrtl((x - one) * (x +
153                                         one)));
154                         else {
155                                 xm1 = x - one;
156                                 LD_IM(ans) = log1pl(xm1 + sqrtl(xm1 * (x +
157                                         one)));
158                         }
159                 }
160         } else if (y <= E * fabsl(x - one)) {   /* region 2: y < tiny*|x-1| */
161                 if (ix < 0x3fff0000) {  /* x < 1 */
162                         LD_RE(ans) = asinl(x);
163                         LD_IM(ans) = y / sqrtl((one + x) * (one - x));
164                 } else {
165                         LD_RE(ans) = pi_2 + pi_2_l;
166                         if (ix >= ip1)  /* i386 ? 2**65 : 2**114 */
167                                 LD_IM(ans) = ln2 + logl(x);
168                         else if (ix >= 0x3fff8000)      /* x > Acrossover */
169                                 LD_IM(ans) = logl(x + sqrtl((x - one) * (x +
170                                         one)));
171                         else
172                                 LD_IM(ans) = log1pl((x - one) + sqrtl((x -
173                                         one) * (x + one)));
174                 }
175         } else if (y < Foursqrtu) {     /* region 3 */
176                 t = sqrtl(y);
177                 LD_RE(ans) = pi_2 - (t - pi_2_l);
178                 LD_IM(ans) = t;
179         } else if (E * y - one >= x) {  /* region 4 */
180                 LD_RE(ans) = x / y;     /* need to fix underflow cases */
181                 LD_IM(ans) = ln2 + logl(y);
182         } else if (ix >= 0x5ffb0000 || iy >= 0x5ffb0000) {
183                 /* region 5: x+1 and y are both (>= sqrt(max)/8) i.e. 2**8188 */
184                 t = x / y;
185                 LD_RE(ans) = atanl(t);
186                 LD_IM(ans) = ln2 + logl(y) + half * log1pl(t * t);
187         } else if (x < Foursqrtu) {
188                 /* region 6: x is very small, < 4sqrt(min) */
189                 A = sqrtl(one + y * y);
190                 LD_RE(ans) = x / A;     /* may underflow */
191                 if (iy >= 0x3fff8000)   /* if y > Acrossover */
192                         LD_IM(ans) = logl(y + A);
```



```
193                 else
194                         LD_IM(ans) = half * log1pl((y + y) * (y + A));
195         } else {        /* safe region */
196                 y2 = y * y;
197                 xp1 = x + one;
198                 xm1 = x - one;
199                 R = sqrtl(xp1 * xp1 + y2);
200                 S = sqrtl(xm1 * xm1 + y2);
201                 A = half * (R + S);
202                 B = x / A;
203                 if (B <= Bcrossover)
204                         LD_RE(ans) = asinl(B);
205                 else {          /* use atan and an accurate approx to a-x */
206                         Apx = A + x;
207                         if (x <= one)
208                                 LD_RE(ans) = atanl(x / sqrtl(half * Apx * (y2 /
209                                         (R + xp1) + (S - xm1))));
210                         else
211                                 LD_RE(ans) = atanl(x / (y * sqrtl(half * (Apx /
212                                         (R + xp1) + Apx / (S + xm1)))));
213                 }
214                 if (A <= Acrossover) {
215                         /* use log1p and an accurate approx to A-1 */
216                         if (x < one)
217                                 Am1 = half * (y2 / (R + xp1) + y2 / (S - xm1));
218                         else
219                                 Am1 = half * (y2 / (R + xp1) + (S + xm1));
220                         LD_IM(ans) = log1pl(Am1 + sqrtl(Am1 * (A + one)));
221                 } else {
222                         LD_IM(ans) = logl(A + sqrtl(A * A - one));
223                 }
224         }

226         if (hx < 0)
227                 LD_RE(ans) = -LD_RE(ans);
228         if (hy < 0)
229                 LD_IM(ans) = -LD_IM(ans);

231         return (ans);
232 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    8750 Tue Nov  4 19:01:08 2014
new/usr/src/lib/libm/common/complex/catan.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catan = catan
#pragma weak catan = __catan

/* INDENT OFF */
/*
 * dcomplex catan(dcomplex z);
 *
 * If
 *     z = x + iy,
 *
 * then
 *          1       (    2x     )    1                2    2
 * Re w  =  - arctan(-----------)  = - ATAN2(2x, 1 - x  - y )
 *          2       (     2    2)    2
 *                  (1 - x  - y )
 *
 *                ( 2         2)
 *          1     (x  +  (y+1) )      1                 4y
 * Im w  =  - log(------------) .=  --- log [ 1 + ------------- ]
 *          4     ( 2         2)      4              2         2
 *                (x  +  (y-1) )                   x  +  (y-1)
 *
 *                  2    16  3                          y
 *         = t - 2t   + -- t  - ..., where t = -----------------
 *                      3                       x*x + (y-1)*(y-1)
 *
 * Note that: if catan( x, y) = ( u, v), then
 *              catan(-x, y) = (-u, v)
 *              catan( x,-y) = ( u,-v)
 *
 * Also,   catan(x,y) = -i*catanh(-y,x), or
 *        catanh(x,y) =  i*catan(-y,x)
```



```
 * So, if catanh(y,x) = (v,u), then catan(x,y) = -i*(-v,u) = (u,v), i.e.,
 *        catan(x,y) = (u,v)
 *
 * EXCEPTION CASES (conform to ISO/IEC 9899:1999(E)):
 *    catan( 0  , 0   ) =  (0    ,  0   )
 *    catan( NaN, 0   ) =  (NaN  ,  0   )
 *    catan( 0  , 1   ) =  (0    ,  +inf) with divide-by-zero
 *    catan( inf, y   ) =  (pi/2 ,  0   ) for finite +y
 *    catan( NaN, y   ) =  (NaN  ,  NaN ) with invalid for finite y != 0
 *    catan( x  , inf ) =  (pi/2 ,  0   ) for finite +x
 *    catan( inf, inf ) =  (pi/2 ,  0   )
 *    catan( NaN, inf ) =  (NaN  ,  0   )
 *    catan( x  , NaN ) =  (NaN  ,  NaN ) with invalid for finite x
 *    catan( inf, NaN ) =  (pi/2 ,  +-0 )
 */
/* INDENT ON */

#include "libm.h"               /* atan/atan2/fabs/log/log1p */
#include "complex_wrapper.h"

/* INDENT OFF */
static const double
        pi_2 = 1.57079632679489656e+00,
        zero = 0.0,
        half = 0.5,
        two = 2.0,
        ln2 = 6.93147180559945309417232121458176568075e-0001,
        one = 1.0;
/* INDENT ON */

dcomplex
catan(dcomplex z) {
        dcomplex ans;
        double x, y, ax, ay, t;
        int hx, hy, ix, iy;
        unsigned lx, ly;

        x = D_RE(z);
        y = D_IM(z);
        ax = fabs(x);
        ay = fabs(y);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;

        /* x is inf or NaN */
        if (ix >= 0x7ff00000) {
                if (ISINF(ix, lx)) {
                        D_RE(ans) = pi_2;
                        D_IM(ans) = zero;
                } else {
                        D_RE(ans) = x + x;
                        if ((iy | ly) == 0 || (ISINF(iy, ly)))
                                D_IM(ans) = zero;
                        else
                                D_IM(ans) = (fabs(y) - ay) / (fabs(y) - ay);
                }
        } else if (iy >= 0x7ff00000) {
                /* y is inf or NaN */
                if (ISINF(iy, ly)) {
                        D_RE(ans) = pi_2;
                        D_IM(ans) = zero;
                } else {
```

```
127                         D_RE(ans) = (fabs(x) - ax) / (fabs(x) - ax);
128                         D_IM(ans) = y;
129                 }
130         } else if ((ix | lx) == 0) {
131                 /* INDENT OFF */
132                 /*
133                  * x = 0
134                  *      1                              1
135                  * A = --- * atan2(2x, 1-x*x-y*y) = --- atan2(0,1-|y|)
136                  *      2                              2
137                  *
138                  *     1     [  (y+1)*(y+1) ]   1          2      1         2y
139                  * B = - log [ ------------ ] = - log (1+ ---) or - log(1+ ----)
140                  *     4     [  (y-1)*(y-1) ]   2         y-1     2         1-y
141                  */
142                 /* INDENT ON */
143                 t = one - ay;
144                 if (((iy - 0x3ff00000) | ly) == 0) {
145                         /* y=1: catan(0,1)=(0,+inf) with 1/0 signal */
146                         D_IM(ans) = ay / ax;
147                         D_RE(ans) = zero;
148                 } else if (iy >= 0x3ff00000) {  /* y>1 */
149                         D_IM(ans) = half * log1p(two / (-t));
150                         D_RE(ans) = pi_2;
151                 } else {                        /* y<1 */
152                         D_IM(ans) = half * log1p((ay + ay) / t);
153                         D_RE(ans) = zero;
154                 }
155         } else if (iy < 0x3e200000 || ((ix - iy) >> 20) >= 30) {
156         /* INDENT OFF */
157         /*
158          * Tiny y (relative to 1+|x|)
159          *     |y| < E*(1+|x|)
160          * where E=2**-29, -35, -60 for double, double extended, quad precision
161          *
162          *      1                                [ x<=1:   atan(x)
163          * A = --- * atan2(2x, 1-x*x-y*y) ~ [       1                1+x
164          *      2                                [ x>=1: - atan2(2,(1-x)*(-----))
165          *                                               2                 x
166          *
167          *                               y/x
168          * B ~ t*(1-2t), where t = ----------------- is tiny
169          *                         x + (y-1)*(y-1)/x
170          */
171                 /* INDENT ON */
172                 if (ix < 0x3ff00000)
173                         D_RE(ans) = atan(ax);
174                 else
175                         D_RE(ans) = half * atan2(two, (one - ax) * (one +
176                                 one / ax));
177                 if ((iy | ly) == 0) {
178                         D_IM(ans) = ay;
179                 } else {
180                         if (ix < 0x3e200000)
181                                 t = ay / ((ay - one) * (ay - one));
182                         else if (ix > 0x41c00000)
183                                 t = (ay / ax) / ax;
184                         else
185                                 t = ay / (ax * ax + (ay - one) * (ay - one));
186                         D_IM(ans) = t * (one - (t + t));
187                 }
188         } else if (iy >= 0x41c00000 && ((iy - ix) >> 20) >= 30) {
189                 /* INDENT OFF */
190                 /*
191                  * Huge y relative to 1+|x|
192                  *            |y| > Einv*(1+|x|), where Einv~2**(prec/2+3),
```



```
193                  *             1
194                  *       A ~ --- * atan2(2x, -y*y) ~ pi/2
195                  *             2
196                  *                                      y
197                  *       B ~ t*(1-2t), where t = ---------------- is tiny
198                  *                                   (y-1)*(y-1)
199                  */
200                 /* INDENT ON */
201                 D_RE(ans) = pi_2;
202                 t = (ay / (ay - one)) / (ay - one);
203                 D_IM(ans) = t * (one - (t + t));
204         } else if (((iy - 0x3ff00000) | ly) == 0) {
205                 /* INDENT OFF */
206                 /*
207                  * y = 1
208                  *      1                          1
209                  * A = --- * atan2(2x, -x*x) = --- atan2(2,-x)
210                  *      2                          2
211                  *
212                  *     1     [x*x + 4]   1          4     [ 0.5(log2-logx) if
213                  * B = - log [-------] = - log (1+ ---) = [ |x|<E, else 0.25*
214                  *     4     [  x*x  ]   4         x*x    [ log1p((2/x)*(2/x))
215                  */
216                 /* INDENT ON */
217                 D_RE(ans) = half * atan2(two, -ax);
218                 if (ix < 0x3e200000)
219                         D_IM(ans) = half * (ln2 - log(ax));
220                 else {
221                         t = two / ax;
222                         D_IM(ans) = 0.25 * log1p(t * t);
223                 }
224         } else if (ix >= 0x43900000) {
225                 /* INDENT OFF */
226                 /*
227                  * Huge x:
228                  * when |x| > 1/E^2,
229                  *      1                          pi
230                  * A ~ --- * atan2(2x, -x*x-y*y) ~ ---
231                  *      2                          2
232                  *                                    y                 y/x
233                  * B ~ t*(1-2t), where t = ---------------- = (-------------- )/x
234                  *                              x*x+(y-1)*(y-1)     1+((y-1)/x)^2
235                  */
236                 /* INDENT ON */
237                 D_RE(ans) = pi_2;
238                 t = ((ay / ax) / (one + ((ay - one) / ax) * ((ay - one) /
239                         ax))) / ax;
240                 D_IM(ans) = t * (one - (t + t));
241         } else if (ix < 0x38b00000) {
242                 /* INDENT OFF */
243                 /*
244                  * Tiny x:
245                  * when |x| < E^4,  (note that y != 1)
246                  *      1                           1
247                  * A = --- * atan2(2x, 1-x*x-y*y) ~ --- * atan2(2x,(1-y)*(1+y))
248                  *      2                           2
249                  *
250                  *     1     [(y+1)*(y+1)]   1          2      1         2y
251                  * B = - log [-----------] = - log (1+ ---) or - log(1+ ----)
252                  *     4     [(y-1)*(y-1)]   2         y-1     2         1-y
253                  */
254                 /* INDENT ON */
255                 D_RE(ans) = half * atan2(ax + ax, (one - ay) * (one + ay));
256                 if (iy >= 0x3ff00000)
257                         D_IM(ans) = half * log1p(two / (ay - one));
258                 else
```

```
                        D_IM(ans) = half * log1p((ay + ay) / (one - ay));
        } else {
                /* INDENT OFF */
                /*
                 * normal x,y
                 *      1
                 * A = --- * atan2(2x, 1-x*x-y*y)
                 *      2
                 *
                 *     1     [x*x+(y+1)*(y+1)]   1               4y
                 * B = - log [---------------] = - log (1+ -----------------)
                 *     4     [x*x+(y-1)*(y-1)]   4         x*x + (y-1)*(y-1)
                 */
                /* INDENT ON */
                t = one - ay;
                if (iy >= 0x3fe00000 && iy < 0x40000000) {
                        /* y close to 1 */
                        D_RE(ans) = half * (atan2((ax + ax), (t * (one + ay) -
                                ax * ax)));
                } else if (ix >= 0x3fe00000 && ix < 0x40000000) {
                        /* x close to 1 */
                        D_RE(ans) = half * atan2((ax + ax), ((one - ax) *
                                (one + ax) - ay * ay));
                } else
                        D_RE(ans) = half * atan2((ax + ax), ((one - ax * ax) -
                                ay * ay));
                D_IM(ans) = 0.25 * log1p((4.0 * ay) / (ax * ax + t * t));
        }
        if (hx < 0)
                D_RE(ans) = -D_RE(ans);
        if (hy < 0)
                D_IM(ans) = -D_IM(ans);
        return (ans);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    3576 Tue Nov  4 19:01:08 2014
new/usr/src/lib/libm/common/complex/catanf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catanf = catanf
#pragma weak catanf = __catanf

#include "libm.h"
#include "complex_wrapper.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const float
        pi_2 = 1.570796326794896558e+00F,
        zero = 0.0F,
        half = 0.5F,
        two = 2.0F,
        one = 1.0F;

fcomplex
catanf(fcomplex z) {
        fcomplex        ans;
        float           x, y, ax, ay, t;
        double          dx, dy, dt;
        int             hx, hy, ix, iy;

        x = F_RE(z);
        y = F_IM(z);
        ax = fabsf(x);
        ay = fabsf(y);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
```



```
        if (ix >= 0x7f800000) {         /* x is inf or NaN */
                if (ix == 0x7f800000) {
                        F_RE(ans) = pi_2;
                        F_IM(ans) = zero;
                } else {
                        F_RE(ans) = x * x;
                        if (iy == 0 || iy == 0x7f800000)
                                F_IM(ans) = zero;
                        else
                                F_IM(ans) = (fabsf(y) - ay) / (fabsf(y) - ay);
                }
        } else if (iy >= 0x7f800000) {  /* y is inf or NaN */
                if (iy == 0x7f800000) {
                        F_RE(ans) = pi_2;
                        F_IM(ans) = zero;
                } else {
                        F_RE(ans) = (fabsf(x) - ax) / (fabsf(x) - ax);
                        F_IM(ans) = y * y;
                }
        } else if (ix == 0) {
                /* INDENT OFF */
                /*
                 * x = 0
                 *      1                               1
                 * A = --- * atan2(2x, 1-x*x-y*y) = --- atan2(0,1-|y|)
                 *      2                               2
                 *
                 *     1     [ (y+1)*(y+1) ]   1          2      1          2y
                 * B = - log [ ----------- ] = - log (1+ ---) or - log(1+ ----)
                 *     4     [ (y-1)*(y-1) ]   2         y-1     2         1-y
                 */
                /* INDENT ON */
                t = one - ay;
                if (iy == 0x3f800000) {
                        /* y=1: catan(0,1)=(0,+inf) with 1/0 signal */
                        F_IM(ans) = ay / ax;
                        F_RE(ans) = zero;
                } else if (iy > 0x3f800000) {   /* y>1 */
                        F_IM(ans) = half * log1pf(two / (-t));
                        F_RE(ans) = pi_2;
                } else {                        /* y<1 */
                        F_IM(ans) = half * log1pf((ay + ay) / t);
                        F_RE(ans) = zero;
                }
        } else {
                /* INDENT OFF */
                /*
                 * use double precision x,y
                 *      1
                 * A = --- * atan2(2x, 1-x*x-y*y)
                 *      2
                 *
                 *     1     [ x*x+(y+1)*(y+1) ]   1              4y
                 * B = - log [ --------------- ] = - log (1+ -----------------)
                 *     4     [ x*x+(y-1)*(y-1) ]   4         x*x + (y-1)*(y-1)
                 */
                /* INDENT ON */
#if defined(__i386) && !defined(__amd64)
                int     rp = __swapRP(fp_extended);
#endif
                dx = (double)ax;
                dy = (double)ay;
                F_RE(ans) = (float)(0.5 * atan2(dx + dx,
                    1.0 - dx * dx - dy * dy));
                dt = dy - 1.0;
                F_IM(ans) = (float)(0.25 * log1p(4.0 * dy /
```

```
 127                     (dx * dx + dt * dt)));
 128 #if defined(__i386) && !defined(__amd64)
 129                 if (rp != fp_extended)
 130                         (void) __swapRP(rp);
 131 #endif
 132         }
 133         if (hx < 0)
 134                 F_RE(ans) = -F_RE(ans);
 135         if (hy < 0)
 136                 F_IM(ans) = -F_IM(ans);
 137         return (ans);
 138 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1428 Tue Nov  4 19:01:09 2014
new/usr/src/lib/libm/common/complex/catanh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catanh = catanh
#pragma weak catanh = __catanh

/* INDENT OFF */
/*
 * z := x + iy
 * catanh(z)    = -i catan(iz)
 *              = -i catan(-y+ix)
 *              = (Im(catan(-y+ix)), -Re(catan(-y+ix)))
 */
/* INDENT ON */

#include "libm.h"
#include "complex_wrapper.h"

dcomplex
catanh(dcomplex z) {
        double x, y;
        dcomplex ans, ct;

        x = D_RE(z);
        y = D_IM(z);
        D_RE(z) = -y;
        D_IM(z) = x;
        ct = catan(z);
        D_RE(ans) = D_IM(ct);
        D_IM(ans) = -D_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

****
1281 Tue Nov 4 19:01:09 2014
new/usr/src/lib/libm/common/complex/catanhf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catanhf = catanhf
#pragma weak catanhf = __catanhf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
catanhf(fcomplex z) {
        float x, y;
        fcomplex ans, ct;

        x = F_RE(z);
        y = F_IM(z);
        F_RE(z) = -y;
        F_IM(z) = x;
        ct = catanf(z);
        F_RE(ans) = F_IM(ct);
        F_IM(ans) = -F_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

****
1298 Tue Nov 4 19:01:10 2014
new/usr/src/lib/libm/common/complex/catanhl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catanhl = catanhl
#pragma weak catanhl = __catanhl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
catanhl(ldcomplex z) {
        long double x, y;
        ldcomplex ans, ct;

        x = LD_RE(z);
        y = LD_IM(z);
        LD_RE(z) = -y;
        LD_IM(z) = x;
        ct = catanl(z);
        LD_RE(ans) = LD_IM(ct);
        LD_IM(ans) = -LD_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   10206 Tue Nov  4 19:01:10 2014
new/usr/src/lib/libm/common/complex/catanl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __catanl = catanl
#pragma weak catanl = __catanl

/* INDENT OFF */
/*
 * ldcomplex catanl(ldcomplex z);
 *
 * Atan(z) return A + Bi where,
 *              1
 *      A = --- * atan2(2x, 1-x*x-y*y)
 *              2
 *
 *              1      [ x*x + (y+1)*(y+1) ]   1              4y
 *       B = --- log [ ----------------- ] = - log (1+ -----------------)
 *              4      [ x*x + (y-1)*(y-1) ]   4         x*x + (y-1)*(y-1)
 *
 *                   2    16  3                        y
 *         = t - 2t   + -- t  - ..., where t = -----------------
 *                      3                      x*x + (y-1)*(y-1)
 * Proof:
 * Let w = atan(z=x+yi) = A + B i. Then tan(w) = z.
 * Since sin(w) = (exp(iw)-exp(-iw))/(2i), cos(w)=(exp(iw)+exp(-iw))/(2),
 * Let p = exp(iw), then z = tan(w) = ((p-1/p)/(p+1/p))/i, or
 * iz = (p*p-1)/(p*p+1), or, after simplification,
 *      p*p = (1+iz)/(1-iz)                                  ... (1)
 * LHS of (1) = exp(2iw) = exp(2i(A+Bi)) = exp(-2B)*exp(2iA)
 *            = exp(-2B)*(cos(2A)+i*sin(2A))                 ... (2)
 *              1-y+ix   (1-y+ix)*(1+y+ix)   1-x*x-y*y + 2xi
 * RHS of (1) = ------ = ----------------- = --------------- ... (3)
 *              1+y-ix    (1+y)**2 + x**2    (1+y)**2 + x**2
 *
 * Comparing the real and imaginary parts of (2) and (3), we have:
```



```
 *      cos(2A) : 1-x*x-y*y = sin(2A) : 2x
 * and hence
 *      tan(2A) = 2x/(1-x*x-y*y), or
 *      A = 0.5 * atan2(2x, 1-x*x-y*y)                       ... (4)
 *
 * For the imaginary part B, Note that |p*p| = exp(-2B), and
 *      |1+iz|   |i-z|   hypot(x,(y-1))
 *       |---| = |---| = --------------
 *      |1-iz|   |i+z|   hypot(x,(y+1))
 * Thus
 *                 x*x + (y+1)*(y+1)
 *      exp(4B) = -----------------, or
 *                 x*x + (y-1)*(y-1)
 *
 *           1     [x^2+(y+1)^2]   1             4y
 *       B =  - log [-----------] = - log(1+ -------------)  ... (5)
 *           4     [x^2+(y-1)^2]   4         x^2+(y-1)^2
 *
 * QED.
 *
 * Note that: if catan( x, y) = ( u, v), then
 *              catan(-x, y) = (-u, v)
 *              catan( x,-y) = ( u,-v)
 *
 * Also,   catan(x,y) = -i*catanh(-y,x), or
 *        catanh(x,y) =  i*catan(-y,x)
 * So, if catanh(y,x) = (v,u), then catan(x,y) = -i*(-v,u) = (u,v), i.e.,
 *         catan(x,y) = (u,v)
 *
 * EXCEPTION CASES (conform to ISO/IEC 9899:1999(E)):
 *    catan( 0  , 0   ) =  (0    ,  0   )
 *    catan( NaN, 0   ) =  (NaN  ,  0   )
 *    catan( 0  , 1   ) =  (0    ,  +inf) with divide-by-zero
 *    catan( inf, y   ) =  (pi/2 ,  0   ) for finite +y
 *    catan( NaN, y   ) =  (NaN  ,  NaN ) with invalid for finite y != 0
 *    catan( x  , inf ) =  (pi/2 ,  0   ) for finite +x
 *    catan( inf, inf ) =  (pi/2 ,  0   )
 *    catan( NaN, inf ) =  (NaN  ,  0   )
 *    catan( x  , NaN ) =  (NaN  ,  NaN ) with invalid for finite x
 *    catan( inf, NaN ) =  (pi/2 ,  +-0 )
 */
/* INDENT ON */

#include "libm.h"       /* atan2l/atanl/fabsl/isinfl/iszerol/log1pl/logl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double
zero = 0.0L,
one = 1.0L,
two = 2.0L,
half = 0.5L,
ln2 = 6.93147180559945309417232121458176568075e-0001L,
pi_2 = 1.57079632679489661923132169163975144209858469968755291048747 2L,
#if defined(__x86)
E = 2.91038304567337036132812500000000000000e-11L,     /* 2**-35 */
Einv = 3.43597383680000000000000000000000000000e+10L;  /* 2**+35 */
#else
E = 8.67361737988403547205962240695953369140 6e-19L,     /* 2**-60 */
Einv = 1.15292150460684697600000000000000000000e18L;   /* 2**+60 */
#endif
/* INDENT ON */

ldcomplex
catanl(ldcomplex z) {
```

```
127         ldcomplex ans;
128         long double x, y, t1, ax, ay, t;
129         int hx, hy, ix, iy;

131         x = LD_RE(z);
132         y = LD_IM(z);
133         ax = fabsl(x);
134         ay = fabsl(y);
135         hx = HI_XWORD(x);
136         hy = HI_XWORD(y);
137         ix = hx & 0x7fffffff;
138         iy = hy & 0x7fffffff;

140         /* x is inf or NaN */
141         if (ix >= 0x7fff0000) {
142                 if (isinfl(x)) {
143                         LD_RE(ans) = pi_2;
144                         LD_IM(ans) = zero;
145                 } else {
146                         LD_RE(ans) = x + x;
147                         if (iszerol(y) || (isinfl(y)))
148                                 LD_IM(ans) = zero;
149                         else
150                                 LD_IM(ans) = (fabsl(y) - ay) / (fabsl(y) - ay);
151                 }
152         } else if (iy >= 0x7fff0000) {
153                 /* y is inf or NaN */
154                 if (isinfl(y)) {
155                         LD_RE(ans) = pi_2;
156                         LD_IM(ans) = zero;
157                 } else {
158                         LD_RE(ans) = (fabsl(x) - ax) / (fabsl(x) - ax);
159                         LD_IM(ans) = y;
160                 }
161         } else if (iszerol(x)) {
162                 /* INDENT OFF */
163                 /*
164                  * x = 0
165                  *      1                               1
166                  * A = --- * atan2(2x, 1-x*x-y*y) = --- atan2(0,1-|y|)
167                  *      2                               2
168                  *
169                  *     1     [ (y+1)*(y+1) ]   1          2      1         2y
170                  * B = - log [ ----------- ] = - log (1+ ---) or - log(1+ ----)
171                  *     4     [ (y-1)*(y-1) ]   2         y-1     2         1-y
172                  */
173                 /* INDENT ON */
174                 t = one - ay;
175                 if (ay == one) {
176                         /* y=1: catan(0,1)=(0,+inf) with 1/0 signal */
177                         LD_IM(ans) = ay / ax;
178                         LD_RE(ans) = zero;
179                 } else if (ay > one) {  /* y>1 */
180                         LD_IM(ans) = half * log1pl(two / (-t));
181                         LD_RE(ans) = pi_2;
182                 } else {                /* y<1 */
183                         LD_IM(ans) = half * log1pl((ay + ay) / t);
184                         LD_RE(ans) = zero;
185                 }
186         } else if (ay < E * (one + ax)) {
187                 /* INDENT OFF */
188                 /*
189                  * Tiny y (relative to 1+|x|)
190                  *     |y| < E*(1+|x|)
191                  * where E=2**-29, -35, -60 for double, extended, quad precision
192                  *
```



```
193                  *     1                               [x<=1:   atan(x)
194                  * A = - * atan2(2x,1-x*x-y*y) ~ [      1                 1+x
195                  *     2                               [x>=1: - atan2(2,(1-x)*(-----))
196                  *                                            2                 x
197                  *
198                  *                                    y/x
199                  * B ~ t*(1-2t), where t = ------------------ is tiny
200                  *                           x + (y-1)*(y-1)/x
201                  *
202                  *                             y
203                  * (when x < 2**-60, t = ----------- )
204                  *                         (y-1)*(y-1)
205                  */
206                 /* INDENT ON */
207                 if (ay == zero)
208                         LD_IM(ans) = ay;
209                 else {
210                         t1 = ay - one;
211                         if (ix < 0x3fc30000)
212                                 t = ay / (t1 * t1);
213                         else if (ix > 0x403b0000)
214                                 t = (ay / ax) / ax;
215                         else
216                                 t = ay / (ax * ax + t1 * t1);
217                         LD_IM(ans) = t * (one - two * t);
218                 }
219                 if (ix < 0x3fff0000)
220                         LD_RE(ans) = atanl(ax);
221                 else
222                         LD_RE(ans) = half * atan2l(two, (one - ax) * (one +
223                                 one / ax));

225         } else if (ay > Einv * (one + ax)) {
226                 /* INDENT OFF */
227                 /*
228                  * Huge y relative to 1+|x|
229                  *     |y| > Einv*(1+|x|), where Einv~2**(prec/2+3),
230                  *     1
231                  * A ~ --- * atan2(2x, -y*y) ~ pi/2
232                  *     2
233                  *                             y
234                  * B ~ t*(1-2t), where t = ---------------- is tiny
235                  *                          (y-1)*(y-1)
236                  */
237                 /* INDENT ON */
238                 LD_RE(ans) = pi_2;
239                 t = (ay / (ay - one)) / (ay - one);
240                 LD_IM(ans) = t * (one - (t + t));
241         } else if (ay == one) {
242                 /* INDENT OFF */
243                 /*
244                  * y=1
245                  *     1                         1
246                  * A = - * atan2(2x, -x*x) = --- atan2(2,-x)
247                  *     2                         2
248                  *
249                  *     1     [ x*x+4]   1          4     [ 0.5(log2-logx) if
250                  * B = - log [ -----] = - log (1+ ---) = [ |x|<E, else 0.25*
251                  *     4     [  x*x ]   4         x*x    [ log1p((2/x)*(2/x))
252                  */
253                 /* INDENT ON */
254                 LD_RE(ans) = half * atan2l(two, -ax);
255                 if (ax < E)
256                         LD_IM(ans) = half * (ln2 - logl(ax));
257                 else {
258                         t = two / ax;
```

```
259                     LD_IM(ans) = 0.25L * log1pl(t * t);
260             }
261         } else if (ax > Einv * Einv) {
262             /* INDENT OFF */
263             /*
264              * Huge x:
265              * when |x| > 1/E^2,
266              *      1                           pi
267              * A ~ --- * atan2(2x, -x*x-y*y) ~ ---
268              *      2                           2
269              *                              y                y/x
270              * B ~ t*(1-2t), where t = --------------- = (-------------- )/x
271              *                         x*x+(y-1)*(y-1)     1+((y-1)/x)^2
272              */
273             /* INDENT ON */
274             LD_RE(ans) = pi_2;
275             t = ((ay / ax) / (one + ((ay - one) / ax) * ((ay - one) /
276                 ax))) / ax;
277             LD_IM(ans) = t * (one - (t + t));
278         } else if (ax < E * E * E * E) {
279             /* INDENT OFF */
280             /*
281              * Tiny x:
282              * when |x| < E^4,  (note that y != 1)
283              *      1                           1
284              * A = --- * atan2(2x, 1-x*x-y*y) ~ --- * atan2(2x,1-y*y)
285              *      2                           2
286              *
287              *     1     [ (y+1)*(y+1) ]   1          2      1         2y
288              * B = - log [ ------------ ] = - log (1+ ---) or - log(1+ ----)
289              *     4     [ (y-1)*(y-1) ]   2         y-1     2         1-y
290              */
291             /* INDENT ON */
292             LD_RE(ans) = half * atan2l(ax + ax, (one - ay) * (one + ay));
293             if (ay > one)   /* y>1 */
294                     LD_IM(ans) = half * log1pl(two / (ay - one));
295             else            /* y<1 */
296                     LD_IM(ans) = half * log1pl((ay + ay) / (one - ay));
297         } else {
298             /* INDENT OFF */
299             /*
300              * normal x,y
301              *      1
302              * A = --- * atan2(2x, 1-x*x-y*y)
303              *      2
304              *
305              *     1     [ x*x+(y+1)*(y+1) ]   1               4y
306              * B = - log [ --------------- ] = - log (1+ -----------------)
307              *     4     [ x*x+(y-1)*(y-1) ]   4         x*x + (y-1)*(y-1)
308              */
309             /* INDENT ON */
310             t = one - ay;
311             if (iy >= 0x3ffe0000 && iy < 0x40000000) {
312                     /* y close to 1 */
313                     LD_RE(ans) = half * (atan2l((ax + ax), (t * (one +
314                             ay) - ax * ax)));
315             } else if (ix >= 0x3ffe0000 && ix < 0x40000000) {
316                     /* x close to 1 */
317                     LD_RE(ans) = half * atan2l((ax + ax), ((one - ax) *
318                             (one + ax) - ay * ay));
319             } else
320                     LD_RE(ans) = half * atan2l((ax + ax), ((one - ax *
321                             ax) - ay * ay));
322             LD_IM(ans) = 0.25L * log1pl((4.0L * ay) / (ax * ax + t * t));
323         }
324         if (hx < 0)
```



```
325                 LD_RE(ans) = -LD_RE(ans);
326         if (hy < 0)
327                 LD_IM(ans) = -LD_IM(ans);
328         return (ans);
329 }
```

*unchanged_portion_omitted_*

****
1420 Tue Nov 4 19:01:11 2014
new/usr/src/lib/libm/common/complex/ccos.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ccos = ccos
#pragma weak ccos = __ccos

/* INDENT OFF */
/*
 * dcomplex ccos(dcomplex z);
 *
 * z := x+iy; since ccos(iz) = cosh(z), we have
 * ccos(z)      = ccos((-1)*(-z)) = ccos(i*i*(-z))
 *              = ccosh(i*(-z)) = ccosh(i*(-x-yi))
 *              = ccosh(y-ix)
 */
/* INDENT ON */

#include "libm.h"
#include "complex_wrapper.h"

dcomplex
ccos(dcomplex z) {
        double x, y;

        x = D_RE(z);
        y = D_IM(z);
        D_RE(z) = y;
        D_IM(z) = -x;
        return (ccosh(z));
}
```

_____unchanged_portion_omitted_

****
1198 Tue Nov 4 19:01:11 2014
new/usr/src/lib/libm/common/complex/ccosf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ccosf = ccosf
#pragma weak ccosf = __ccosf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
ccosf(fcomplex z) {
        float x, y;

        x = F_RE(z);
        y = F_IM(z);
        F_RE(z) = y;
        F_IM(z) = -x;
        return (ccoshf(z));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    3894 Tue Nov  4 19:01:11 2014
new/usr/src/lib/libm/common/complex/ccosh.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ccosh = ccosh
#pragma weak ccosh = __ccosh

/* INDENT OFF */
/*
 * dcomplex ccosh(dcomplex z);
 *
 *              z      -z       x                      -x
 *             e   +  e        e  (cos(y)+i*sin(y)) + e  (cos(-y)+i*sin(-y))
 * cosh z = -------------- =  --------------------------------------------
 *                  2                                 2
 *                      x    -x                x    -x
 *            cos(y) ( e  + e  )  + i*sin(y) (e  - e   )
 *         = --------------------------------------------
 *                                  2
 *
 *         =  cos(y) cosh(x)  + i sin(y) sinh(x)
 *
 * Implementation Note
 * -------------------
 *
 *              |x|    -|x|   |x|        -2|x|       -2|x|    -P-4
 * Note that  e   +- e    = e   ( 1 +- e     ). If e      < 2     , where
 *
 * P stands for the number of significant bits of the machine precision,
 *                                      |x|
 * then the result will be rounded to e   . Therefore, we have
 *
 *                   z
 *                  e
 *     cosh z = -----  if |x| >= (P/2 + 2)*ln2
 *                  2
```



```
 *
 * EXCEPTION (conform to ISO/IEC 9899:1999(E)):
 *      ccosh(0,0)=(1,0)
 *      ccosh(0,inf)=(NaN,+-0)
 *      ccosh(0,NaN)=(NaN,+-0)
 *      ccosh(x,inf) = (NaN,NaN) for finite non-zero x
 *      ccosh(x,NaN) = (NaN,NaN) for finite non-zero x
 *      ccosh(inf,0) = (inf, 0)
 *      ccosh(inf,y) = (inf*cos(y),inf*sin(y)) for finite non-zero y
 *      ccosh(inf,inf) = (+-inf,NaN)
 *      ccosh(inf,NaN) = (+inf,NaN)
 *      ccosh(NaN,0) = (NaN,+-0)
 *      ccosh(NaN,y) = (NaN,NaN) for non-zero y
 *      ccosh(NaN,NaN) = (NaN,NaN)
 */
/* INDENT ON */

#include "libm.h"               /* cosh/exp/fabs/scalbn/sinh/sincos/__k_cexp */
#include "complex_wrapper.h"

dcomplex
ccosh(dcomplex z) {
        double t, x, y, S, C;
        int hx, ix, lx, hy, iy, ly, n;
        dcomplex ans;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        iy = hy & 0x7fffffff;
        x = fabs(x);
        y = fabs(y);

        (void) sincos(y, &S, &C);
        if (ix >= 0x403c0000) { /* |x| > 28 = prec/2 (14,28,34,60) */
                if (ix >= 0x40862E42) { /* |x| > 709.78... ~ log(2**1024) */
                        if (ix >= 0x7ff00000) { /* |x| is inf or NaN */
                                if ((iy | ly) == 0) {
                                        D_RE(ans) = x;
                                        D_IM(ans) = y;
                                } else if (iy >= 0x7ff00000) {
                                        D_RE(ans) = x;
                                        D_IM(ans) = x - y;
                                } else {
                                        D_RE(ans) = C * x;
                                        D_IM(ans) = S * x;
                                }
                        } else {
                                t = __k_cexp(x, &n);
                                                /* return exp(x)=t*2**n */
                                D_RE(ans) = scalbn(C * t, n - 1);
                                D_IM(ans) = scalbn(S * t, n - 1);
                        }
                } else {
                        t = exp(x) * 0.5;
                        D_RE(ans) = C * t;
                        D_IM(ans) = S * t;
                }
        } else {
                if ((ix | lx) == 0) {   /* x = 0, return (C,0) */
                        D_RE(ans) = C;
                        D_IM(ans) = 0.0;
```

```
127                 } else {
128                         D_RE(ans) = C * cosh(x);
129                         D_IM(ans) = S * sinh(x);
130                 }
131         }
132         if ((hx ^ hy) < 0)
133                 D_IM(ans) = -D_IM(ans);
134         return (ans);
135 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    2523 Tue Nov  4 19:01:12 2014
new/usr/src/lib/libm/common/complex/ccoshf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29 #pragma weak __ccoshf = ccoshf
  29 #pragma weak ccoshf = __ccoshf

  31 #include "libm.h"
  32 #include "complex_wrapper.h"

  34 #if defined(__i386) && !defined(__amd64)
  35 extern int __swapRP(int);
  36 #endif

  38 static const float zero = 0.0F, half = 0.5F;

  40 fcomplex
  41 ccoshf(fcomplex z) {
  42         float           t, x, y, S, C;
  43         double          w;
  44         int             hx, ix, hy, iy, n;
  45         fcomplex        ans;

  47         x = F_RE(z);
  48         y = F_IM(z);
  49         hx = THE_WORD(x);
  50         ix = hx & 0x7fffffff;
  51         hy = THE_WORD(y);
  52         iy = hy & 0x7fffffff;
  53         x = fabsf(x);
  54         y = fabsf(y);

  56         sincosf(y, &S, &C);
  57         if (ix >= 0x41600000) { /* |x| > 14 = prec/2 (14,28,34,60) */
  58                 if (ix >= 0x42B171AA) { /* |x| > 88.722... ~ log(2**128) */
  59                         if (ix >= 0x7f800000) { /* |x| is inf or NaN */
  60                                 if (iy == 0) {
```



```
  61                                         F_RE(ans) = x;
  62                                         F_IM(ans) = y;
  63                                 } else if (iy >= 0x7f800000) {
  64                                         F_RE(ans) = x;
  65                                         F_IM(ans) = x - y;
  66                                 } else {
  67                                         F_RE(ans) = C * x;
  68                                         F_IM(ans) = S * x;
  69                                 }
  70                         } else {
  71 #if defined(__i386) && !defined(__amd64)
  72                                 int     rp = __swapRP(fp_extended);
  73 #endif
  74                                 /* return (C, S) * exp(x) / 2 */
  75                                 w = __k_cexp((double)x, &n);
  76                                 F_RE(ans) = (float)scalbn(C * w, n - 1);
  77                                 F_IM(ans) = (float)scalbn(S * w, n - 1);
  78 #if defined(__i386) && !defined(__amd64)
  79                                 if (rp != fp_extended)
  80                                         (void) __swapRP(rp);
  81 #endif
  82                         }
  83                 } else {
  84                         t = expf(x) * half;
  85                         F_RE(ans) = C * t;
  86                         F_IM(ans) = S * t;
  87                 }
  88         } else {
  89                 if (ix == 0) {  /* x = 0, return (C,0) */
  90                         F_RE(ans) = C;
  91                         F_IM(ans) = zero;
  92                 } else {
  93                         F_RE(ans) = C * coshf(x);
  94                         F_IM(ans) = S * sinhf(x);
  95                 }
  96         }
  97         if ((hx ^ hy) < 0)
  98                 F_IM(ans) = -F_IM(ans);
  99         return (ans);
 100 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2363 Tue Nov  4 19:01:12 2014
new/usr/src/lib/libm/common/complex/ccoshl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ccoshl = ccoshl
#pragma weak ccoshl = __ccoshl

#include "libm.h"       /* coshl/expl/fabsl/scalbnl/sincosl/sinhl/__k_cexpl */
#include "complex_wrapper.h"

/* INDENT OFF */
static const long double zero = 0.0L, half = 0.5L;
/* INDENT ON */

ldcomplex
ccoshl(ldcomplex z) {
        long double t, x, y, S, C;
        int hx, ix, hy, iy, n;
        ldcomplex ans;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_XWORD(y);
        iy = hy & 0x7fffffff;
        x = fabsl(x);
        y = fabsl(y);

        (void) sincosl(y, &S, &C);
        if (ix >= 0x4004e000) { /* |x| > 60 = prec/2 (14,28,34,60) */
                if (ix >= 0x400C62E4) { /* |x| > 11356.52... ~ log(2**16384) */
                        if (ix >= 0x7fff0000) { /* |x| is inf or NaN */
                                if (y == zero) {
                                        LD_RE(ans) = x;
                                        LD_IM(ans) = y;
```



```
                                } else if (iy >= 0x7fff0000) {
                                        LD_RE(ans) = x;
                                        LD_IM(ans) = x - y;
                                } else {
                                        LD_RE(ans) = C * x;
                                        LD_IM(ans) = S * x;
                                }
                        } else {
                                t = __k_cexpl(x, &n);
                                                /* return exp(x)=t*2**n */
                                LD_RE(ans) = scalbnl(C * t, n - 1);
                                LD_IM(ans) = scalbnl(S * t, n - 1);
                        }
                } else {
                        t = expl(x) * half;
                        LD_RE(ans) = C * t;
                        LD_IM(ans) = S * t;
                }
        } else {
                if (x == zero) {        /* x = 0, return (C,0) */
                        LD_RE(ans) = C;
                        LD_IM(ans) = zero;
                } else {
                        LD_RE(ans) = C * coshl(x);
                        LD_IM(ans) = S * sinhl(x);
                }
        }
        if ((hx ^ hy) < 0)
                LD_IM(ans) = -LD_IM(ans);
        return (ans);
}
```

*_____unchanged_portion_omitted_*

****
1210 Tue Nov 4 19:01:13 2014
new/usr/src/lib/libm/common/complex/ccosl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ccosl = ccosl
#pragma weak ccosl = __ccosl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
ccosl(ldcomplex z) {
        long double x, y;

        x = LD_RE(z);
        y = LD_IM(z);
        LD_RE(z) = y;
        LD_IM(z) = -x;
        return (ccoshl(z));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    3080 Tue Nov  4 19:01:13 2014
new/usr/src/lib/libm/common/complex/cexp.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cexp = cexp
#pragma weak cexp = __cexp

/* INDENT OFF */
/*
 * dcomplex cexp(dcomplex z);
 *
 *  x+iy    x
 * e     = e  (cos(y)+i*sin(y))
 *
 * Over/underflow issue
 * --------------------
 * exp(x) may be huge but cos(y) or sin(y) may be tiny. So we use
 * function __k_cexp(x,&n) to return exp(x) = __k_cexp(x,&n)*2**n.
 * Thus if exp(x+iy) = A + Bi and t = __k_cexp(x,&n), then
 *         A = t*cos(y)*2**n,   B = t*sin(y)*2**n
 *
 * Purge off all exceptional arguments:
 *      (x,0) --> (exp(x),0)         for all x, include inf and NaN
 *      (+inf, y) --> (+inf, NaN)    for inf, nan
 *      (-inf, y) --> (+-0, +-0)     for y = inf, nan
 *      (x,+-inf/NaN) --> (NaN,NaN)  for finite x
 * For all other cases, return
 *      (x,y) --> exp(x)*cos(y)+i*exp(x)*sin(y))
 *
 * Algorithm for out of range x and finite y
 *      1. compute exp(x) in factor form (t=__k_cexp(x,&n))*2**n
 *      2. compute sincos(y,&s,&c)
 *      3. compute t*s+i*(t*c), then scale back to 2**n and return.
 */
/* INDENT ON */
```



```
#include "libm.h"               /* exp/scalbn/sincos/__k_cexp */
#include "complex_wrapper.h"

static const double zero = 0.0;

dcomplex
cexp(dcomplex z) {
        dcomplex ans;
        double x, y, t, c, s;
        int n, ix, iy, hx, hy, lx, ly;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        if ((iy | ly) == 0) {   /* y = 0 */
                D_RE(ans) = exp(x);
                D_IM(ans) = y;
        } else if (ISINF(ix, lx)) {     /* x is +-inf */
                if (hx < 0) {
                        if (iy >= 0x7ff00000) {
                                D_RE(ans) = zero;
                                D_IM(ans) = zero;
                        } else {
                                sincos(y, &s, &c);
                                D_RE(ans) = zero * c;
                                D_IM(ans) = zero * s;
                        }
                } else {
                        if (iy >= 0x7ff00000) {
                                D_RE(ans) = x;
                                D_IM(ans) = y - y;
                        } else {
                                (void) sincos(y, &s, &c);
                                D_RE(ans) = x * c;
                                D_IM(ans) = x * s;
                        }
                }
        } else {
                (void) sincos(y, &s, &c);
                if (ix >= 0x40862E42) { /* |x| > 709.78... ~ log(2**1024) */
                        t = __k_cexp(x, &n);
                        D_RE(ans) = scalbn(t * c, n);
                        D_IM(ans) = scalbn(t * s, n);
                } else {
                        t = exp(x);
                        D_RE(ans) = t * c;
                        D_IM(ans) = t * s;
                }
        }
        return (ans);
}
```

```
______unchanged_portion_omitted_
```

```
**********************************************************
    2356 Tue Nov  4 19:01:13 2014
new/usr/src/lib/libm/common/complex/cexpf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cexpf = cexpf
#pragma weak cexpf = __cexpf

#include "libm.h"
#include "complex_wrapper.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const float zero = 0.0F;

fcomplex
cexpf(fcomplex z) {
        fcomplex        ans;
        float           x, y, c, s;
        double          t;
        int             n, ix, iy, hx, hy;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        if (iy == 0) {          /* y = 0 */
                F_RE(ans) = expf(x);
                F_IM(ans) = y;
        } else if (ix == 0x7f800000) {  /* x is +-inf */
                if (hx < 0) {
                        if (iy >= 0x7f800000) {
                                F_RE(ans) = zero;
                                F_IM(ans) = zero;
```



```
                        } else {
                                sincosf(y, &s, &c);
                                F_RE(ans) = zero * c;
                                F_IM(ans) = zero * s;
                        }
                } else {
                        if (iy >= 0x7f800000) {
                                F_RE(ans) = x;
                                F_IM(ans) = y - y;
                        } else {
                                sincosf(y, &s, &c);
                                F_RE(ans) = x * c;
                                F_IM(ans) = x * s;
                        }
                }
        } else {
                sincosf(y, &s, &c);
                if (ix >= 0x42B171AA) { /* |x| > 88.722... ~ log(2**128) */
#if defined(__i386) && !defined(__amd64)
                        int     rp = __swapRP(fp_extended);
#endif
                        t = __k_cexp(x, &n);
                        F_RE(ans) = (float)scalbn(t * (double)c, n);
                        F_IM(ans) = (float)scalbn(t * (double)s, n);
#if defined(__i386) && !defined(__amd64)
                        if (rp != fp_extended)
                                (void) __swapRP(rp);
#endif
                } else {
                        t = expf(x);
                        F_RE(ans) = t * c;
                        F_IM(ans) = t * s;
                }
        }
        return (ans);
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    2248 Tue Nov  4 19:01:14 2014
new/usr/src/lib/libm/common/complex/cexpl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cexpl = cexpl
#pragma weak cexpl = __cexpl

#include "libm.h"               /* expl/isinfl/iszerol/scalbnl/sincosl */
#include "complex_wrapper.h"

extern int isinfl(long double);
extern int iszerol(long double);

/* INDENT OFF */
static const long double zero = 0.0L;
/* INDENT ON */

ldcomplex
cexpl(ldcomplex z) {
        ldcomplex ans;
        long double x, y, t, c, s;
        int n, ix, iy, hx, hy;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        hy = HI_XWORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        if (iszerol(y)) {       /* y = 0 */
                LD_RE(ans) = expl(x);
                LD_IM(ans) = y;
        } else if (isinfl(x)) { /* x is +-inf */
                if (hx < 0) {
                        if (iy >= 0x7fff0000) {
                                LD_RE(ans) = zero;
```



```
                                LD_IM(ans) = zero;
                        } else {
                                sincosl(y, &s, &c);
                                LD_RE(ans) = zero * c;
                                LD_IM(ans) = zero * s;
                        }
                } else {
                        if (iy >= 0x7fff0000) {
                                LD_RE(ans) = x;
                                LD_IM(ans) = y - y;
                        } else {
                                (void) sincosl(y, &s, &c);
                                LD_RE(ans) = x * c;
                                LD_IM(ans) = x * s;
                        }
                }
        } else {
                (void) sincosl(y, &s, &c);
                if (ix >= 0x400C62E4) { /* |x| > 11356.52... ~ log(2**16384) */
                        t = __k_cexpl(x, &n);
                        LD_RE(ans) = scalbnl(t * c, n);
                        LD_IM(ans) = scalbnl(t * s, n);
                } else {
                        t = expl(x);
                        LD_RE(ans) = t * c;
                        LD_IM(ans) = t * s;
                }
        }
        return (ans);
}
```

*_____unchanged_portion_omitted_*

****
1123 Tue Nov 4 19:01:14 2014
new/usr/src/lib/libm/common/complex/cimag.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cimag = cimag
#pragma weak cimag = __cimag

#include "libm.h"
#include "complex_wrapper.h"

double
cimag(dcomplex z) {
        return (D_IM(z));
}
```

_____***unchanged_portion_omitted_***

****
1125 Tue Nov 4 19:01:15 2014
new/usr/src/lib/libm/common/complex/cimagf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cimagf = cimagf
#pragma weak cimagf = __cimagf

#include "libm.h"
#include "complex_wrapper.h"

float
cimagf(fcomplex z) {
        return (F_IM(z));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1133 Tue Nov  4 19:01:15 2014
new/usr/src/lib/libm/common/complex/cimagl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cimagl = cimagl
#pragma weak cimagl = __cimagl

#include "libm.h"
#include "complex_wrapper.h"

long double
cimagl(ldcomplex z) {
        return (LD_IM(z));
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    3776 Tue Nov  4 19:01:15 2014
new/usr/src/lib/libm/common/complex/clog.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak clog = __clog

/* INDENT OFF */
/*
 * dcomplex clog(dcomplex z);
 *
 *                      _________
 *                    / 2    2          -1   y
 * log(x+iy) = log(\/ x  + y    ) + i tan   (---)
 *                                         x
 *
 *              1       2    2         -1   y
 *           = --- log(x  + y ) + i tan   (---)
 *              2                       x
 *
 * Note that the arctangent ranges from -PI to +PI, thus the imaginary
 * part of clog is atan2(y,x).
 *
 * EXCEPTION CASES (conform to ISO/IEC 9899:1999(E)):
 *    clog(-0 + i 0   ) =  -inf + i pi
 *    clog( 0 + i 0   ) =  -inf + i 0
 *    clog( x + i inf ) =  -inf + i pi/2, for finite x
 *    clog( x + i NaN ) =  NaN  + i NaN with invalid for finite x
 *    clog(-inf + iy   )=  +inf + i pi, for finite positive-signed y
 *    clog(+inf + iy   )=  +inf + i 0 , for finite positive-signed y
 *    clog(-inf + i inf)=  inf  + i 3pi/4
 *    clog(+inf + i inf)=  inf  + i pi/4
 *    clog(+-inf+ i NaN)=  inf  + i NaN
 *    clog(NaN  + i y  )=  NaN  + i NaN for finite y
 *    clog(NaN  + i inf)=  inf  + i NaN
 *    clog(NaN  + i NaN)=  NaN  + i NaN
 */
/* INDENT ON */
```



```
#include "libm_synonyms.h"
#include <math.h>               /* atan2/fabs/log/log1p */
#include "complex_wrapper.h"
#include "libm_protos.h"        /* __k_clog_r */


static const double half = 0.5, one = 1.0;

dcomplex
__clog(dcomplex z) {
clog(dcomplex z) {
        dcomplex        ans;
        double          x, y, t, ax, ay, w;
        int             n, ix, iy, hx, hy;
        unsigned        lx, ly;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        ay = fabs(y);
        ax = fabs(x);
        D_IM(ans) = carg(z);
        if (ix < iy || (ix == iy && lx < ly)) {
                /* swap x and y to force ax >= ay */
                t = ax;
                ax = ay;
                ay = t;
                n = ix, ix = iy;
                iy = n;
                n = lx, lx = ly;
                ly = n;
        }
        n = (ix - iy) >> 20;
        if (ix >= 0x7ff00000) { /* x or y is Inf or NaN */
                if (ISINF(ix, lx))
                        D_RE(ans) = ax;
                else if (ISINF(iy, ly))
                        D_RE(ans) = ay;
                else
                        D_RE(ans) = ax * ay;
        } else if ((iy | ly) == 0) {
                D_RE(ans) = ((ix | lx) == 0)? -one / ax : log(ax);
        } else if ((((0x3fffffff - ix) ^ (ix - 0x3fe00000)) >= 0) {
                /* 0.5 <= x < 2 */
                if (ix >= 0x3ff00000) {
                        if (((ix - 0x3ff00000) | lx) == 0)
                                D_RE(ans) = half * log1p(ay * ay);
                        else if (n >= 60)
                                D_RE(ans) = log(ax);
                        else
                                D_RE(ans) = half * (log1p(ay * ay + (ax -
                                    one) * (ax + one)));
                } else if (n >= 60) {
                        D_RE(ans) = log(ax);
                } else {
                        D_RE(ans) = __k_clog_r(ax, ay, &w);
                }
        } else if (n >= 30) {
                D_RE(ans) = log(ax);
        } else if (ix < 0x5f300000 && iy >= 0x20b00000) {
```

```
 126                 /* 2**-500< y < x < 2**500 */
 127                 D_RE(ans) = half * log(ax * ax + ay * ay);
 128         } else {
 129                 t = ay / ax;
 130                 D_RE(ans) = log(ax) + half * log1p(t * t);
 131         }
 132         return (ans);
 133 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2129 Tue Nov  4 19:01:16 2014
new/usr/src/lib/libm/common/complex/clogf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __clogf = clogf
#pragma weak clogf = __clogf

#include "libm.h"
#include "complex_wrapper.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

fcomplex
clogf(fcomplex z) {
        fcomplex        ans;
        float           x, y, ax, ay;
        double          dx, dy;
        int             ix, iy, hx, hy;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        ay = fabsf(y);
        ax = fabsf(x);
        F_IM(ans) = atan2f(y, x);
        if (ix >= 0x7f800000 || iy >= 0x7f800000) {
                /* x or y is Inf or NaN */
                if (iy == 0x7f800000)
                        F_RE(ans) = ay;
                else if (ix == 0x7f800000)
                        F_RE(ans) = ax;
                else
```



```
                        F_RE(ans) = ax + ay;
        } else {
#if defined(__i386) && !defined(__amd64)
                int     rp = __swapRP(fp_extended);
#endif
                dx = (double)ax;
                dy = (double)ay;
                if (ix == 0x3f800000)
                        F_RE(ans) = (float)(0.5 * log1p(dy * dy));
                else if (iy == 0x3f800000)
                        F_RE(ans) = (float)(0.5 * log1p(dx * dx));
                else if ((ix | iy) == 0)
                        F_RE(ans) = -1.0f / ax;
                else
                        F_RE(ans) = (float)(0.5 * log(dx * dx + dy * dy));
#if defined(__i386) && !defined(__amd64)
                if (rp != fp_extended)
                        (void) __swapRP(rp);
#endif
        }
        return (ans);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2709 Tue Nov  4 19:01:16 2014
new/usr/src/lib/libm/common/complex/clogl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
  1 /*
  2  * CDDL HEADER START
  3  *
  4  * The contents of this file are subject to the terms of the
  5  * Common Development and Distribution License (the "License").
  6  * You may not use this file except in compliance with the License.
  7  *
  8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9  * or http://www.opensolaris.org/os/licensing.
 10  * See the License for the specific language governing permissions
 11  * and limitations under the License.
 12  *
 13  * When distributing Covered Code, include this CDDL HEADER in each
 14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15  * If applicable, add the following below this CDDL HEADER, with the
 16  * fields enclosed by brackets "[]" replaced with your own identifying
 17  * information: Portions Copyright [yyyy] [name of copyright owner]
 18  *
 19  * CDDL HEADER END
 20  */

 22 /*
 23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 24  */
 25 /*
 26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 27  * Use is subject to license terms.
 28  */

 30 #pragma weak __clogl = clogl
 30 #pragma weak clogl = __clogl

 32 #include "libm.h"       /* atan2l/fabsl/isinfl/log1pl/logl/__k_clog_rl */
 33 #include "complex_wrapper.h"
 34 #include "longdouble.h"

 36 #if defined(__sparc)
 37 #define SIGP7   120
 38 #define HSIGP7  60
 39 #elif defined(__x86)
 40 #define SIGP7   70
 41 #define HSIGP7  35
 42 #endif

 44 /* INDENT OFF */
 45 static const long double zero = 0.0L, half = 0.5L, one = 1.0L;
 46 /* INDENT ON */

 48 ldcomplex
 49 clogl(ldcomplex z) {
 50         ldcomplex ans;
 51         long double x, y, t, ax, ay;
 52         int n, ix, iy, hx, hy;

 54         x = LD_RE(z);
 55         y = LD_IM(z);
 56         hx = HI_XWORD(x);
 57         hy = HI_XWORD(y);
 58         ix = hx & 0x7fffffff;
 59         iy = hy & 0x7fffffff;
 60         ay = fabsl(y);
```



```
 61         ax = fabsl(x);
 62         LD_IM(ans) = atan2l(y, x);
 63         if (ix < iy || (ix == iy && ix < 0x7fff0000 && ax < ay)) {
 64                         /* swap x and y to force ax>=ay */
 65                 t = ax;
 66                 ax = ay;
 67                 ay = t;
 68                 n = ix, ix = iy;
 69                 iy = n;
 70         }
 71         n = (ix - iy) >> 16;
 72         if (ix >= 0x7fff0000) { /* x or y is Inf or NaN */
 73                 if (isinfl(ax))
 74                         LD_RE(ans) = ax;
 75                 else if (isinfl(ay))
 76                         LD_RE(ans) = ay;
 77                 else
 78                         LD_RE(ans) = ax + ay;
 79         } else if (ay == zero)
 80                 LD_RE(ans) = logl(ax);
 81         else if (((0x3fffffff - ix) ^ (ix - 0x3ffe0000)) >= 0) {
 82                                                 /* 0.5 <= x < 2 */
 83                 if (ix >= 0x3fff0000) {
 84                         if (ax == one)
 85                                 LD_RE(ans) = half * log1pl(ay * ay);
 86                         else if (n >= SIGP7)
 87                                 LD_RE(ans) = logl(ax);
 88                         else
 89                                 LD_RE(ans) = half * (log1pl(ay * ay + (ax -
 90                                         one) * (ax + one)));
 91                 } else if (n >= SIGP7)
 92                         LD_RE(ans) = logl(ax);
 93                 else
 94                         LD_RE(ans) = __k_clog_rl(x, y, &t);
 95         } else if (n >= HSIGP7)
 96                 LD_RE(ans) = logl(ax);
 97         else if (ix < 0x5f3f0000 && iy >= 0x20bf0000)
 98                 /* 2**-8000 < y < x < 2**8000 */
 99                 LD_RE(ans) = half * logl(ax * ax + ay * ay);
100         else {
101                 t = ay / ax;
102                 LD_RE(ans) = logl(ax) + half * log1pl(t * t);
103         }
104         return (ans);
105 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1137 Tue Nov  4 19:01:17 2014
new/usr/src/lib/libm/common/complex/conj.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __conj = conj
#pragma weak conj = __conj

#include "libm.h"
#include "complex_wrapper.h"

dcomplex
conj(dcomplex z) {
        D_IM(z) = -D_IM(z);
        return (z);
}
```

_____***unchanged_portion_omitted_***

****
1140 Tue Nov 4 19:01:17 2014
new/usr/src/lib/libm/common/complex/conjf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __conjf = conjf
#pragma weak conjf = __conjf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
conjf(fcomplex z) {
        F_IM(z) = -F_IM(z);
        return (z);
}
```

_____***unchanged_portion_omitted_***

****
1144 Tue Nov 4 19:01:17 2014
new/usr/src/lib/libm/common/complex/conjl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __conjl = conjl
#pragma weak conjl = __conjl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
conjl(ldcomplex z) {
        LD_IM(z) = -LD_IM(z);
        return (z);
}
```

*_____unchanged_portion_omitted_*

```
************************************************************
    9501 Tue Nov  4 19:01:18 2014
new/usr/src/lib/libm/common/complex/cpow.c
5261 libm should stop using synonyms.h
************************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cpow = cpow
#pragma weak cpow = __cpow

/* INDENT OFF */
/*
 * dcomplex cpow(dcomplex z);
 *
 * z**w analytically equivalent to
 *
 * cpow(z,w) = cexp(w clog(z))
 *
 * Let z = x+iy, w = u+iv.
 * Since
 *                          _________
 *                         / 2   2            -1   y
 *      log(x+iy) = log(\/ x  + y    ) + i tan   (---)
 *                                                  x
 *
 *                  1       2    2         -1   y
 *               = --- log(x  + y ) + i tan   (---)
 *                  2                           x
 *                         u       2    2         -1  y
 * (u+iv)* log(x+iy) =  --- log(x  + y ) - v tan  (---)  +          (1)
 *                         2                          x
 *
 *                              v       2    2         -1  y
 *                      i * [ --- log(x  + y ) + u tan  (---) ]      (2)
 *                              2                          x
 *
 *                    = r + i q
 *
 * Therefore,
 *      w     r+iq    r
 *     z  =  e     = e  (cos(q)+i*sin(q))
 *                                  _______
 *                                 / 2   2
 *       r                       \/ x + y     -v*atan2(y,x)
 * Here e  can be expressed as:  u          * e
 *
 * Special cases (in the order of appearance):
 *      1.  (anything) ** 0  is 1
 *      2.  (anything) ** 1  is itself
 *      3.  When v = 0, y = 0:
 *            If x is finite and negative, and u is finite, then
 *               x ** u = exp(u*pi i) * pow(|x|, u);
 *            otherwise,
 *               x ** u = pow(x, u);
 *      4.  When v = 0, x = 0 or |x| = |y| or x is inf or y is inf:
 *               (x + y i) ** u = r * exp(q i)
 *          where
 *               r = hypot(x,y) ** u
 *               q = u * atan2pi(y, x)
 *
 *      5.  otherwise, z**w is NAN if any x, y, u, v is a Nan or inf
 *
 *      Note: many results of special cases are obtained in terms of
 *      polar coordinate. In the conversion from polar to rectangle:
 *                  r exp(q i) = r * cos(q) + r * sin(q) i,
 *      we regard r * 0 is 0 except when r is a NaN.
 */
/* INDENT ON */

#include "libm.h"       /* atan2/exp/fabs/hypot/log/pow/scalbn */
                        /* atan2pi/exp2/sincos/sincospi/__k_clog_r/__k_atan2 */
#include "complex_wrapper.h"

extern void sincospi(double, double *, double *);

static const double
        huge = 1e300,
        tiny = 1e-300,
        invln2 = 1.44269504088896338700e+00,
        ln2hi = 6.93147180369123816490e-01,   /* 0x3fe62e42, 0xfee00000 */
        ln2lo = 1.90821492927058770002e-10,   /* 0x3dea39ef, 0x35793c76 */
        one = 1.0,
        zero = 0.0;

static const int hiinf = 0x7ff00000;
extern double atan2pi(double, double);

/*
 * Assuming |t[0]| > |t[1]| and |t[2]| > |t[3]|, sum4fp subroutine
 * compute t[0] + t[1] + t[2] + t[3] into two double fp numbers.
 */
static double
sum4fp(double ta[], double *w) {
        double t1, t2, t3, t4, w1, w2, t;
        t1 = ta[0]; t2 = ta[1]; t3 = ta[2]; t4 = ta[3];
        /*
         * Rearrange ti so that |t1| >= |t2| >= |t3| >= |t4|
         */
        if (fabs(t4) > fabs(t1)) {
                t = t1; t1 = t3; t3 = t;
                t = t2; t2 = t4; t4 = t;
        } else if (fabs(t3) > fabs(t1)) {
                t = t1; t1 = t3;
                if (fabs(t4) > fabs(t2)) {
                        t3 = t4; t4 = t2; t2 = t;
```

```
 127                 } else {
 128                         t3 = t2; t2 = t;
 129                 }
 130         } else if (fabs(t3) > fabs(t2)) {
 131                 t = t2; t2 = t3;
 132                 if (fabs(t4) > fabs(t2)) {
 133                         t3 = t4; t4 = t;
 134                 } else
 135                         t3 = t;
 136         }
 137         /* summing r = t1 + t2 + t3 + t4 to w1 + w2 */
 138         w1 = t3 + t4;
 139         w2 = t4 - (w1 - t3);
 140         t  = t2 + w1;
 141         w2 += w1 - (t - t2);
 142         w1 = t + w2;
 143         w2 += t - w1;
 144         t  = t1 + w1;
 145         w2 += w1 - (t - t1);
 146         w1 = t + w2;
 147         *w = w2 - (w1 - t);
 148         return (w1);
 149 }
```

unchanged_portion_omitted_

```
**********************************************************
    4751 Tue Nov  4 19:01:18 2014
new/usr/src/lib/libm/common/complex/cpowf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cpowf = cpowf
#pragma weak cpowf = __cpowf

#include "libm.h"
#include "complex_wrapper.h"

extern void sincospi(double, double *, double *);
extern void sincospif(float, float *, float *);
extern double atan2pi(double, double);
extern float atan2pif(float, float);

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const double
        dpi = 3.14159265358979311600E0,  /* Hex 2^ 1 * 1.921FB54442D18 */
        dhalf = 0.5,
        dsqrt2 = 1.41421356237309514547,        /* 3FF6A09E 667F3BCD */
        dinvpi = 0.31830988618379067153776752;

static const float one = 1.0F, zero = 0.0F;

#define hiinf   0x7f800000

fcomplex
cpowf(fcomplex z, fcomplex w) {
        fcomplex        ans;
        float           x, y, u, v, t, c, s;
        double          dx, dy, du, dv, dt, dc, ds, dp, dq, dr;
        int             ix, iy, hx, hy, hv, hu, iu, iv, j;

        x = F_RE(z);
```



```
        y = F_IM(z);
        u = F_RE(w);
        v = F_IM(w);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        hu = THE_WORD(u);
        hv = THE_WORD(v);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        iu = hu & 0x7fffffff;
        iv = hv & 0x7fffffff;

        j = 0;
        if (iv == 0) {          /* z**(real) */
                if (hu == 0x3f800000) { /* (anything) ** 1  is itself */
                        F_RE(ans) = x;
                        F_IM(ans) = y;
                } else if (iu == 0) {   /* (anything) ** 0  is 1 */
                        F_RE(ans) = one;
                        F_IM(ans) = zero;
                } else if (iy == 0) {   /* (real)**(real) */
                        F_IM(ans) = zero;
                        if (hx < 0 && ix < hiinf && iu < hiinf) {
                                /* -x ** u  is exp(i*pi*u)*pow(x,u) */
                                t = powf(-x, u);
                                sincospif(u, &s, &c);
                                F_RE(ans) = (c == zero)? c: c * t;
                                F_IM(ans) = (s == zero)? s: s * t;
                        } else {
                                F_RE(ans) = powf(x, u);
                        }
                } else if (ix == 0 || ix >= hiinf || iy >= hiinf) {
                        if (ix > hiinf || iy > hiinf || iu > hiinf) {
                                F_RE(ans) = F_IM(ans) = x + y + u;
                        } else {
                                v = fabsf(y);
                                if (ix != 0)
                                        v += fabsf(x);
                                t = atan2pif(y, x);
                                sincospif(t * u, &s, &c);
                                F_RE(ans) = (c == zero)? c: c * v;
                                F_IM(ans) = (s == zero)? s: s * v;
                        }
                } else if (ix == iy) {  /* if |x| == |y| */
#if defined(__i386) && !defined(__amd64)
                        int     rp = __swapRP(fp_extended);
#endif
                        dx = (double)x;
                        du = (double)u;
                        dt = (hx >= 0)? 0.25 : 0.75;
                        if (hy < 0)
                                dt = -dt;
                        dr = pow(dsqrt2 * dx, du);
                        sincospi(dt * du, &ds, &dc);
                        F_RE(ans) = (float)(dr * dc);
                        F_IM(ans) = (float)(dr * ds);
#if defined(__i386) && !defined(__amd64)
                        if (rp != fp_extended)
                                (void) __swapRP(rp);
#endif
                } else {
                        j = 1;
                }
                if (j == 0)
                        return (ans);
        }
```

```
127         if (iu >= hiinf || iv >= hiinf || ix >= hiinf || iy >= hiinf) {
128                 /*
129                  * non-zero imaginery part(s) with inf component(s) yields NaN
130                  */
131                 t = fabsf(x) + fabsf(y) + fabsf(u) + fabsf(v);
132                 F_RE(ans) = F_IM(ans) = t - t;
133         } else {
134 #if defined(__i386) && !defined(__amd64)
135                 int     rp = __swapRP(fp_extended);
136 #endif
137                 /* INDENT OFF */
138                 /*
139                  * r = u*log(hypot(x,y))-v*atan2(y,x),
140                  * q = u*atan2(y,x)+v*log(hypot(x,y))
141                  * or
142                  * r = u*log(hypot(x,y))-v*pi*atan2pi(y,x),
143                  * q/pi = u*atan2pi(y,x)+v*log(hypot(x,y))/pi
144                  * ans = exp(r)*(cospi(q/pi)  + i sinpi(q/pi))
145                  */
146                 /* INDENT ON */
147                 dx = (double)x;
148                 dy = (double)y;
149                 du = (double)u;
150                 dv = (double)v;
151                 if (ix > 0x3f000000 && ix < 0x40000000) /* .5 < |x| < 2 */
152                         dt = dhalf * log1p((dx - 1.0) * (dx + 1.0) + dy * dy);
153                 else if (iy > 0x3f000000 && iy < 0x40000000) /* .5 < |y| < 2 */
154                         dt = dhalf * log1p((dy - 1.0) * (dy + 1.0) + dx * dx);
155                 else
156                         dt = dhalf * log(dx * dx + dy * dy);
157                 dp = atan2pi(dy, dx);
158                 if (iv == 0) {  /* dv = 0 */
159                         dr = exp(du * dt);
160                         dq = du * dp;
161                 } else {
162                         dr = exp(du * dt - dv * dp * dpi);
163                         dq = du * dp + dv * dt * dinvpi;
164                 }
165                 sincospi(dq, &ds, &dc);
166                 F_RE(ans) = (float)(dr * dc);
167                 F_IM(ans) = (float)(dr * ds);
168 #if defined(__i386) && !defined(__amd64)
169                 if (rp != fp_extended)
170                         (void) __swapRP(rp);
171 #endif
172         }
173         return (ans);
174 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7657 Tue Nov  4 19:01:19 2014
new/usr/src/lib/libm/common/complex/cpowl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cpowl = cpowl
#pragma weak cpowl = __cpowl

#include "libm.h"                               /* __k_clog_rl/__k_atan2l */
/* atan2l/atan2pil/exp2l/expl/fabsl/hypotl/isinfl/logl/powl/sincosl/sincospil */
#include "complex_wrapper.h"
#include "longdouble.h"

#if defined(__sparc)
#define HALF(x)  ((int *) &x)[3] = 0; ((int *) &x)[2] &= 0xfe000000
#define LAST(x)  ((int *) &x)[3]
#elif defined(__x86)
#define HALF(x)  ((int *) &x)[0] = 0
#define LAST(x)  ((int *) &x)[0]
#endif

/* INDENT OFF */
static const int hiinf = 0x7fff0000;
static const long double
        tiny = 1.0e-4000L,
        huge = 1.0e4000L,
#if defined(__x86)
                /* 43 significant bits, 21 trailing zeros */
        ln2hil = 0.693147180559890330187045037746429443359375L,
        ln2lol = 5.4979230187083711747124716125134360255254120668e-14L,
#else   /* sparc */
                /* 0x3FF962E4 2FEFA39E F35793C7 00000000 */
        ln2hil = 0.693147180559945309417231592858066493070671489074L,
        ln2lol = 5.2860011007500482864528623582064673010680244656615 3e-25L,
#endif
        invln2  = 1.44269504088896340735992468100189213742 7e+0000L,
        one = 1.0L,
```



```
        zero = 0.0L;
/* INDENT ON */

/*
 * Assuming |t[0]| > |t[1]| and |t[2]| > |t[3]|, sum4fpl subroutine
 * compute t[0] + t[1] + t[2] + t[3] into two long double fp numbers.
 */
static long double sum4fpl(long double ta[], long double *w)
{
        long double t1, t2, t3, t4, w1, w2, t;
        t1 = ta[0]; t2 = ta[1]; t3 = ta[2]; t4 = ta[3];
        /*
         * Rearrange ti so that |t1| >= |t2| >= |t3| >= |t4|
         */
        if (fabsl(t4) > fabsl(t1)) {
                t = t1; t1 = t3; t3 = t;
                t = t2; t2 = t4; t4 = t;
        } else if (fabsl(t3) > fabsl(t1)) {
                t = t1; t1 = t3;
                if (fabsl(t4) > fabsl(t2)) {
                        t3 = t4; t4 = t2; t2 = t;
                } else {
                        t3 = t2; t2 = t;
                }
        } else if (fabsl(t3) > fabsl(t2)) {
                t = t2; t2 = t3;
                if (fabsl(t4) > fabsl(t2)) {
                        t3 = t4; t4 = t;
                } else
                        t3 = t;
        }
        /* summing r = t1 + t2 + t3 + t4 to w1 + w2 */
        w1 = t3 + t4;
        w2 = t4 - (w1 - t3);
        t  = t2 + w1;
        w2 += w1 - (t - t2);
        w1 = t + w2;
        w2 += t - w1;
        t  = t1 + w1;
        w2 += w1 - (t - t1);
        w1 = t + w2;
        *w = w2 - (w1 - t);
        return (w1);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1721 Tue Nov  4 19:01:19 2014
new/usr/src/lib/libm/common/complex/cproj.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cproj = cproj
#pragma weak cproj = __cproj

/* INDENT OFF */
/*
 * dcomplex cproj(dcomplex z);
 *
 * If one of the component of z = (x,y) is an inf, then
 *      cproj(z) = (+inf, copysign(0,y));
 * otherwise,
 *      cproj(z) = z
 */
/* INDENT ON */

#include "libm.h"                       /* fabs */
#include "complex_wrapper.h"

static const double zero = 0.0;

dcomplex
cproj(dcomplex z) {
        double x, y;
        int ix, iy, hx, hy, lx, ly;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
```



```
        if (ISINF(iy, ly)) {
                D_RE(z) = fabs(y);
                D_IM(z) = hy >= 0 ? zero : -zero;
        } else if (ISINF(ix, lx)) {
                D_RE(z) = fabs(x);
                D_IM(z) = hy >= 0 ? zero : -zero;
        }
        return (z);
}
```

*unchanged_portion_omitted_*

****
1511 Tue Nov 4 19:01:19 2014
new/usr/src/lib/libm/common/complex/cprojf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cprojf = cprojf
#pragma weak cprojf = __cprojf

#include "libm.h"
#include "complex_wrapper.h"

/* INDENT OFF */
static const float zero = 0.0F;
/* INDENT ON */

fcomplex
cprojf(fcomplex z) {
        float x, y;
        int ix, iy, hx, hy;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        if (iy == 0x7f800000) {
                F_RE(z) = fabsf(y);
                F_IM(z) = hy >= 0 ? zero : -zero;
        } else if (ix == 0x7f800000) {
                F_RE(z) = fabsf(x);
                F_IM(z) = hy >= 0 ? zero : -zero;
        }
        return (z);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1542 Tue Nov  4 19:01:20 2014
new/usr/src/lib/libm/common/complex/cprojl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __cprojl = cprojl
#pragma weak cprojl = __cprojl

#include "libm.h"               /* fabsl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double zero = 0.0L;
/* INDENT ON */

ldcomplex
cprojl(ldcomplex z) {
        long double x, y;
        int hy;

        x = LD_RE(z);
        y = LD_IM(z);
#if defined(__x86)
        hy = ((int *) &y)[2] << 16;
#else
        hy = ((int *) &y)[0];
#endif
        if (isinfl(y)) {
                LD_RE(z) = fabsl(y);
                LD_IM(z) = hy >= 0 ? zero : -zero;
        } else if (isinfl(x)) {
                LD_RE(z) = fabsl(x);
                LD_IM(z) = hy >= 0 ? zero : -zero;
        }
        return (z);
}
```

****
1123 Tue Nov 4 19:01:20 2014
new/usr/src/lib/libm/common/complex/creal.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __creal = creal
#pragma weak creal = __creal

#include "libm.h"
#include "complex_wrapper.h"

double
creal(dcomplex z) {
        return (D_RE(z));
}
```

_____***unchanged_portion_omitted_***

****
1125 Tue Nov 4 19:01:21 2014
new/usr/src/lib/libm/common/complex/crealf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __crealf = crealf
#pragma weak crealf = __crealf

#include "libm.h"
#include "complex_wrapper.h"

float
crealf(fcomplex z) {
        return (F_RE(z));
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1133 Tue Nov  4 19:01:21 2014
new/usr/src/lib/libm/common/complex/creall.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __creall = creall
#pragma weak creall = __creall

#include "libm.h"
#include "complex_wrapper.h"

long double
creall(ldcomplex z) {
        return (LD_RE(z));
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1558 Tue Nov  4 19:01:22 2014
new/usr/src/lib/libm/common/complex/csin.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csin = csin
#pragma weak csin = __csin

/* INDENT OFF */
/*
 * dcomplex csin(dcomplex z);
 *
 * If z = x+iy, then since csin(iz) = i*csinh(z),  we have
 *
 * csin(z)      = csin((-1)*(-z)) = csin(i*i*(-z))
 *              = i*csinh(i*(-z)) = i*csinh(i*(-x-yi))
 *              = i*csinh(y-ix)
 *              = -Im(csinh(y-ix))+i*Re(csinh(y-ix))
 */
/* INDENT ON */

#include "libm.h"
#include "complex_wrapper.h"

dcomplex
csin(dcomplex z) {
        double x, y;
        dcomplex ans, ct;

        x = D_RE(z);
        y = D_IM(z);
        D_RE(z) = y;
        D_IM(z) = -x;
        ct = csinh(z);
        D_RE(ans) = -D_IM(ct);
        D_IM(ans) = D_RE(ct);
        return (ans);
```



```
}
______unchanged_portion_omitted_
```

****
1275 Tue Nov 4 19:01:22 2014
new/usr/src/lib/libm/common/complex/csinf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csinf = csinf
#pragma weak csinf = __csinf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
csinf(fcomplex z) {
        float x, y;
        fcomplex ans, ct;

        x = F_RE(z);
        y = F_IM(z);
        F_RE(z) = y;
        F_IM(z) = -x;
        ct = csinhf(z);
        F_RE(ans) = -F_IM(ct);
        F_IM(ans) = F_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3923 Tue Nov  4 19:01:22 2014
new/usr/src/lib/libm/common/complex/csinh.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csinh = csinh
#pragma weak csinh = __csinh

/* INDENT OFF */
/*
 * dcomplex csinh(dcomplex z);
 *
 *             z      -z       x                      -x
 *            e   -  e        e  (cos(y)+i*sin(y)) - e  (cos(-y)+i*sin(-y))
 * sinh z = -------------- =  ---------------------------------------------
 *                 2                                  2
 *                      x    -x               x    -x
 *           cos(y) ( e  - e  )  + i*sin(y) (e  + e   )
 *        = --------------------------------------------
 *                                 2
 *
 *        =  cos(y) sinh(x)  + i sin(y) cosh(x)
 *
 * Implementation Note
 * -------------------
 *
 *             |x|    -|x|   |x|        -2|x|       -2|x|    -P-4
 * Note that  e   +- e    = e   ( 1 +- e     ). If e      < 2     , where
 *
 * P stands for the number of significant bits of the machine precision,
 *                                       |x|
 * then the result will be rounded to e   . Therefore, we have
 *
 *                   z
 *                  e
 *     sinh z = -----  if |x| >= (P/2 + 2)*ln2
 *                  2
 *
 * EXCEPTION (conform to ISO/IEC 9899:1999(E)):
 *      csinh(0,0)=(0,0)
 *      csinh(0,inf)=(+-0,NaN)
 *      csinh(0,NaN)=(+-0,NaN)
 *      csinh(x,inf) = (NaN,NaN) for finite positive x
 *      csinh(x,NaN) = (NaN,NaN) for finite non-zero x
 *      csinh(inf,0) = (inf, 0)
 *      csinh(inf,y) = (inf*cos(y),inf*sin(y)) for positive finite y
 *      csinh(inf,inf) = (+-inf,NaN)
 *      csinh(inf,NaN) = (+-inf,NaN)
 *      csinh(NaN,0) = (NaN,0)
 *      csinh(NaN,y) = (NaN,NaN) for non-zero y
 *      csinh(NaN,NaN) = (NaN,NaN)
 */
/* INDENT ON */

#include "libm.h"               /* cosh/exp/fabs/scalbn/sinh/sincos/__k_cexp */
#include "complex_wrapper.h"

dcomplex
csinh(dcomplex z) {
        double t, x, y, S, C;
        int hx, ix, lx, hy, iy, ly, n;
        dcomplex ans;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        iy = hy & 0x7fffffff;
        x = fabs(x);
        y = fabs(y);

        (void) sincos(y, &S, &C);
        if (ix >= 0x403c0000) { /* |x| > 28 = prec/2 (14,28,34,60) */
                if (ix >= 0x40862E42) { /* |x| > 709.78... ~ log(2**1024) */
                        if (ix >= 0x7ff00000) { /* |x| is inf or NaN */
                                if ((iy | ly) == 0) {
                                        D_RE(ans) = x;
                                        D_IM(ans) = y;
                                } else if (iy >= 0x7ff00000) {
                                        D_RE(ans) = x;
                                        D_IM(ans) = x - y;
                                } else {
                                        D_RE(ans) = C * x;
                                        D_IM(ans) = S * x;
                                }
                        } else {
                                /* return exp(x)=t*2**n */
                                t = __k_cexp(x, &n);
                                D_RE(ans) = scalbn(C * t, n - 1);
                                D_IM(ans) = scalbn(S * t, n - 1);
                        }
                } else {
                        t = exp(x) * 0.5;
                        D_RE(ans) = C * t;
                        D_IM(ans) = S * t;
                }
        } else {
                if ((ix | lx) == 0) {   /* x = 0, return (0,S) */
                        D_RE(ans) = 0.0;
                        D_IM(ans) = S;
```

```
 127                 } else {
 128                         D_RE(ans) = C * sinh(x);
 129                         D_IM(ans) = S * cosh(x);
 130                 }
 131         }
 132         if (hx < 0)
 133                 D_RE(ans) = -D_RE(ans);
 134         if (hy < 0)
 135                 D_IM(ans) = -D_IM(ans);
 136         return (ans);
 137 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2552 Tue Nov  4 19:01:23 2014
new/usr/src/lib/libm/common/complex/csinhf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csinhf = csinhf
#pragma weak csinhf = __csinhf

#include "libm.h"
#include "complex_wrapper.h"

#if defined(__i386) && !defined(__amd64)
extern int __swapRP(int);
#endif

static const float zero = 0.0F, half = 0.5F;

fcomplex
csinhf(fcomplex z) {
        float           x, y, S, C;
        double          t;
        int             hx, ix, hy, iy, n;
        fcomplex        ans;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        ix = hx & 0x7fffffff;
        hy = THE_WORD(y);
        iy = hy & 0x7fffffff;
        x = fabsf(x);
        y = fabsf(y);

        sincosf(y, &S, &C);
        if (ix >= 0x41600000) { /* |x| > 14 = prec/2 (14,28,34,60) */
                if (ix >= 0x42B171AA) { /* |x| > 88.722... ~ log(2**128) */
                        if (ix >= 0x7f800000) { /* |x| is inf or NaN */
                                if (iy == 0) {
```



```
                                        F_RE(ans) = x;
                                        F_IM(ans) = y;
                                } else if (iy >= 0x7f800000) {
                                        F_RE(ans) = x;
                                        F_IM(ans) = x - y;
                                } else {
                                        F_RE(ans) = C * x;
                                        F_IM(ans) = S * x;
                                }
                        } else {
#if defined(__i386) && !defined(__amd64)
                                int     rp = __swapRP(fp_extended);
#endif
                                /* return (C, S) * exp(x) / 2 */
                                t = __k_cexp((double)x, &n);
                                F_RE(ans) = (float)scalbn(C * t, n - 1);
                                F_IM(ans) = (float)scalbn(S * t, n - 1);
#if defined(__i386) && !defined(__amd64)
                                if (rp != fp_extended)
                                        (void) __swapRP(rp);
#endif
                        }
                } else {
                        t = expf(x) * half;
                        F_RE(ans) = C * t;
                        F_IM(ans) = S * t;
                }
        } else {
                if (ix == 0) {  /* x = 0, return (0,S) */
                        F_RE(ans) = zero;
                        F_IM(ans) = S;
                } else {
                        F_RE(ans) = C * sinhf(x);
                        F_IM(ans) = S * coshf(x);
                }
        }
        if (hx < 0)
                F_RE(ans) = -F_RE(ans);
        if (hy < 0)
                F_IM(ans) = -F_IM(ans);
        return (ans);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2395 Tue Nov  4 19:01:24 2014
new/usr/src/lib/libm/common/complex/csinhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csinhl = csinhl
#pragma weak csinhl = __csinhl

#include "libm.h"       /* coshl/expl/fabsl/scalbnl/sincosl/sinhl/__k_cexpl */
#include "complex_wrapper.h"

/* INDENT OFF */
static const long double zero = 0.0L, half = 0.5L;
/* INDENT ON */

ldcomplex
csinhl(ldcomplex z) {
        long double t, x, y, S, C;
        int hx, ix, hy, iy, n;
        ldcomplex ans;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_XWORD(y);
        iy = hy & 0x7fffffff;
        x = fabsl(x);
        y = fabsl(y);

        (void) sincosl(y, &S, &C);
        if (ix >= 0x4004e000) { /* |x| > 60 = prec/2 (14,28,34,60) */
                if (ix >= 0x400C62E4) { /* |x| > 11356.52... ~ log(2**16384) */
                        if (ix >= 0x7fff0000) { /* |x| is inf or NaN */
                                if (y == zero) {
                                        LD_RE(ans) = x;
                                        LD_IM(ans) = y;
```



```
                                } else if (iy >= 0x7fff0000) {
                                        LD_RE(ans) = x;
                                        LD_IM(ans) = x - y;
                                } else {
                                        LD_RE(ans) = C * x;
                                        LD_IM(ans) = S * x;
                                }
                        } else {
                                /* return exp(x)=t*2**n */
                                t = __k_cexpl(x, &n);
                                LD_RE(ans) = scalbnl(C * t, n - 1);
                                LD_IM(ans) = scalbnl(S * t, n - 1);
                        }
                } else {
                        t = expl(x) * half;
                        LD_RE(ans) = C * t;
                        LD_IM(ans) = S * t;
                }
        } else {
                if (x == zero) {        /* x = 0, return (0,S) */
                        LD_RE(ans) = zero;
                        LD_IM(ans) = S;
                } else {
                        LD_RE(ans) = C * sinhl(x);
                        LD_IM(ans) = S * coshl(x);
                }
        }
        if (hx < 0)
                LD_RE(ans) = -LD_RE(ans);
        if (hy < 0)
                LD_IM(ans) = -LD_IM(ans);
        return (ans);
}
```

*_____unchanged_portion_omitted_*

****
1292 Tue Nov 4 19:01:24 2014
new/usr/src/lib/libm/common/complex/csinl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csinl = csinl
#pragma weak csinl = __csinl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
csinl(ldcomplex z) {
        long double x, y;
        ldcomplex ans, ct;

        x = LD_RE(z);
        y = LD_IM(z);
        LD_RE(z) = y;
        LD_IM(z) = -x;
        ct = csinhl(z);
        LD_RE(ans) = -LD_IM(ct);
        LD_IM(ans) = LD_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5665 Tue Nov  4 19:01:24 2014
new/usr/src/lib/libm/common/complex/csqrt.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csqrt = csqrt
#pragma weak csqrt = __csqrt

/* INDENT OFF */
/*
 * dcomplex csqrt(dcomplex z);
 *
 *                                                  2    2    2
 * Let w=r+i*s = sqrt(x+iy). Then (r + i s)  = r  - s  + i 2sr = x + i y.
 *
 * Hence x = r*r-s*s, y = 2sr.
 *
 * Note that x*x+y*y = (s*s+r*r)**2. Thus, we have
 *
 *                                  ________
 *           2    2     / 2    2
 *      (1) r  + s  = \/ x  + y  ,
 *
 *           2    2
 *       (2) r  - s  = x
 *
 *      (3) 2sr = y.
 *
 * Perform (1)-(2) and (1)+(2), we obtain
 *
 *              2
 *      (4) 2 r   = hypot(x,y)+x,
 *
 *              2
 *       (5) 2*s   = hypot(x,y)-x
 *
 *                     ________
 *                    / 2    2
 * where hypot(x,y) = \/ x  + y  .
```



```
 *
 * In order to avoid numerical cancellation, we use formula (4) for
 * positive x, and (5) for negative x. The other component is then
 * computed by formula (3).
 *
 *
 * ALGORITHM
 * ------------------
 *
 * (assume x and y are of medium size, i.e., no over/underflow in squaring)
 *
 * If x >=0 then
 *                                  ________
 *                                 /  2    2
 *            2     \/  x  + y    +  x               y
 *           r =   ---------------------,      s = -------;    (6)
 *                                 2                 2 r
 *
 * (note that we choose sign(s) = sign(y) to force r >=0).
 * Otherwise,
 *                                  ________
 *                                 /  2    2
 *            2     \/  x  + y    -  x               y
 *           s =   ---------------------,      r = -------;    (7)
 *                                 2                 2 s
 *
 * EXCEPTION:
 *
 * One may use the polar coordinate of a complex number to justify the
 * following exception cases:
 *
 * EXCEPTION CASES (conform to ISO/IEC 9899:1999(E)):
 *    csqrt(+-0+ i 0   ) =  0    + i 0
 *    csqrt( x + i inf ) =  inf  + i inf for all x (including NaN)
 *    csqrt( x + i NaN ) =  NaN  + i NaN with invalid for finite x
 *    csqrt(-inf+ iy   ) =  0    + i inf for finite positive-signed y
 *    csqrt(+inf+ iy   ) =  inf  + i 0   for finite positive-signed y
 *    csqrt(-inf+ i NaN) =  NaN  +-i inf
 *    csqrt(+inf+ i NaN) =  inf  + i NaN
 *    csqrt(NaN + i y  ) =  NaN  + i NaN for finite y
 *    csqrt(NaN + i NaN) =  NaN  + i NaN
 */
/* INDENT ON */

#include "libm.h"               /* fabs/sqrt */
#include "complex_wrapper.h"

/* INDENT OFF */
static const double
        two300 = 2.037035976334486086627e+90,
        twom300 = 4.909093465297726553310e-91,
        two599 = 2.074757784440496479260e+180,
        twom601 = 1.204959932551442058870e-181,
        two = 2.0,
        zero = 0.0,
        half = 0.5;
/* INDENT ON */

dcomplex
csqrt(dcomplex z) {
        dcomplex ans;
        double x, y, t, ax, ay;
        int n, ix, iy, hx, hy, lx, ly;

        x = D_RE(z);
        y = D_IM(z);
```

```
127         hx = HI_WORD(x);
128         lx = LO_WORD(x);
129         hy = HI_WORD(y);
130         ly = LO_WORD(y);
131         ix = hx & 0x7fffffff;
132         iy = hy & 0x7fffffff;
133         ay = fabs(y);
134         ax = fabs(x);
135         if (ix >= 0x7ff00000 || iy >= 0x7ff00000) {
136                 /* x or y is Inf or NaN */
137                 if (ISINF(iy, ly))
138                         D_IM(ans) = D_RE(ans) = ay;
139                 else if (ISINF(ix, lx)) {
140                         if (hx > 0) {
141                                 D_RE(ans) = ax;
142                                 D_IM(ans) = ay * zero;
143                         } else {
144                                 D_RE(ans) = ay * zero;
145                                 D_IM(ans) = ax;
146                         }
147                 } else
148                         D_IM(ans) = D_RE(ans) = ax + ay;
149         } else if ((iy | ly) == 0) {    /* y = 0 */
150                 if (hx >= 0) {
151                         D_RE(ans) = sqrt(ax);
152                         D_IM(ans) = zero;
153                 } else {
154                         D_IM(ans) = sqrt(ax);
155                         D_RE(ans) = zero;
156                 }
157         } else if (ix >= iy) {
158                 n = (ix - iy) >> 20;
159                 if (n >= 30) {  /* x >> y or y=0 */
160                         t = sqrt(ax);
161                 } else if (ix >= 0x5f300000) {  /* x > 2**500 */
162                         ax *= twom601;
163                         y *= twom601;
164                         t = two300 * sqrt(ax + sqrt(ax * ax + y * y));
165                 } else if (iy < 0x20b00000) {   /* y < 2**-500 */
166                         ax *= two599;
167                         y *= two599;
168                         t = twom300 * sqrt(ax + sqrt(ax * ax + y * y));
169                 } else
170                         t = sqrt(half * (ax + sqrt(ax * ax + ay * ay)));
171                 if (hx >= 0) {
172                         D_RE(ans) = t;
173                         D_IM(ans) = ay / (t + t);
174                 } else {
175                         D_IM(ans) = t;
176                         D_RE(ans) = ay / (t + t);
177                 }
178         } else {
179                 n = (iy - ix) >> 20;
180                 if (n >= 30) {  /* y >> x */
181                         if (n >= 60)
182                                 t = sqrt(half * ay);
183                         else if (iy >= 0x7fe00000)
184                                 t = sqrt(half * ay + half * ax);
185                         else if (ix <= 0x00100000)
186                                 t = half * sqrt(two * (ay + ax));
187                         else
188                                 t = sqrt(half * (ay + ax));
189                 } else if (iy >= 0x5f300000) {  /* y > 2**500 */
190                         ax *= twom601;
191                         y *= twom601;
192                         t = two300 * sqrt(ax + sqrt(ax * ax + y * y));
```



```
193                 } else if (ix < 0x20b00000) {   /* x < 2**-500 */
194                         ax *= two599;
195                         y *= two599;
196                         t = twom300 * sqrt(ax + sqrt(ax * ax + y * y));
197                 } else
198                         t = sqrt(half * (ax + sqrt(ax * ax + ay * ay)));
199                 if (hx >= 0) {
200                         D_RE(ans) = t;
201                         D_IM(ans) = ay / (t + t);
202                 } else {
203                         D_IM(ans) = t;
204                         D_RE(ans) = ay / (t + t);
205                 }
206         }
207         if (hy < 0)
208                 D_IM(ans) = -D_IM(ans);
209         return (ans);
210 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    2221 Tue Nov  4 19:01:25 2014
new/usr/src/lib/libm/common/complex/csqrtf.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csqrtf = csqrtf
#pragma weak csqrtf = __csqrtf

#include "libm.h"               /* sqrt/fabsf/sqrtf */
#include "complex_wrapper.h"

/* INDENT OFF */
static const float zero = 0.0F;
/* INDENT ON */

fcomplex
csqrtf(fcomplex z) {
        fcomplex ans;
        double dt, dx, dy;
        float x, y, t, ax, ay, w;
        int ix, iy, hx, hy;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        hy = THE_WORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        ay = fabsf(y);
        ax = fabsf(x);
        if (ix >= 0x7f800000 || iy >= 0x7f800000) {
                /* x or y is Inf or NaN */
                if (iy == 0x7f800000)
                        F_IM(ans) = F_RE(ans) = ay;
                else if (ix == 0x7f800000) {
                        if (hx > 0) {
                                F_RE(ans) = ax;
```



```
                                F_IM(ans) = ay * zero;
                        } else {
                                F_RE(ans) = ay * zero;
                                F_IM(ans) = ax;
                        }
                } else
                        F_IM(ans) = F_RE(ans) = ax + ay;
        } else if (iy == 0) {
                if (hx >= 0) {
                        F_RE(ans) = sqrtf(ax);
                        F_IM(ans) = zero;
                } else {
                        F_IM(ans) = sqrtf(ax);
                        F_RE(ans) = zero;
                }
        } else {
                dx = (double) ax;
                dy = (double) ay;
                dt = sqrt(0.5 * (sqrt(dx * dx + dy * dy) + dx));
                t = (float) dt;
                w = (float) (dy / (dt + dt));
                if (hx >= 0) {
                        F_RE(ans) = t;
                        F_IM(ans) = w;
                } else {
                        F_IM(ans) = t;
                        F_RE(ans) = w;
                }
        }
        if (hy < 0)
                F_IM(ans) = -F_IM(ans);
        return (ans);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    3769 Tue Nov  4 19:01:25 2014
new/usr/src/lib/libm/common/complex/csqrtl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __csqrtl = csqrtl
#pragma weak csqrtl = __csqrtl

#include "libm.h"               /* fabsl/isinfl/sqrtl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double
        twom9001 = 2.685400271600303495742176510061569304365e-2710L,
        twom4500 = 2.317498768759242942326324862381544149252e-1355L,
        two8999 = 9.309599118012234350258234737216329031093 4e+2708L,
        two4500 = 4.314996898727097428377780354557172225080 6e+1354L,
        zero = 0.0L,
        half = 0.5L,
        two = 2.0L;
/* INDENT ON */

ldcomplex
csqrtl(ldcomplex z) {
        ldcomplex ans;
        long double x, y, t, ax, ay;
        int n, ix, iy, hx, hy;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        hy = HI_XWORD(y);
        ix = hx & 0x7fffffff;
        iy = hy & 0x7fffffff;
        ay = fabsl(y);
        ax = fabsl(x);
```



```
        if (ix >= 0x7fff0000 || iy >= 0x7fff0000) {
                /* x or y is Inf or NaN */
                if (isinfl(y))
                        LD_IM(ans) = LD_RE(ans) = ay;
                else if (isinfl(x)) {
                        if (hx > 0) {
                                LD_RE(ans) = ax;
                                LD_IM(ans) = ay * zero;
                        } else {
                                LD_RE(ans) = ay * zero;
                                LD_IM(ans) = ax;
                        }
                } else
                        LD_IM(ans) = LD_RE(ans) = ax + ay;
        } else if (y == zero) {
                if (hx >= 0) {
                        LD_RE(ans) = sqrtl(ax);
                        LD_IM(ans) = zero;
                } else {
                        LD_IM(ans) = sqrtl(ax);
                        LD_RE(ans) = zero;
                }
        } else if (ix >= iy) {
                n = (ix - iy) >> 16;
#if defined(__x86)              /* 64 significant bits */
                if (n >= 35)
#else                           /* 113 significant bits  */
                if (n >= 60)
#endif
                        t = sqrtl(ax);
                else if (ix >= 0x5f3f0000) {    /* x > 2**8000 */
                        ax *= twom9001;
                        y *= twom9001;
                        t = two4500 * sqrtl(ax + sqrtl(ax * ax + y * y));
                } else if (iy <= 0x20bf0000) {  /* y < 2**-8000 */
                        ax *= two8999;
                        y *= two8999;
                        t = twom4500 * sqrtl(ax + sqrtl(ax * ax + y * y));
                } else
                        t = sqrtl(half * (ax + sqrtl(ax * ax + y * y)));

                if (hx >= 0) {
                        LD_RE(ans) = t;
                        LD_IM(ans) = ay / (t + t);
                } else {
                        LD_IM(ans) = t;
                        LD_RE(ans) = ay / (t + t);
                }
        } else {
                n = (iy - ix) >> 16;
#if defined(__x86)              /* 64 significant bits */
                if (n >= 35) {  /* } */
#else                           /* 113 significant bits  */
                if (n >= 60) {
#endif
                        if (n >= 120)
                                t = sqrtl(half * ay);
                        else if (iy >= 0x7ffe0000)
                                t = sqrtl(half * ay + half * ax);
                        else if (ix <= 0x00010000)
                                t = half * (sqrtl(two * (ax + ay)));
                        else
                                t = sqrtl(half * (ax + ay));
                } else if (iy >= 0x5f3f0000) {  /* y > 2**8000 */
                        ax *= twom9001;
                        y *= twom9001;
```

```
127                         t = two4500 * sqrtl(ax + sqrtl(ax * ax + y * y));
128                 } else if (ix <= 0x20bf0000) {
129                         ax *= two8999;
130                         y *= two8999;
131                         t = twom4500 * sqrtl(ax + sqrtl(ax * ax + y * y));
132                 } else
133                         t = sqrtl(half * (ax + sqrtl(ax * ax + y * y)));

135                 if (hx >= 0) {
136                         LD_RE(ans) = t;
137                         LD_IM(ans) = ay / (t + t);
138                 } else {
139                         LD_IM(ans) = t;
140                         LD_RE(ans) = ay / (t + t);
141                 }
142         }
143         if (hy < 0)
144                 LD_IM(ans) = -LD_IM(ans);
145         return (ans);
146 }
```

unchanged_portion_omitted_

```
**********************************************************
    1558 Tue Nov  4 19:01:26 2014
new/usr/src/lib/libm/common/complex/ctan.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #pragma weak __ctan = ctan
  30 #pragma weak ctan = __ctan

  32 /* INDENT OFF */
  33 /*
  34  * dcomplex ctan(dcomplex z);
  35  *
  36  * If z = x+iy, then since ctan(iz) = i*ctanh(z),  we have
  37  *
  38  * ctan(z)      = ctan((-1)*(-z)) = ctan(i*i*(-z))
  39  *              = i*ctanh(i*(-z)) = i*ctanh(i*(-x-yi))
  40  *              = i*ctanh(y-ix)
  41  *              = -Im(ctanh(y-ix))+i*Re(ctanh(y-ix))
  42  */
  43 /* INDENT ON */

  45 #include "libm.h"
  46 #include "complex_wrapper.h"

  48 dcomplex
  49 ctan(dcomplex z) {
  50         double x, y;
  51         dcomplex ans, ct;

  53         x = D_RE(z);
  54         y = D_IM(z);
  55         D_RE(z) = y;
  56         D_IM(z) = -x;
  57         ct = ctanh(z);
  58         D_RE(ans) = -D_IM(ct);
  59         D_IM(ans) = D_RE(ct);
  60         return (ans);
```



```
  61 }
______unchanged_portion_omitted_
```

****
1275 Tue Nov 4 19:01:26 2014
new/usr/src/lib/libm/common/complex/ctanf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ctanf = ctanf
#pragma weak ctanf = __ctanf

#include "libm.h"
#include "complex_wrapper.h"

fcomplex
ctanf(fcomplex z) {
        float x, y;
        fcomplex ans, ct;

        x = F_RE(z);
        y = F_IM(z);
        F_RE(z) = y;
        F_IM(z) = -x;
        ct = ctanhf(z);
        F_RE(ans) = -F_IM(ct);
        F_IM(ans) = F_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5411 Tue Nov  4 19:01:26 2014
new/usr/src/lib/libm/common/complex/ctanh.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ctanh = ctanh
#pragma weak ctanh = __ctanh

/* INDENT OFF */
/*
 * dcomplex ctanh(dcomplex z);
 *
 *            tanh x  + i tan y      sinh 2x  +  i sin 2y
 * ctanh z = --------------------- = -------------------
 *           1 + i tanh(x)tan(y)       cosh 2x + cos 2y
 *
 * For |x| >= prec/2 (14,28,34,60 for single, double, double extended, quad),
 * we use
 *
 *                         1   2x                                  2 sin 2y
 *    cosh 2x = sinh 2x = --- e    and hence  ctanh z = 1 + i -----------;
 *                         2                                        2x
 *                                                                 e
 *
 * otherwise, to avoid cancellation, for |x| < prec/2,
 *                               2x     2
 *                            (e   - 1)        2       2
 *    cosh 2x + cos 2y = 1 + ------------ + cos y - sin y
 *                                2x
 *                             2 e
 *
 *                        1    2x     2  -2x         2
 *                    = --- (e   - 1)  e     + 2 cos y
 *                        2
 * and
 *
 *                  [            2x      ]
```



```
 *                1  [  2x       e   - 1  ]
 *    sinh 2x = --- [ e  - 1 + --------- ]
 *                2  [              2x   ]
 *                   [             e     ]
 *                                          2x
 * Implementation notes:  let t = expm1(2x) = e   - 1, then
 *
 *                       1    [  t*t        2  ]             1    [      t  ]
 * cosh 2x + cos 2y = --- * [ ----- + 4 cos y ];  sinh 2x = --- * [ t + --- ]
 *                       2    [  t+1          ]             2    [     t+1 ]
 *
 * Hence,
 *
 *                           t*t+2t                  [4(t+1)(cos y)]*(sin y)
 *    ctanh z = ------------------------------ + i --------------------------
 *              t*t+[4(t+1)(cos y)](cos y)     t*t+[4(t+1)(cos y)](cos y)
 *
 * EXCEPTION (conform to ISO/IEC 9899:1999(E)):
 *      ctanh(0,0)=(0,0)
 *      ctanh(x,inf) = (NaN,NaN) for finite x
 *      ctanh(x,NaN) = (NaN,NaN) for finite x
 *      ctanh(inf,y) = 1+ i*0*sin(2y) for positive-signed finite y
 *      ctanh(inf,inf) = (1, +-0)
 *      ctanh(inf,NaN) = (1, +-0)
 *      ctanh(NaN,0) = (NaN,0)
 *      ctanh(NaN,y) = (NaN,NaN) for non-zero y
 *      ctanh(NaN,NaN) = (NaN,NaN)
 */
/* INDENT ON */

#include "libm.h"              /* exp/expm1/fabs/sin/tanh/sincos */
#include "complex_wrapper.h"

static const double four = 4.0, two = 2.0, one = 1.0, zero = 0.0;

dcomplex
ctanh(dcomplex z) {
        double t, r, v, u, x, y, S, C;
        int hx, ix, lx, hy, iy, ly;
        dcomplex ans;

        x = D_RE(z);
        y = D_IM(z);
        hx = HI_WORD(x);
        lx = LO_WORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_WORD(y);
        ly = LO_WORD(y);
        iy = hy & 0x7fffffff;
        x = fabs(x);
        y = fabs(y);

        if ((iy | ly) == 0) {   /* ctanh(x,0) = (x,0) for x = 0 or NaN */
                D_RE(ans) = tanh(x);
                D_IM(ans) = zero;
        } else if (iy >= 0x7ff00000) {  /* y is inf or NaN */
                if (ix < 0x7ff00000)    /* catanh(finite x,inf/nan) is nan */
                        D_RE(ans) = D_IM(ans) = y - y;
                else if (((ix - 0x7ff00000) | lx) == 0) {       /* x is inf */
                        D_RE(ans) = one;
                        D_IM(ans) = zero;
                } else {
                        D_RE(ans) = x + y;
                        D_IM(ans) = y - y;
                }
```

```
127         } else if (ix >= 0x403c0000) {
128                 /*
129                  * |x| > 28 = prec/2 (14,28,34,60)
130                  * ctanh z ~ 1 + i (sin2y)/(exp(2x))
131                  */
132                 D_RE(ans) = one;
133                 if (iy < 0x7fe00000)    /* t = sin(2y) */
134                         S = sin(y + y);
135                 else {
136                         (void) sincos(y, &S, &C);
137                         S = (S + S) * C;
138                 }
139                 if (ix >= 0x7fe00000) { /* |x| > max/2 */
140                         if (ix >= 0x7ff00000) { /* |x| is inf or NaN */
141                                 if (((ix - 0x7ff00000) | lx) != 0)
142                                         D_RE(ans) = D_IM(ans) = x + y;
143                                                                 /* x is NaN */
144                                 else
145                                         D_IM(ans) = zero * S;   /* x is inf */
146                         } else
147                                 D_IM(ans) = S * exp(-x);        /* underflow */
148                 } else
149                         D_IM(ans) = (S + S) * exp(-(x + x));
150                                                         /* 2 sin 2y / exp(2x) */
151         } else {
152                 /* INDENT OFF */
153                 /*
154                  *                          t*t+2t
155                  *    ctanh z = -------------------------- +
156                  *               t*t+[4(t+1)(cos y)](cos y)
157                  *
158                  *                  [4(t+1)(cos y)]*(sin y)
159                  *              i -------------------------
160                  *                t*t+[4(t+1)(cos y)](cos y)
161                  */
162                 /* INDENT ON */
163                 (void) sincos(y, &S, &C);
164                 t = expm1(x + x);
165                 r = (four * C) * (t + one);
166                 u = t * t;
167                 v = one / (u + r * C);
168                 D_RE(ans) = (u + two * t) * v;
169                 D_IM(ans) = (r * S) * v;
170         }
171         if (hx < 0)
172                 D_RE(ans) = -D_RE(ans);
173         if (hy < 0)
174                 D_IM(ans) = -D_IM(ans);
175         return (ans);
176 }
```

unchanged_portion_omitted_

```
**********************************************************
    3091 Tue Nov  4 19:01:27 2014
new/usr/src/lib/libm/common/complex/ctanhf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ctanhf = ctanhf
#pragma weak ctanhf = __ctanhf

#include "libm.h"               /* expf/expm1f/fabsf/sincosf/sinf/tanhf */
#include "complex_wrapper.h"

/* INDENT OFF */
static const float four = 4.0F, two = 2.0F, one = 1.0F, zero = 0.0F;
/* INDENT ON */

fcomplex
ctanhf(fcomplex z) {
        float r, u, v, t, x, y, S, C;
        int hx, ix, hy, iy;
        fcomplex ans;

        x = F_RE(z);
        y = F_IM(z);
        hx = THE_WORD(x);
        ix = hx & 0x7fffffff;
        hy = THE_WORD(y);
        iy = hy & 0x7fffffff;
        x = fabsf(x);
        y = fabsf(y);

        if (iy == 0) {          /* ctanh(x,0) = (x,0) for x = 0 or NaN */
                F_RE(ans) = tanhf(x);
                F_IM(ans) = zero;
        } else if (iy >= 0x7f800000) {  /* y is inf or NaN */
                if (ix < 0x7f800000)    /* catanh(finite x,inf/nan) is nan */
                        F_RE(ans) = F_IM(ans) = y - y;
                else if (ix == 0x7f800000) {    /* x is inf */
```



```
                        F_RE(ans) = one;
                        F_IM(ans) = zero;
                } else {
                        F_RE(ans) = x + y;
                        F_IM(ans) = y - y;
                }
        } else if (ix >= 0x41600000) {
                /*
                 * |x| > 14 = prec/2 (14,28,34,60)
                 * ctanh z ~ 1 + i (sin2y)/(exp(2x))
                 */
                F_RE(ans) = one;
                if (iy < 0x7f000000)    /* t = sin(2y) */
                        S = sinf(y + y);
                else {
                        (void) sincosf(y, &S, &C);
                        S = (S + S) * C;
                }
                if (ix >= 0x7f000000) { /* |x| > max/2 */
                        if (ix >= 0x7f800000) { /* |x| is inf or NaN */
                                if (ix > 0x7f800000)    /* x is NaN */
                                        F_RE(ans) = F_IM(ans) = x + y;
                                else
                                        F_IM(ans) = zero * S;   /* x is inf */
                        } else
                                F_IM(ans) = S * expf(-x);       /* underflow */
                } else
                        F_IM(ans) = (S + S) * expf(-(x + x));
                                                        /* 2 sin 2y / exp(2x) */
        } else {
                /* INDENT OFF */
                /*
                 *                          t*t+2t
                 *    ctanh z = --------------------------
                 *               t*t+[4(t+1)(cos y)](cos y)
                 *
                 *                  [4(t+1)(cos y)]*(sin y)
                 *              i --------------------------
                 *                t*t+[4(t+1)(cos y)](cos y)
                 */
                /* INDENT ON */
                (void) sincosf(y, &S, &C);
                t = expm1f(x + x);
                r = (four * C) * (t + one);
                u = t * t;
                v = one / (u + r * C);
                F_RE(ans) = (u + two * t) * v;
                F_IM(ans) = (r * S) * v;
        }
        if (hx < 0)
                F_RE(ans) = -F_RE(ans);
        if (hy < 0)
                F_IM(ans) = -F_IM(ans);
        return (ans);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    3183 Tue Nov  4 19:01:27 2014
new/usr/src/lib/libm/common/complex/ctanhl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ctanhl = ctanhl
#pragma weak ctanhl = __ctanhl

#include "libm.h"       /* expl/expm1l/fabsl/isinfl/isnanl/sincosl/sinl/tanhl */
#include "complex_wrapper.h"
#include "longdouble.h"

/* INDENT OFF */
static const long double four = 4.0L, two = 2.0L, one = 1.0L, zero = 0.0L;
/* INDENT ON */

ldcomplex
ctanhl(ldcomplex z) {
        long double r, u, v, t, x, y, S, C;
        int hx, ix, hy, iy;
        ldcomplex ans;

        x = LD_RE(z);
        y = LD_IM(z);
        hx = HI_XWORD(x);
        ix = hx & 0x7fffffff;
        hy = HI_XWORD(y);
        iy = hy & 0x7fffffff;
        x = fabsl(x);
        y = fabsl(y);

        if (y == zero) {        /* ctanh(x,0) = (x,0) for x = 0 or NaN */
                LD_RE(ans) = tanhl(x);
                LD_IM(ans) = zero;
        } else if (iy >= 0x7fff0000) {  /* y is inf or NaN */
                if (ix < 0x7fff0000)    /* catanh(finite x,inf/nan) is nan */
                        LD_RE(ans) = LD_IM(ans) = y - y;
```



```
                else if (isinfl(x)) {   /* x is inf */
                        LD_RE(ans) = one;
                        LD_IM(ans) = zero;
                } else {
                        LD_RE(ans) = x + y;
                        LD_IM(ans) = y - y;
                }
        } else if (ix >= 0x4004e000) {
                /* INDENT OFF */
                /*
                 * |x| > 60 = prec/2 (14,28,34,60)
                 * ctanh z ~ 1 + i (sin2y)/(exp(2x))
                 */
                /* INDENT ON */
                LD_RE(ans) = one;
                if (iy < 0x7ffe0000)    /* t = sin(2y) */
                        S = sinl(y + y);
                else {
                        (void) sincosl(y, &S, &C);
                        S = (S + S) * C;
                }
                if (ix >= 0x7ffe0000) { /* |x| > max/2 */
                        if (ix >= 0x7fff0000) { /* |x| is inf or NaN */
                                if (isnanl(x))  /* x is NaN */
                                        LD_RE(ans) = LD_IM(ans) = x + y;
                                else
                                        LD_IM(ans) = zero * S;  /* x is inf */
                        } else
                                LD_IM(ans) = S * expl(-x);      /* underflow */
                } else
                        LD_IM(ans) = (S + S) * expl(-(x + x));
                                                        /* 2 sin 2y / exp(2x) */
        } else {
                /* INDENT OFF */
                /*
                 *                          t*t+2t
                 *    ctanh z = --------------------------
                 *               t*t+[4(t+1)(cos y)](cos y)
                 *
                 *                  [4(t+1)(cos y)]*(sin y)
                 *              i --------------------------
                 *                t*t+[4(t+1)(cos y)](cos y)
                 */
                /* INDENT ON */
                sincosl(y, &S, &C);
                t = expm1l(x + x);
                r = (four * C) * (t + one);
                u = t * t;
                v = one / (u + r * C);
                LD_RE(ans) = (u + two * t) * v;
                LD_IM(ans) = (r * S) * v;
        }
        if (hx < 0)
                LD_RE(ans) = -LD_RE(ans);
        if (hy < 0)
                LD_IM(ans) = -LD_IM(ans);
        return (ans);
}
```

*_____unchanged_portion_omitted_*

****
1292 Tue Nov 4 19:01:28 2014
new/usr/src/lib/libm/common/complex/ctanl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ctanl = ctanl
#pragma weak ctanl = __ctanl

#include "libm.h"
#include "complex_wrapper.h"

ldcomplex
ctanl(ldcomplex z) {
        long double x, y;
        ldcomplex ans, ct;

        x = LD_RE(z);
        y = LD_IM(z);
        LD_RE(z) = y;
        LD_IM(z) = -x;
        ct = ctanhl(z);
        LD_RE(ans) = -LD_IM(ct);
        LD_IM(ans) = LD_RE(ct);
        return (ans);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   21395 Tue Nov  4 19:01:28 2014
new/usr/src/lib/libm/common/m9x/__fex_hdlr.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "fenv_synonyms.h"
#undef lint
#include <signal.h>
#include <siginfo.h>
#include <ucontext.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <thread.h>
#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif
#include <fenv.h>
#include "fex_handler.h"
#include "fenv_inlines.h"

#if defined(__sparc) && !defined(__sparcv9)
#include <sys/procfs.h>
#endif

/* 2.x signal.h doesn't declare sigemptyset or sigismember
   if they're #defined (see sys/signal.h) */
extern int sigemptyset(sigset_t *);
extern int sigismember(const sigset_t *, int);

/* external globals */
void (*__mt_fex_sync)() = NULL; /* for synchronization with libmtsk */
#pragma weak __mt_fex_sync

void (*__libm_mt_fex_sync)() = NULL; /* new, improved version of above */
#pragma weak __libm_mt_fex_sync
```



```
/* private variables */
static fex_handler_t main_handlers;
static int handlers_initialized = 0;
static thread_key_t handlers_key;
static mutex_t handlers_key_lock = DEFAULTMUTEX;

static struct sigaction oact = { 0, SIG_DFL };
static mutex_t hdlr_lock = DEFAULTMUTEX;
static int hdlr_installed = 0;

/* private const data */
static const int te_bit[FEX_NUM_EXC] = {
        1 << fp_trap_inexact,
        1 << fp_trap_division,
        1 << fp_trap_underflow,
        1 << fp_trap_overflow,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid,
        1 << fp_trap_invalid
};
______unchanged_portion_omitted_
```

```
**********************************************************
   36554 Tue Nov  4 19:01:29 2014
new/usr/src/lib/libm/common/m9x/__fex_i386.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "fenv_synonyms.h"
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <siginfo.h>
#include <ucontext.h>
#include <thread.h>
#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif
#include <fenv.h>
#include "fex_handler.h"
#include "fenv_inlines.h"

#if defined(__amd64)
#define test_sse_hw     1
#else
/*
 * The following variable lives in libc on Solaris 10, where it
 * gets set to a nonzero value at startup time on systems with SSE.
 */
int _sse_hw = 0;
#pragma weak _sse_hw
#define test_sse_hw     &_sse_hw && _sse_hw
#endif

static int accrued = 0;
static thread_key_t accrued_key;
static mutex_t accrued_key_lock = DEFAULTMUTEX;
```



```
int *
__fex_accrued()
{
        int             *p;

        if (thr_main())
                return &accrued;
        else {
                p = NULL;
                mutex_lock(&accrued_key_lock);
                if (thr_getspecific(accrued_key, (void **)&p) != 0 &&
                        thr_keycreate(&accrued_key, free) != 0) {
                        mutex_unlock(&accrued_key_lock);
                        return NULL;
                }
                mutex_unlock(&accrued_key_lock);
                if (!p) {
                        if ((p = (int*) malloc(sizeof(int))) == NULL)
                                return NULL;
                        if (thr_setspecific(accrued_key, (void *)p) != 0) {
                                (void)free(p);
                                return NULL;
                        }
                        *p = 0;
                }
                return p;
        }
}
```

*unchanged_portion_omitted_*

****
21295 Tue Nov 4 19:01:29 2014
new/usr/src/lib/libm/common/m9x/__fex_sparc.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#if defined(__sparc)
#include "fenv_synonyms.h"
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>
#include <siginfo.h>
#include <thread.h>
#include <ucontext.h>
#include <math.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#endif
#include <fenv.h>

#include "fenv_inlines.h"
#include "libm_inlines.h"

#ifdef __sparcv9

#define FPreg(X)        &uap->uc_mcontext.fpregs.fpu_fr.fpu_regs[X]

#define FPREG(X)        &uap->uc_mcontext.fpregs.fpu_fr.fpu_dregs[(X>>1)| \
                                ((X&1)<<4)]

#else

#include <sys/procfs.h>

#define FPxreg(X)       &((prxregset_t*)uap->uc_mcontext.xrs.xrs_ptr)->pr_un.pr_

#define FPreg(X)        &uap->uc_mcontext.fpregs.fpu_fr.fpu_regs[X]
```



```
#define FPREG(X)        ((X & 1)? FPxreg(X - 1) : FPreg(X))

#endif  /* __sparcv9 */

#include "fex_handler.h"

/* avoid dependence on libsunmath */
static enum fp_class_type
my_fp_classl(long double *a)
{
        int             msw = *(int*)a & ~0x80000000;

        if (msw >= 0x7fff0000) {
                if (((msw & 0xffff) | *(1+(int*)a) | *(2+(int*)a) | *(3+(int*)a)
                        return fp_infinity;
                else if (msw & 0x8000)
                        return fp_quiet;
                else
                        return fp_signaling;
        } else if (msw < 0x10000) {
                if ((msw | *(1+(int*)a) | *(2+(int*)a) | *(3+(int*)a)) == 0)
                        return fp_zero;
                else
                        return fp_subnormal;
        } else
                return fp_normal;
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   39067 Tue Nov  4 19:01:30 2014
new/usr/src/lib/libm/common/m9x/__fex_sse.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "fenv_synonyms.h"
#include <ucontext.h>
#include <fenv.h>
#if defined(__SUNPRO_C)
#include <sunmath.h>
#else
#include <sys/ieeefp.h>
#endif
#include "fex_handler.h"
#include "fenv_inlines.h"

#if !defined(REG_PC)
#define REG_PC  EIP
#endif

#if !defined(REG_PS)
#define REG_PS  EFL
#endif

#ifdef __amd64
#define regno(X)        ((X < 4)? REG_RAX - X : \
                        ((X > 4)? REG_RAX + 1 - X : REG_RSP))
#else
#define regno(X)        (EAX - X)
#endif

/*
 * Support for SSE instructions
 */

/*
 * Decode an SSE instruction.  Fill in *inst and return the length of the
```



```
 * instruction in bytes.  Return 0 if the instruction is not recognized.
 */
int
__fex_parse_sse(ucontext_t *uap, sseinst_t *inst)
{
        unsigned char   *ip;
        char            *addr;
        int             i, dbl, simd, rex, modrm, sib, r;

        i = 0;
        ip = (unsigned char *)uap->uc_mcontext.gregs[REG_PC];

        /* look for pseudo-prefixes */
        dbl = 0;
        simd = SIMD;
        if (ip[i] == 0xF3) {
                simd = 0;
                i++;
        } else if (ip[i] == 0x66) {
                dbl = DOUBLE;
                i++;
        } else if (ip[i] == 0xF2) {
                dbl = DOUBLE;
                simd = 0;
                i++;
        }

        /* look for AMD64 REX prefix */
        rex = 0;
        if (ip[i] >= 0x40 && ip[i] <= 0x4F) {
                rex = ip[i];
                i++;
        }

        /* parse opcode */
        if (ip[i++] != 0x0F)
                return 0;
        switch (ip[i++]) {
        case 0x2A:
                inst->op = (int)cvtsi2ss + simd + dbl;
                if (!simd)
                        inst->op = (int)inst->op + (rex & 8);
                break;

        case 0x2C:
                inst->op = (int)cvttss2si + simd + dbl;
                if (!simd)
                        inst->op = (int)inst->op + (rex & 8);
                break;

        case 0x2D:
                inst->op = (int)cvtss2si + simd + dbl;
                if (!simd)
                        inst->op = (int)inst->op + (rex & 8);
                break;

        case 0x2E:
                /* oddball: scalar instruction in a SIMD opcode group */
                if (!simd)
                        return 0;
                inst->op = (int)ucomiss + dbl;
                break;

        case 0x2F:
                /* oddball: scalar instruction in a SIMD opcode group */
                if (!simd)
```

```
127                         return 0;
128                 inst->op = (int)comiss + dbl;
129                 break;

131         case 0x51:
132                 inst->op = (int)sqrtss + simd + dbl;
133                 break;

135         case 0x58:
136                 inst->op = (int)addss + simd + dbl;
137                 break;

139         case 0x59:
140                 inst->op = (int)mulss + simd + dbl;
141                 break;

143         case 0x5A:
144                 inst->op = (int)cvtss2sd + simd + dbl;
145                 break;

147         case 0x5B:
148                 if (dbl) {
149                         if (simd)
150                                 inst->op = cvtps2dq;
151                         else
152                                 return 0;
153                 } else {
154                         inst->op = (simd)? cvtdq2ps : cvttps2dq;
155                 }
156                 break;

158         case 0x5C:
159                 inst->op = (int)subss + simd + dbl;
160                 break;

162         case 0x5D:
163                 inst->op = (int)minss + simd + dbl;
164                 break;

166         case 0x5E:
167                 inst->op = (int)divss + simd + dbl;
168                 break;

170         case 0x5F:
171                 inst->op = (int)maxss + simd + dbl;
172                 break;

174         case 0xC2:
175                 inst->op = (int)cmpss + simd + dbl;
176                 break;

178         case 0xE6:
179                 if (simd) {
180                         if (dbl)
181                                 inst->op = cvttpd2dq;
182                         else
183                                 return 0;
184                 } else {
185                         inst->op = (dbl)? cvtpd2dq : cvtdq2pd;
186                 }
187                 break;

189         default:
190                 return 0;
191         }
```



```
193         /* locate operands */
194         modrm = ip[i++];

196         if (inst->op == cvtss2si || inst->op == cvttss2si ||
197             inst->op == cvtsd2si || inst->op == cvttsd2si ||
198             inst->op == cvtss2siq || inst->op == cvttss2siq ||
199             inst->op == cvtsd2siq || inst->op == cvttsd2siq) {
200                 /* op1 is a gp register */
201                 r = ((rex & 4) << 1) | ((modrm >> 3) & 7);
202                 inst->op1 = (sseoperand_t *)&uap->uc_mcontext.gregs[regno(r)];
203         } else if (inst->op == cvtps2pi || inst->op == cvttps2pi ||
204             inst->op == cvtpd2pi || inst->op == cvttpd2pi) {
205                 /* op1 is a mmx register */
206 #ifdef __amd64
207                 inst->op1 = (sseoperand_t *)&uap->uc_mcontext.fpregs.fp_reg_set.
208                     fpchip_state.st[(modrm >> 3) & 7];
209 #else
210                 inst->op1 = (sseoperand_t *)(10 * ((modrm >> 3) & 7) +
211                     (char *)&uap->uc_mcontext.fpregs.fp_reg_set.
212                     fpchip_state.state[7]);
213 #endif
214         } else {
215                 /* op1 is a xmm register */
216                 r = ((rex & 4) << 1) | ((modrm >> 3) & 7);
217                 inst->op1 = (sseoperand_t *)&uap->uc_mcontext.fpregs.
218                     fp_reg_set.fpchip_state.xmm[r];
219         }

221         if ((modrm >> 6) == 3) {
222                 if (inst->op == cvtsi2ss || inst->op == cvtsi2sd ||
223                     inst->op == cvtsi2ssq || inst->op == cvtsi2sdq) {
224                         /* op2 is a gp register */
225                         r = ((rex & 1) << 3) | (modrm & 7);
226                         inst->op2 = (sseoperand_t *)&uap->uc_mcontext.
227                             gregs[regno(r)];
228                 } else if (inst->op == cvtpi2ps || inst->op == cvtpi2pd) {
229                         /* op2 is a mmx register */
230 #ifdef __amd64
231                         inst->op2 = (sseoperand_t *)&uap->uc_mcontext.fpregs.
232                             fp_reg_set.fpchip_state.st[modrm & 7];
233 #else
234                         inst->op2 = (sseoperand_t *)(10 * (modrm & 7) +
235                             (char *)&uap->uc_mcontext.fpregs.fp_reg_set.
236                             fpchip_state.state[7]);
237 #endif
238                 } else {
239                         /* op2 is a xmm register */
240                         r = ((rex & 1) << 3) | (modrm & 7);
241                         inst->op2 = (sseoperand_t *)&uap->uc_mcontext.fpregs.
242                             fp_reg_set.fpchip_state.xmm[r];
243                 }
244         } else if ((modrm & 0xc7) == 0x05) {
245 #ifdef __amd64
246                 /* address of next instruction + offset */
247                 r = i + 4;
248                 if (inst->op == cmpss || inst->op == cmpps ||
249                     inst->op == cmpsd || inst->op == cmppd)
250                         r++;
251                 inst->op2 = (sseoperand_t *)(ip + r + *(int *)(ip + i));
252 #else
253                 /* absolute address */
254                 inst->op2 = (sseoperand_t *)(*(int *)(ip + i));
255 #endif
256                 i += 4;
257         } else {
258                 /* complex address */
```

```
259                 if ((modrm & 7) == 4) {
260                         /* parse sib byte */
261                         sib = ip[i++];
262                         if ((sib & 7) == 5 && (modrm >> 6) == 0) {
263                                 /* start with absolute address */
264                                 addr = (char *)(uintptr_t)(*(int *)(ip + i));
265                                 i += 4;
266                         } else {
267                                 /* start with base */
268                                 r = ((rex & 1) << 3) | (sib & 7);
269                                 addr = (char *)uap->uc_mcontext.gregs[regno(r)];
270                         }
271                         r = ((rex & 2) << 2) | ((sib >> 3) & 7);
272                         if (r != 4) {
273                                 /* add scaled index */
274                                 addr += uap->uc_mcontext.gregs[regno(r)]
275                                     << (sib >> 6);
276                         }
277                 } else {
278                         r = ((rex & 1) << 3) | (modrm & 7);
279                         addr = (char *)uap->uc_mcontext.gregs[regno(r)];
280                 }

282                 /* add displacement, if any */
283                 if ((modrm >> 6) == 1) {
284                         addr += (char)ip[i++];
285                 } else if ((modrm >> 6) == 2) {
286                         addr += *(int *)(ip + i);
287                         i += 4;
288                 }
289                 inst->op2 = (sseoperand_t *)addr;
290         }

292         if (inst->op == cmpss || inst->op == cmpps || inst->op == cmpsd ||
293             inst->op == cmppd) {
294                 /* get the immediate operand */
295                 inst->imm = ip[i++];
296         }

298         return i;
299 }
```

unchanged_portion_omitted_

```
**********************************************************
    7672 Tue Nov  4 19:01:30 2014
new/usr/src/lib/libm/common/m9x/__fex_sym.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #include "fenv_synonyms.h"
  30 #include <elf.h>
  31 #include <stdio.h>
  32 #include <stdlib.h>
  33 #include <unistd.h>
  34 #include <fcntl.h>
  35 #include <procfs.h>
  36 #include <string.h>
  37 #include <sys/stat.h>

  39 #if defined(__sparcv9) || defined(__amd64)

  41 #define Elf_Ehdr        Elf64_Ehdr
  42 #define Elf_Phdr        Elf64_Phdr
  43 #define Elf_Shdr        Elf64_Shdr
  44 #define Elf_Sym         Elf64_Sym
  45 #define ELF_ST_BIND     ELF64_ST_BIND
  46 #define ELF_ST_TYPE     ELF64_ST_TYPE

  48 #else

  50 #define Elf_Ehdr        Elf32_Ehdr
  51 #define Elf_Phdr        Elf32_Phdr
  52 #define Elf_Shdr        Elf32_Shdr
  53 #define Elf_Sym         Elf32_Sym
  54 #define ELF_ST_BIND     ELF32_ST_BIND
  55 #define ELF_ST_TYPE     ELF32_ST_TYPE

  57 #endif  /* __sparcv9 */

  59 /* semi-permanent data established by __fex_sym_init */
  60 static  prmap_t         *pm = NULL;             /* prmap_t array */
```



```
  61 static  int                     npm = 0;                /* number of entries in

  63 /* transient data modified by __fex_sym */
  64 static  prmap_t         *lpm = NULL;    /* prmap_t found in last call */
  65 static  Elf_Phdr        *ph = NULL;             /* program header array */
  66 static  int                     phsize = 0;             /* size of ph */
  67 static  int                     nph;                    /* number of entries in
  68 static  char            *stbuf = NULL;  /* symbol and string table buffer */
  69 static  int                     stbufsize = 0;  /* size of stbuf */
  70 static  int                     stoffset;               /* offset of string tabl
  71 static  int                     nsyms;                  /* number of symbols in

  73 /* get a current prmap_t list (must call this before each stack trace) */
  74 void
  75 __fex_sym_init()
  76 {
  77         struct stat     statbuf;
  78         long            n;
  79         int                     i;

  81         /* clear out the previous prmap_t list */
  82         if (pm != NULL)
  83                 free(pm);
  84         pm = lpm = NULL;
  85         npm = 0;

  87         /* get the current prmap_t list */
  88         if (stat("/proc/self/map", &statbuf) < 0 || statbuf.st_size <= 0 ||
  89                 (pm = (prmap_t*)malloc(statbuf.st_size)) == NULL)
  90                 return;
  91         if ((i = open("/proc/self/map", O_RDONLY)) < 0)
  92         {
  93                 free(pm);
  94                 pm = NULL;
  95                 return;
  96         }
  97         n = read(i, pm, statbuf.st_size);
  98         close(i);
  99         if (n != statbuf.st_size)
 100         {
 101                 free(pm);
 102                 pm = NULL;
 103         }
 104         else
 105                 npm = (int) (n / sizeof(prmap_t));
 106 }
______unchanged_portion_omitted_
```

```
**********************************************************
    3206 Tue Nov  4 19:01:31 2014
new/usr/src/lib/libm/common/m9x/feexcept.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __feclearexcept = feclearexcept
#pragma weak __feraiseexcept = feraiseexcept
#pragma weak __fetestexcept = fetestexcept
#pragma weak __fegetexceptflag = fegetexceptflag
#pragma weak __fesetexceptflag = fesetexceptflag

#pragma weak feclearexcept96 = feclearexcept
#pragma weak feraiseexcept96 = feraiseexcept
#pragma weak fetestexcept96 = fetestexcept
#pragma weak fegetexceptflag96 = fegetexceptflag
#pragma weak fesetexceptflag96 = fesetexceptflag
#pragma weak feclearexcept = __feclearexcept
#pragma weak feraiseexcept = __feraiseexcept
#pragma weak fetestexcept = __fetestexcept
#pragma weak fegetexceptflag = __fegetexceptflag
#pragma weak fesetexceptflag = __fesetexceptflag

#pragma weak feclearexcept96 = __feclearexcept
#pragma weak feraiseexcept96 = __feraiseexcept
#pragma weak fetestexcept96 = __fetestexcept
#pragma weak fegetexceptflag96 = __fegetexceptflag
#pragma weak fesetexceptflag96 = __fesetexceptflag

#include "fenv_synonyms.h"
#include <fenv.h>
#include <sys/ieeefp.h>
#include <ucontext.h>
#include <thread.h>
#include "fex_handler.h"
#include "fenv_inlines.h"
```



```
int feclearexcept(int e)
{
        unsigned long   fsr;

        __fenv_getfsr(&fsr);
        __fenv_set_ex(fsr, __fenv_get_ex(fsr) & ~e);
        __fenv_setfsr(&fsr);
        if (fex_get_log())
                __fex_update_te();
        return 0;
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    2813 Tue Nov  4 19:01:31 2014
new/usr/src/lib/libm/common/m9x/fenv.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fex_merge_flags = fex_merge_flags
#pragma weak fex_merge_flags = __fex_merge_flags

#pragma weak __feholdexcept = feholdexcept
#pragma weak __feupdateenv = feupdateenv
#pragma weak __fegetenv = fegetenv
#pragma weak __fesetenv = fesetenv

#pragma weak feupdateenv96 = feupdateenv
#pragma weak fegetenv96 = fegetenv
#pragma weak fesetenv96 = fesetenv
#pragma weak feholdexcept = __feholdexcept
#pragma weak feupdateenv = __feupdateenv
#pragma weak fegetenv = __fegetenv
#pragma weak fesetenv = __fesetenv

#pragma weak feholdexcept96 = __feholdexcept96
#pragma weak feupdateenv96 = __feupdateenv
#pragma weak fegetenv96 = __fegetenv
#pragma weak fesetenv96 = __fesetenv

#include "fenv_synonyms.h"
#include <fenv.h>
#include <ucontext.h>
#include <thread.h>
#include "fex_handler.h"
#include "fenv_inlines.h"

const fenv_t __fenv_dfl_env = {
        {
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
```



```
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
                { FEX_NONSTOP, (void(*)())0 },
        },
#ifdef __x86
        0x13000000
#else
        0
#endif
};
______unchanged_portion_omitted_
```

****
1452 Tue Nov 4 19:01:32 2014
new/usr/src/lib/libm/common/m9x/feprec.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fegetprec = fegetprec
#pragma weak __fesetprec = fesetprec
#pragma weak fegetprec = __fegetprec
#pragma weak fesetprec = __fesetprec

#include "fenv_synonyms.h"
#include <fenv.h>
#include <ucontext.h>
#include <thread.h>
#include "fex_handler.h"

int fegetprec(void)
{
        unsigned long   fsr;

        __fenv_getfsr(&fsr);
        return __fenv_get_rp(fsr);
}
```

*_____unchanged_portion_omitted_*

****
1936 Tue Nov 4 19:01:32 2014
new/usr/src/lib/libm/common/m9x/feround.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fegetround = fegetround
#pragma weak __fesetround = fesetround
#pragma weak fegetround = __fegetround
#pragma weak fesetround = __fesetround

#pragma weak fegetround96 = __fegetround
#pragma weak fesetround96 = __fesetround96

#include "fenv_synonyms.h"
#include <fenv.h>
#include <ucontext.h>
#include <thread.h>
#include "fex_handler.h"
#include "fenv_inlines.h"

#if defined(__i386) && !defined(__amd64)
#include <float.h>
#endin

int fegetround(void)
{
        unsigned long   fsr;

        __fenv_getfsr(&fsr);
        return (int)__fenv_get_rd(fsr);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2429 Tue Nov  4 19:01:33 2014
new/usr/src/lib/libm/common/m9x/fex_handler.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fex_get_handling = fex_get_handling
#pragma weak __fex_set_handling = fex_set_handling
#pragma weak __fex_getexcepthandler = fex_getexcepthandler
#pragma weak __fex_setexcepthandler = fex_setexcepthandler
#pragma weak fex_get_handling = __fex_get_handling
#pragma weak fex_set_handling = __fex_set_handling
#pragma weak fex_getexcepthandler = __fex_getexcepthandler
#pragma weak fex_setexcepthandler = __fex_setexcepthandler

#include "fenv_synonyms.h"
#include <fenv.h>
#include <ucontext.h>
#include <thread.h>
#include "fex_handler.h"

int fex_get_handling(int e)
{
        struct fex_handler_data *thr_handlers;
        int                                     i;

        thr_handlers = __fex_get_thr_handlers();
        for (i = 0; i < FEX_NUM_EXC; i++)
                if (e & (1 << i))
                        return thr_handlers[i].__mode;
        return FEX_NOHANDLER;
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9349 Tue Nov  4 19:01:34 2014
new/usr/src/lib/libm/common/m9x/fex_log.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __fex_get_log = fex_get_log
#pragma weak __fex_set_log = fex_set_log
#pragma weak __fex_get_log_depth = fex_get_log_depth
#pragma weak __fex_set_log_depth = fex_set_log_depth
#pragma weak __fex_log_entry = fex_log_entry
#pragma weak fex_get_log = __fex_get_log
#pragma weak fex_set_log = __fex_set_log
#pragma weak fex_get_log_depth = __fex_get_log_depth
#pragma weak fex_set_log_depth = __fex_set_log_depth
#pragma weak fex_log_entry = __fex_log_entry

#include "fenv_synonyms.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>
#include <ucontext.h>
#include <sys/frame.h>
#include <fenv.h>
#include <sys/ieeefp.h>
#include <thread.h>
#include "fex_handler.h"

#if !defined(PC)
#if defined(REG_PC)
#define PC      REG_PC
#else
#error Neither PC nor REG_PC is defined!
#endif
#endif
```



```
static FILE *log_fp = NULL;
static mutex_t log_lock = DEFAULTMUTEX;
static int log_depth = 100;

FILE *fex_get_log(void)
{
        FILE    *fp;

        mutex_lock(&log_lock);
        fp = log_fp;
        mutex_unlock(&log_lock);
        return fp;
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1614 Tue Nov  4 19:01:34 2014
new/usr/src/lib/libm/common/m9x/ldexp.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ldexp = ldexp
#pragma weak ldexp = __ldexp

#include "libm.h"
#include <errno.h>

double
ldexp(double x, int n) {
        int *px = (int *) &x, ix = px[HIWORD] & ~0x80000000;

        if (ix >= 0x7ff00000 || (px[LOWORD] | ix) == 0)
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (ix >= 0x7ff80000 ? x : x + x);
                /* assumes sparc-like QNaN */
#else
                return (x + x);
#endif
        x = scalbn(x, n);
        ix = px[HIWORD] & ~0x80000000;
        /*
         * SVID3 requires both overflow and underflow cases to set errno
         * XPG3/XPG4/XPG4.2/SUSv2 requires overflow to set errno
         */
        if (ix >= 0x7ff00000 || (px[LOWORD] | ix) == 0)
                errno = ERANGE;
        return (x);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1106 Tue Nov  4 19:01:35 2014
new/usr/src/lib/libm/common/m9x/ldexpf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ldexpf = ldexpf
#pragma weak ldexpf = __ldexpf

#include "libm.h"

float
ldexpf(float x, int n) {
        return (scalbnf(x, n));
}
```

_____***unchanged_portion_omitted_***

****
1118 Tue Nov 4 19:01:35 2014
new/usr/src/lib/libm/common/m9x/ldexpl.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __ldexpl = ldexpl
#pragma weak ldexpl = __ldexpl

#include "libm.h"

long double
ldexpl(long double x, int n) {
        return (scalbnl(x, n));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2280 Tue Nov  4 19:01:36 2014
new/usr/src/lib/libm/common/m9x/llrint.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llrint = llrint
#pragma weak llrint = __llrint
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lrint = llrint
#pragma weak __lrint = llrint
#pragma weak lrint = __llrint
#pragma weak __lrint = __llrint
#endif

/*
 * llrint(x) rounds its argument to the nearest integer according
 * to the current rounding direction and converts the result to a
 * 64 bit signed integer.
 *
 * If x is NaN, infinite, or so large that the nearest integer would
 * exceed 64 bits, the invalid operation exception is raised.  If x
 * is not an integer, the inexact exception is raised.
 */

#include "libm.h"

long long
llrint(double x) {
        /*
         * Note: The following code works on x86 (in the default rounding
         * precision mode), but one should just use the fistpll instruction
         * instead.
         */
        union {
                unsigned i[2];
                double d;
        } xx, yy;
```



```
        unsigned hx;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;

        if (hx < 0x43300000) { /* |x| < 2^52 */
                /* add and subtract a power of two to round x to an integer */
#if defined(__sparc) || defined(__amd64)
                yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43300000;
#elif defined(__i386)   /* !defined(__amd64) */
                yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43e00000;
#else
#error Unknown architecture
#endif
                yy.i[LOWORD] = 0;
                x = (x + yy.d) - yy.d;
        }

        /* now x is nan, inf, or integral */
        return ((long long) x);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1996 Tue Nov  4 19:01:36 2014
new/usr/src/lib/libm/common/m9x/llrintf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llrintf = llrintf
#pragma weak llrintf = __llrintf
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lrintf = llrintf
#pragma weak __lrintf = llrintf
#pragma weak lrintf = __llrintf
#pragma weak __lrintf = __llrintf
#endif

#include "libm.h"

long long
llrintf(float x) {
        /*
         * Note: The following code works on x86 (in the default rounding
         * precision mode), but one should just use the fistpll instruction
         * instead.
         */
        union {
                unsigned i;
                float f;
        } xx, yy;
        unsigned hx;

        xx.f = x;
        hx = xx.i & ~0x80000000;

        if (hx < 0x4b000000) { /* |x| < 2^23 */
                /* add and subtract a power of two to round x to an integer */
#if defined(__sparc) || defined(__amd64)
                yy.i = (xx.i & 0x80000000) | 0x4b000000;
#elif defined(__i386)
```



```
                /* assume 64-bit precision */
                yy.i = (xx.i & 0x80000000) | 0x5f000000;
#else
#error Unknown architecture
#endif
                x = (x + yy.f) - yy.f;

                /*
                 * on LP32 architectures, we can just convert x to a 32-bit
                 * integer and sign-extend it
                 */
                return ((long) x);
        }

        /* now x is nan, inf, or integral */
        return ((long long) x);
}
```

unchanged_portion_omitted_

```
**********************************************************
    4313 Tue Nov  4 19:01:37 2014
new/usr/src/lib/libm/common/m9x/llrintl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llrintl = llrintl
#pragma weak llrintl = __llrintl
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lrintl = llrintl
#pragma weak __lrintl = llrintl
#pragma weak lrintl = __llrintl
#pragma weak __lrintl = __llrintl
#endif

#include "libm.h"

#if defined(__sparc)

#include "fma.h"
#include "fenv_inlines.h"

long long
llrintl(long double x) {
        union {
                unsigned i[4];
                long double q;
        } xx;
        union {
                unsigned i[2];
                long long l;
        } zz;
        union {
                unsigned i;
                float f;
        } tt;
        unsigned int hx, sx, frac, fsr;
        int rm, j;
```



```
        volatile float dummy;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;

        /* handle trivial cases */
        if (hx > 0x403e0000) { /* |x| > 2^63 + ... or x is nan */
                /* convert an out-of-range float */
                tt.i = sx | 0x7f000000;
                return ((long long) tt.f);
        } else if ((hx | xx.i[1] | xx.i[2] | xx.i[3]) == 0) /* x is zero */
                return (0LL);

        /* get the rounding mode */
        __fenv_getfsr32(&fsr);
        rm = fsr >> 30;

        /* flip the sense of directed roundings if x is negative */
        if (sx)
                rm ^= rm >> 1;

        /* handle |x| < 1 */
        if (hx < 0x3fff0000) {
                dummy = 1.0e30f; /* x is nonzero, so raise inexact */
                dummy += 1.0e-30f;
                if (rm == FSR_RP || (rm == FSR_RN && (hx >= 0x3ffe0000 &&
                        ((hx & 0xffff) | xx.i[1] | xx.i[2] | xx.i[3]))))
                        return (sx ? -1LL : 1LL);
                return (0LL);
        }

        /* extract the integer and fractional parts of x */
        j = 0x406f - (hx >> 16);
        xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
        if (j >= 96) {
                zz.i[0] = 0;
                zz.i[1] = xx.i[0] >> (j - 96);
                frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
                if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
                        frac |= 1;
        } else if (j >= 64) {
                zz.i[0] = xx.i[0] >> (j - 64);
                zz.i[1] = ((xx.i[0] << 1) << (95 - j)) | (xx.i[1] >> (j - 64));
                frac = ((xx.i[1] << 1) << (95 - j)) | (xx.i[2] >> (j - 64));
                if (((xx.i[2] << 1) << (95 - j)) | xx.i[3])
                        frac |= 1;
        } else {
                zz.i[0] = ((xx.i[0] << 1) << (63 - j)) | (xx.i[1] >> (j - 32));
                zz.i[1] = ((xx.i[1] << 1) << (63 - j)) | (xx.i[2] >> (j - 32));
                frac = ((xx.i[2] << 1) << (63 - j)) | (xx.i[3] >> (j - 32));
                if ((xx.i[3] << 1) << (63 - j))
                        frac |= 1;
        }

        /* round */
        if (frac && (rm == FSR_RP || (rm == FSR_RN && (frac > 0x80000000u ||
                (frac == 0x80000000 && (zz.i[1] & 1)))))) {
                if (++zz.i[1] == 0)
                        zz.i[0]++;
        }

        /* check for result out of range (note that z is |x| at this point) */
        if (zz.i[0] > 0x80000000u || (zz.i[0] == 0x80000000 && (zz.i[1] ||
                !sx))) {
                tt.i = sx | 0x7f000000;
```

```
 125                 return ((long long) tt.f);
 126         }

 128         /* raise inexact if need be */
 129         if (frac) {
 130                 dummy = 1.0e30F;
 131                 dummy += 1.0e-30F;
 132         }

 134         /* negate result if need be */
 135         if (sx) {
 136                 zz.i[0] = ~zz.i[0];
 137                 zz.i[1] = -zz.i[1];
 138                 if (zz.i[1] == 0)
 139                         zz.i[0]++;
 140         }
 141         return (zz.l);
 142 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2185 Tue Nov  4 19:01:37 2014
new/usr/src/lib/libm/common/m9x/llround.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llround = llround
#pragma weak llround = __llround
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lround = llround
#pragma weak __lround = llround
#pragma weak lround = __llround
#pragma weak __lround = __llround
#endif

/*
 * llround(x) rounds its argument to the nearest integer, rounding
 * ties away from zero, and converts the result to a 64 bit signed
 * integer.
 *
 * If x is NaN, infinite, or so large that the nearest integer
 * would exceed 64 bits, the invalid operation exception is raised.
 */

#include "libm.h"

long long
llround(double x) {
        union {
                unsigned i[2];
                double d;
        } xx;
        unsigned hx, sx, i;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;
        sx = xx.i[HIWORD] & 0x80000000;
```



```
        if (hx < 0x43300000) { /* |x| < 2^52 */
                /* handle |x| < 1 */
                if (hx < 0x3ff00000) {
                        if (hx >= 0x3fe00000)
                                return (sx ? -1LL : 1LL);
                        return (0LL);
                }

                /* round x at the integer bit */
                if (hx < 0x41300000) {
                        i = 1 << (0x412 - (hx >> 20));
                        xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
                        xx.i[LOWORD] = 0;
                } else {
                        i = 1 << (0x432 - (hx >> 20));
                        xx.i[LOWORD] += i;
                        if (xx.i[LOWORD] < i)
                                xx.i[HIWORD]++;
                        xx.i[LOWORD] &= ~(i | (i - 1));
                }
        }

        /* now x is nan, inf, or integral */
        return ((long long) xx.d);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1795 Tue Nov  4 19:01:37 2014
new/usr/src/lib/libm/common/m9x/llroundf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llroundf = llroundf
#pragma weak llroundf = __llroundf
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lroundf = llroundf
#pragma weak __lroundf = llroundf
#pragma weak lroundf = __llroundf
#pragma weak __lroundf = __llroundf
#endif

#include "libm.h"

long long
llroundf(float x) {
        union {
                unsigned i;
                float f;
        } xx;
        unsigned hx, sx, i;

        xx.f = x;
        hx = xx.i & ~0x80000000;
        sx = xx.i & 0x80000000;

        if (hx < 0x4b000000) { /* |x| < 2^23 */
                /* handle |x| < 1 */
                if (hx < 0x3f800000) {
                        if (hx >= 0x3f000000)
                                return (sx ? -1LL : 1LL);
                        return (0LL);
                }

                /* round x at the integer bit */
```



```
                i = 1 << (0x95 - (hx >> 23));
                xx.i = (xx.i + i) & ~((i << 1) - 1);

                /*
                 * on LP32 architectures, we can just convert x to a 32-bit
                 * integer and sign-extend it
                 */
                return ((long) xx.f);
        }

        /* now x is nan, inf, or integral */
        return ((long long) x);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    3819 Tue Nov  4 19:01:38 2014
new/usr/src/lib/libm/common/m9x/llroundl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __llroundl = llroundl
#pragma weak llroundl = __llroundl
#if defined(__sparcv9) || defined(__amd64)
#pragma weak lroundl = llroundl
#pragma weak __lroundl = llroundl
#pragma weak lroundl = __llroundl
#pragma weak __lroundl = __llroundl
#endif

#include "libm.h"

#if defined(__sparc)
long long
llroundl(long double x) {
        union {
                unsigned i[4];
                long double q;
        } xx;
        union {
                unsigned i[2];
                long long l;
        } zz;
        union {
                unsigned i;
                float f;
        } tt;
        unsigned hx, sx, frac;
        int j;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;
```



```
        /* handle trivial cases */
        if (hx > 0x403e0000) { /* |x| > 2^63 + ... or x is nan */
                /* convert an out-of-range float */
                tt.i = sx | 0x7f000000;
                return ((long long) tt.f);
        }

        /* handle |x| < 1 */
        if (hx < 0x3fff0000) {
                if (hx >= 0x3ffe0000)
                        return (sx ? -1LL : 1LL);
                return (0LL);
        }

        /* extract the integer and fractional parts of x */
        j = 0x406f - (hx >> 16);
        xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
        if (j >= 96) {
                zz.i[0] = 0;
                zz.i[1] = xx.i[0] >> (j - 96);
                frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
                if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
                        frac |= 1;
        } else if (j >= 64) {
                zz.i[0] = xx.i[0] >> (j - 64);
                zz.i[1] = ((xx.i[0] << 1) << (95 - j)) | (xx.i[1] >> (j - 64));
                frac = ((xx.i[1] << 1) << (95 - j)) | (xx.i[2] >> (j - 64));
                if (((xx.i[2] << 1) << (95 - j)) | xx.i[3])
                        frac |= 1;
        } else {
                zz.i[0] = ((xx.i[0] << 1) << (63 - j)) | (xx.i[1] >> (j - 32));
                zz.i[1] = ((xx.i[1] << 1) << (63 - j)) | (xx.i[2] >> (j - 32));
                frac = ((xx.i[2] << 1) << (63 - j)) | (xx.i[3] >> (j - 32));
                if ((xx.i[3] << 1) << (63 - j))
                        frac |= 1;
        }

        /* round */
        if (frac >= 0x80000000u) {
                if (++zz.i[1] == 0)
                        zz.i[0]++;
        }

        /* check for result out of range (note that z is |x| at this point) */
        if (zz.i[0] > 0x80000000u || (zz.i[0] == 0x80000000 && (zz.i[1] ||
            !sx))) {
                tt.i = sx | 0x7f000000;
                return ((long long) tt.f);
        }

        /* negate result if need be */
        if (sx) {
                zz.i[0] = ~zz.i[0];
                zz.i[1] = -zz.i[1];
                if (zz.i[1] == 0)
                        zz.i[0]++;
        }

        return (zz.l);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2236 Tue Nov  4 19:01:39 2014
new/usr/src/lib/libm/common/m9x/lrint.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lrint = lrint
#pragma weak lrint = __lrint

/*
 * lrint(x) rounds its argument to the nearest integer according
 * to the current rounding direction and converts the result to
 * a 32 bit signed integer.
 *
 * If x is NaN, infinite, or so large that the nearest integer
 * would exceed 32 bits, the invalid operation exception is raised.
 * If x is not an integer, the inexact exception is raised.
 */

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
long
lrint(double x) {
        /*
         * Note: The following code works on x86 (in the default rounding
         * precision mode), but one should just use the fistpl instruction
         * instead.
         */
        union {
                unsigned i[2];
                double d;
        } xx, yy;
        unsigned hx;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;
```



```
        if (hx < 0x43300000) { /* |x| < 2^52 */
                /* add and subtract a power of two to round x to an integer */
#if defined(__sparc)
                yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43300000;
#elif defined(__x86)
                yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43e00000;
#else
#error Unknown architecture
#endif
                yy.i[LOWORD] = 0;
                x = (x + yy.d) - yy.d;
        }

        /* now x is nan, inf, or integral */
        return ((long) x);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    1868 Tue Nov  4 19:01:39 2014
new/usr/src/lib/libm/common/m9x/lrintf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lrintf = lrintf
#pragma weak lrintf = __lrintf

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
long
lrintf(float x) {
        /*
         * Note: The following code works on x86 (in the default rounding
         * precision mode), but one should just use the fistpl instruction
         * instead.
         */
        union {
                unsigned i;
                float f;
        } xx, yy;
        unsigned hx;

        xx.f = x;
        hx = xx.i & ~0x80000000;
        if (hx < 0x4b000000) {  /* |x| < 2^23 */
                /* add and subtract a power of two to round x to an integer */
#if defined(__sparc)
                yy.i = (xx.i & 0x80000000) | 0x4b000000;
#elif defined(__x86)
                /* assume 64-bit precision */
                yy.i = (xx.i & 0x80000000) | 0x5f000000;
#else
#error Unknown architecture
#endif
```



```
                x = (x + yy.f) - yy.f;
                return ((long) x);
        }

        /* now x is nan, inf, or integral */
        return ((long) x);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    3883 Tue Nov  4 19:01:39 2014
new/usr/src/lib/libm/common/m9x/lrintl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lrintl = lrintl
#pragma weak lrintl = __lrintl

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
#if defined(__sparc)

#include "fma.h"
#include "fenv_inlines.h"

long
lrintl(long double x) {
        union {
                unsigned int i[4];
                long double q;
        } xx;
        union {
                unsigned int i;
                float f;
        } tt;
        unsigned int hx, sx, frac, l, fsr;
        int rm, j;
        volatile float dummy;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;

        /* handle trivial cases */
        if (hx > 0x401e0000) { /* |x| > 2^31 + ... or x is nan */
```



```
                /* convert an out-of-range float */
                tt.i = sx | 0x7f000000;
                return ((long) tt.f);
        } else if ((hx | xx.i[1] | xx.i[2] | xx.i[3]) == 0) /* x is zero */
                return (0L);

        /* get the rounding mode */
        __fenv_getfsr32(&fsr);
        rm = fsr >> 30;

        /* flip the sense of directed roundings if x is negative */
        if (sx)
                rm ^= rm >> 1;

        /* handle |x| < 1 */
        if (hx < 0x3fff0000) {
                dummy = 1.0e30F; /* x is nonzero, so raise inexact */
                dummy += 1.0e-30F;
                if (rm == FSR_RP || (rm == FSR_RN && (hx >= 0x3ffe0000 &&
                        ((hx & 0xffff) | xx.i[1] | xx.i[2] | xx.i[3]))))
                        return (sx ? -1L : 1L);
                return (0L);
        }

        /* extract the integer and fractional parts of x */
        j = 0x406f - (hx >> 16);                /* 91 <= j <= 112 */
        xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
        if (j >= 96) {                          /* 96 <= j <= 112 */
                l = xx.i[0] >> (j - 96);
                frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
                if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
                        frac |= 1;
        } else {                                /* 91 <= j <= 95 */
                l = (xx.i[0] << (96 - j)) | (xx.i[1] >> (j - 64));
                frac = (xx.i[1] << (96 - j)) | (xx.i[2] >> (j - 64));
                if ((xx.i[2] << (96 - j)) | xx.i[3])
                        frac |= 1;
        }

        /* round */
        if (frac && (rm == FSR_RP || (rm == FSR_RN && (frac > 0x80000000U ||
                (frac == 0x80000000 && (l & 1))))))
                l++;

        /* check for result out of range (note that z is |x| at this point) */
        if (l > 0x80000000U || (l == 0x80000000U && !sx)) {
                tt.i = sx | 0x7f000000;
                return ((long) tt.f);
        }

        /* raise inexact if need be */
        if (frac) {
                dummy = 1.0e30F;
                dummy += 1.0e-30F;
        }

        /* negate result if need be */
        if (sx)
                l = -l;
        return ((long) l);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2169 Tue Nov  4 19:01:40 2014
new/usr/src/lib/libm/common/m9x/lround.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lround = lround
#pragma weak lround = __lround

/*
 * lround(x) rounds its argument to the nearest integer, rounding ties
 * away from zero, and converts the result to a 32 bit signed integer.
 *
 * If x is NaN, infinite, or so large that the nearest integer
 * would exceed 32 bits, the invalid operation exception is raised.
 */

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
long
lround(double x) {
        union {
                unsigned i[2];
                double d;
        } xx;
        unsigned hx, sx, i;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;
        sx = xx.i[HIWORD] & 0x80000000;
        if (hx < 0x43300000) {  /* |x| < 2^52 */
                if (hx < 0x3ff00000) {  /* |x| < 1 */
                        if (hx >= 0x3fe00000)
                                return (sx ? -1L : 1L);
                        return (0L);
                }
```



```
                /* round x at the integer bit */
                if (hx < 0x41300000) {
                        i = 1 << (0x412 - (hx >> 20));
                        xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
                        xx.i[LOWORD] = 0;
                } else {
                        i = 1 << (0x432 - (hx >> 20));
                        xx.i[LOWORD] += i;
                        if (xx.i[LOWORD] < i)
                                xx.i[HIWORD]++;
                        xx.i[LOWORD] &= ~(i | (i - 1));
                }
        }

        /* now x is nan, inf, or integral */
        return ((long) xx.d);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    1675 Tue Nov  4 19:01:40 2014
new/usr/src/lib/libm/common/m9x/lroundf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lroundf = lroundf
#pragma weak lroundf = __lroundf

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
long
lroundf(float x) {
        union {
                unsigned i;
                float f;
        } xx;
        unsigned hx, sx, i;

        xx.f = x;
        hx = xx.i & ~0x80000000;
        sx = xx.i & 0x80000000;
        if (hx < 0x4b000000) {          /* |x| < 2^23 */
                if (hx < 0x3f800000) {          /* |x| < 1 */
                        if (hx >= 0x3f000000)
                                return (sx ? -1L : 1L);
                        return (0L);
                }

                /* round x at the integer bit */
                i = 1 << (0x95 - (hx >> 23));
                xx.i = (xx.i + i) & ~((i << 1) - 1);
                return ((long) xx.f);
        }

        /* now x is nan, inf, or integral */
```



```
        return ((long) x);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    3372 Tue Nov  4 19:01:41 2014
new/usr/src/lib/libm/common/m9x/lroundl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __lroundl = lroundl
#pragma weak lroundl = __lroundl

#include <sys/isa_defs.h>       /* _ILP32 */
#include "libm.h"

#if defined(_ILP32)
#if defined(__sparc)
long
lroundl(long double x) {
        union {
                unsigned i[4];
                long double q;
        } xx;
        union {
                unsigned i;
                float f;
        } tt;
        unsigned hx, sx, frac, l;
        int j;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;

        /* handle trivial cases */
        if (hx > 0x401e0000) { /* |x| > 2^31 + ... or x is nan */
                /* convert an out-of-range float */
                tt.i = sx | 0x7f000000;
                return ((long) tt.f);
        }
```



```
        /* handle |x| < 1 */
        if (hx < 0x3fff0000) {
                if (hx >= 0x3ffe0000)
                        return (sx ? -1L : 1L);
                return (0L);
        }

        /* extract the integer and fractional parts of x */
        j = 0x406f - (hx >> 16);                /* 91 <= j <= 112 */
        xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
        if (j >= 96) {                          /* 96 <= j <= 112 */
                l = xx.i[0] >> (j - 96);
                frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
                if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
                        frac |= 1;
        } else {                                /* 91 <= j <= 95 */
                l = (xx.i[0] << (96 - j)) | (xx.i[1] >> (j - 64));
                frac = (xx.i[1] << (96 - j)) | (xx.i[2] >> (j - 64));
                if ((xx.i[2] << (96 - j)) | xx.i[3])
                        frac |= 1;
        }

        /* round */
        if (frac >= 0x80000000U)
                l++;

        /* check for result out of range (note that z is |x| at this point) */
        if (l > 0x80000000U || (l == 0x80000000U && !sx)) {
                tt.i = sx | 0x7f000000;
                return ((long) tt.f);
        }

        /* negate result if need be */
        if (sx)
                l = -l;
        return ((long) l);
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3162 Tue Nov  4 19:01:41 2014
new/usr/src/lib/libm/common/m9x/nearbyint.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak nearbyint = __nearbyint

/*
 * nearbyint(x) returns the nearest fp integer to x in the direction
 * corresponding to the current rounding direction without raising
 * the inexact exception.
 *
 * nearbyint(x) is x unchanged if x is +/-0 or +/-inf.  If x is NaN,
 * nearbyint(x) is also NaN.
 */

#include "libm.h"
#include "fenv_synonyms.h"
#include <fenv.h>

double
__nearbyint(double x) {
        union {
                unsigned i[2];
                double d;
        } xx;
        unsigned hx, sx, i, frac;
        int rm, j;

        xx.d = x;
        sx = xx.i[HIWORD] & 0x80000000;
        hx = xx.i[HIWORD] & ~0x80000000;

        /* handle trivial cases */
        if (hx >= 0x43300000) { /* x is nan, inf, or already integral */
                if (hx >= 0x7ff00000)   /* x is inf or nan */
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
```



```
                        return (hx >= 0x7ff80000 ? x : x + x);
                        /* assumes sparc-like QNaN */
#else
                        return (x + x);
#endif
                return (x);
        } else if ((hx | xx.i[LOWORD]) == 0)    /* x is zero */
                return (x);

        /* get the rounding mode */
        rm = fegetround();

        /* flip the sense of directed roundings if x is negative */
        if (sx && (rm == FE_UPWARD || rm == FE_DOWNWARD))
                rm = (FE_UPWARD + FE_DOWNWARD) - rm;

        /* handle |x| < 1 */
        if (hx < 0x3ff00000) {
                if (rm == FE_UPWARD || (rm == FE_TONEAREST &&
                        (hx >= 0x3fe00000 && ((hx & 0xfffff) | xx.i[LOWORD])))
                        xx.i[HIWORD] = sx | 0x3ff00000;
                else
                        xx.i[HIWORD] = sx;
                xx.i[LOWORD] = 0;
                return (xx.d);
        }

        /* round x at the integer bit */
        j = 0x433 - (hx >> 20);
        if (j >= 32) {
                i = 1 << (j - 32);
                frac = ((xx.i[HIWORD] << 1) << (63 - j)) |
                        (xx.i[LOWORD] >> (j - 32));
                if (xx.i[LOWORD] & (i - 1))
                        frac |= 1;
                if (!frac)
                        return (x);
                xx.i[LOWORD] = 0;
                xx.i[HIWORD] &= ~(i - 1);
                if ((rm == FE_UPWARD) || ((rm == FE_TONEAREST) &&
                        ((frac > 0x80000000u) || ((frac == 0x80000000) &&
                        (xx.i[HIWORD] & i)))))
                        xx.i[HIWORD] += i;
        } else {
                i = 1 << j;
                frac = (xx.i[LOWORD] << 1) << (31 - j);
                if (!frac)
                        return (x);
                xx.i[LOWORD] &= ~(i - 1);
                if ((rm == FE_UPWARD) || ((rm == FE_TONEAREST) &&
                        (frac > 0x80000000u || ((frac == 0x80000000) &&
                        (xx.i[LOWORD] & i))))) {
                        xx.i[LOWORD] += i;
                        if (xx.i[LOWORD] == 0)
                                xx.i[HIWORD]++;
                }
        }
        return (xx.d);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    3970 Tue Nov  4 19:01:41 2014
new/usr/src/lib/libm/common/m9x/nearbyintf.c
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak nearbyintf = __nearbyintf

#include "libm.h"
#include "fenv_synonyms.h"
#include <fenv.h>

float
__nearbyintf(float x) {
        union {
                unsigned i;
                float f;
        } xx;
        unsigned hx, sx, i, frac;
        int rm;

        xx.f = x;
        sx = xx.i & 0x80000000;
        hx = xx.i & ~0x80000000;

        /* handle trivial cases */
        if (hx >= 0x4b000000) { /* x is nan, inf, or already integral */
                if (hx > 0x7f800000)    /* x is nan */
                        return (x * x);         /* + -> * for Cheetah */
                return (x);
        } else if (hx == 0)             /* x is zero */
                return (x);

        /* get the rounding mode */
        rm = fegetround();

        /* flip the sense of directed roundings if x is negative */
        if (sx && (rm == FE_UPWARD || rm == FE_DOWNWARD))
```



```
                rm = (FE_UPWARD + FE_DOWNWARD) - rm;

        /* handle |x| < 1 */
        if (hx < 0x3f800000) {
                if (rm == FE_UPWARD || (rm == FE_TONEAREST && hx > 0x3f000000))
                        xx.i = sx | 0x3f800000;
                else
                        xx.i = sx;
                return (xx.f);
        }

        /* round x at the integer bit */
        i = 1 << (0x96 - (hx >> 23));
        frac = hx & (i - 1);
        if (!frac)
                return (x);

        hx &= ~(i - 1);
        if (rm == FE_UPWARD || (rm == FE_TONEAREST && (frac > (i >> 1) ||
                ((frac == (i >> 1)) && (hx & i)))))
                xx.i = sx | (hx + i);
        else
                xx.i = sx | hx;
        return (xx.f);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    2735 Tue Nov  4 19:01:42 2014
new/usr/src/lib/libm/common/m9x/nexttowardl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __nexttowardl = nexttowardl
#pragma weak nexttowardl = __nexttowardl

#include "libm.h"
#include <float.h>              /* LDBL_MAX, LDBL_MIN */

#if defined(__sparc)
#define n0      0
#define n1      1
#define n2      2
#define n3      3
#define X86PDNRM1(x)
#define INC(px) { \
                        if (++px[n3] == 0) \
                                if (++px[n2] == 0) \
                                        if (++px[n1] == 0) \
                                                ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n3] == 0xffffffff) \
                                if (--px[n2] == 0xffffffff) \
                                        if (--px[n1] == 0xffffffff) \
                                                --px[n0]; \
                }
#elif defined(__x86)
#define n0      2
#define n1      1
#define n2      0
#define n3      0
/*
 * if pseudo-denormal, replace by the equivalent normal
 */
```



```
#define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
                                0x80000000) != 0) \
                                ((int *) &x)[2] |= 1
#define INC(px) { \
                        if (++px[n2] == 0) \
                                if ((++px[n1] & ~0x80000000) == 0) \
                                        px[n1] = 0x80000000, ++px[n0]; \
                }
#define DEC(px) { \
                        if (--px[n2] == 0xffffffff) \
                                if (--px[n1] == 0x7fffffff) \
                                        if ((--px[n0] & 0x7fff) != 0) \
                                                px[n1] |= 0x80000000; \
                }
#endif

long double
nexttowardl(long double x, long double y) {
        int *px = (int *) &x;
        int *py = (int *) &y;

        if (x == y)
                return (y);             /* C99 requirement */
        if (x != x || y != y)
                return (x * y);

        if (ISZEROL(x)) {       /* x == 0.0 */
                px[n0] = py[n0] & XSGNMSK;
                px[n1] = px[n2] = 0;
                px[n3] = 1;
        } else {
                X86PDNRM1(x);
                if ((px[n0] & XSGNMSK) == 0) {  /* x > 0.0 */
                        if (x > y)      /* x > y */
                                DEC(px)
                        else
                                INC(px)
                } else {
                        if (x < y)      /* x < y */
                                DEC(px)
                        else
                                INC(px)
                }
        }
#ifndef lint
        {
                volatile long double dummy;
                int k = XBIASED_EXP(x);

                if (k == 0)
                        dummy = LDBL_MIN * copysignl(LDBL_MIN, x);
                else if (k == 0x7fff)
                        dummy = LDBL_MAX * copysignl(LDBL_MAX, x);
        }
#endif
        return (x);
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    5960 Tue Nov  4 19:01:42 2014
new/usr/src/lib/libm/common/m9x/remquo.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remquo = remquo
#pragma weak remquo = __remquo

/* INDENT OFF */
/*
 * double remquo(double x, double y, int *quo) return remainder(x,y) and an
 * integer pointer quo such that *quo = N mod {2**31}, where N is the
 * exact integral part of x/y rounded to nearest even.
 *
 * remquo call internal fmodquo
 */
/* INDENT ON */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>               /* fabs() */
#include <sys/isa_defs.h>

#if defined(_BIG_ENDIAN)
#define HIWORD  0
#define LOWORD  1
#else
#define HIWORD  1
#define LOWORD  0
#endif
#define __HI(x) ((int *) &x)[HIWORD]
#define __LO(x) ((int *) &x)[LOWORD]

static const double one = 1.0, Zero[] = {0.0, -0.0};

static double
```



```
fmodquo(double x, double y, int *quo) {
        int n, hx, hy, hz, ix, iy, sx, sq, i, m;
        unsigned lx, ly, lz;

        hx = __HI(x);           /* high word of x */
        lx = __LO(x);           /* low  word of x */
        hy = __HI(y);           /* high word of y */
        ly = __LO(y);           /* low  word of y */
        sx = hx & 0x80000000;   /* sign of x */
        sq = (hx ^ hy) & 0x80000000;    /* sign of x/y */
        hx ^= sx;               /* |x| */
        hy &= 0x7fffffff;       /* |y| */

        /* purge off exception values */
        *quo = 0;
        if ((hy | ly) == 0 || hx >= 0x7ff00000 ||       /* y=0, or x !finite */
            (hy | ((ly | -ly) >> 31)) > 0x7ff00000)     /* or y is NaN */
                return ((x * y) / (x * y));
        if (hx <= hy) {
                if (hx < hy || lx < ly)
                        return (x);     /* |x|<|y| return x */
                if (lx == ly) {
                        *quo = 1 + (sq >> 30);
                        /* |x|=|y| return x*0 */
                        return (Zero[(unsigned) sx >> 31]);
                }
        }

        /* determine ix = ilogb(x) */
        if (hx < 0x00100000) {  /* subnormal x */
                if (hx == 0) {
                        for (ix = -1043, i = lx; i > 0; i <<= 1)
                                ix -= 1;
                } else {
                        for (ix = -1022, i = (hx << 11); i > 0; i <<= 1)
                                ix -= 1;
                }
        } else
                ix = (hx >> 20) - 1023;

        /* determine iy = ilogb(y) */
        if (hy < 0x00100000) {  /* subnormal y */
                if (hy == 0) {
                        for (iy = -1043, i = ly; i > 0; i <<= 1)
                                iy -= 1;
                } else {
                        for (iy = -1022, i = (hy << 11); i > 0; i <<= 1)
                                iy -= 1;
                }
        } else
                iy = (hy >> 20) - 1023;

        /* set up {hx,lx}, {hy,ly} and align y to x */
        if (ix >= -1022)
                hx = 0x00100000 | (0x000fffff & hx);
        else {                  /* subnormal x, shift x to normal */
                n = -1022 - ix;
                if (n <= 31) {
                        hx = (hx << n) | (lx >> (32 - n));
                        lx <<= n;
                } else {
                        hx = lx << (n - 32);
                        lx = 0;
                }
        }
        if (iy >= -1022)
```

```
126                 hy = 0x00100000 | (0x000fffff & hy);
127         else {                  /* subnormal y, shift y to normal */
128                 n = -1022 - iy;
129                 if (n <= 31) {
130                         hy = (hy << n) | (ly >> (32 - n));
131                         ly <<= n;
132                 } else {
133                         hy = ly << (n - 32);
134                         ly = 0;
135                 }
136         }

137         /* fix point fmod */
138         n = ix - iy;
139         m = 0;
140         while (n--) {
141                 hz = hx - hy;
142                 lz = lx - ly;
143                 if (lx < ly)
144                         hz -= 1;
145                 if (hz < 0) {
146                         hx = hx + hx + (lx >> 31);
147                         lx = lx + lx;
148                 } else {
149                         m += 1;
150                         if ((hz | lz) == 0) {   /* return sign(x)*0 */
151                                 if (n < 31)
152                                         m <<= 1 + n;
153                                 else
154                                         m = 0;
155                                 m &= 0x7fffffff;
156                                 *quo = sq >= 0 ? m : -m;
157                                 return (Zero[(unsigned) sx >> 31]);
158                         }
159                         hx = hz + hz + (lz >> 31);
160                         lx = lz + lz;
161                 }
162                 m += m;
163         }
164         hz = hx - hy;
165         lz = lx - ly;
166         if (lx < ly)
167                 hz -= 1;
168         if (hz >= 0) {
169                 hx = hz;
170                 lx = lz;
171                 m += 1;
172         }
173         m &= 0x7fffffff;
174         *quo = sq >= 0 ? m : -m;

175         /* convert back to floating value and restore the sign */
176         if ((hx | lx) == 0) {   /* return sign(x)*0 */
177                 return (Zero[(unsigned) sx >> 31]);
178         }
179         while (hx < 0x00100000) {       /* normalize x */
180                 hx = hx + hx + (lx >> 31);
181                 lx = lx + lx;
182                 iy -= 1;
183         }
184         if (iy >= -1022) {      /* normalize output */
185                 hx = (hx - 0x00100000) | ((iy + 1023) << 20);
186                 __HI(x) = hx | sx;
187                 __LO(x) = lx;
188         } else {                        /* subnormal output */
189                 n = -1022 - iy;
```



```
192                 if (n <= 20) {
193                         lx = (lx >> n) | ((unsigned) hx << (32 - n));
194                         hx >>= n;
195                 } else if (n <= 31) {
196                         lx = (hx << (32 - n)) | (lx >> n);
197                         hx = sx;
198                 } else {
199                         lx = hx >> (n - 32);
200                         hx = sx;
201                 }
202                 __HI(x) = hx | sx;
203                 __LO(x) = lx;
204                 x *= one;       /* create necessary signal */
205         }
206         return (x);             /* exact output */
207 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    5166 Tue Nov  4 19:01:43 2014
new/usr/src/lib/libm/common/m9x/remquof.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remquof = remquof
#pragma weak remquof = __remquof

/* INDENT OFF */
/*
 * float remquof(float x, float y, int *quo) return remainderf(x,y) and an
 * integer pointer quo such that *quo = N mod (2**31),  where N is the
 * exact integeral part of x/y rounded to nearest even.
 *
 * remquof call internal fmodquof
 */

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#include <math.h>
extern float fabsf(float);

static const int
        is = (int) 0x80000000,
        im = 0x007fffff,
        ii = 0x7f800000,
        iu = 0x00800000;

static const float zero = 0.0F, half = 0.5F;
/* INDENT ON */

static float
fmodquof(float x, float y, int *quo) {
        float w;
        int hx, ix, iy, iz, k, ny, nd, m, sq;
```



```
        hx = *(int *) &x;
        ix = hx & 0x7fffffff;
        iy = *(int *) &y;
        sq = (iy ^ hx) & is;    /* sign of x/y */
        iy &= 0x7fffffff;

        /* purge off exception values */
        *quo = 0;
        if (ix >= ii || iy > ii || iy == 0) {
                w = x * y;
                w = w / w;
        } else if (ix <= iy) {
                if (ix < iy)
                        w = x;  /* return x if |x|<|y| */
                else {
                        *quo = 1 + (sq >> 30);
                        w = zero * x;   /* return sign(x)*0.0  */
                }
        } else {
                /* INDENT OFF */
                /*
                 * scale x,y to "normal" with
                 *      ny = exponent of y
                 *      nd = exponent of x minus exponent of y
                 */
                /* INDENT ON */
                ny = iy >> 23;
                k = ix >> 23;

                /* special case for subnormal y or x */
                if (ny == 0) {
                        ny = 1;
                        while (iy < iu) {
                                ny -= 1;
                                iy += iy;
                        }
                        nd = k - ny;
                        if (k == 0) {
                                nd += 1;
                                while (ix < iu) {
                                        nd -= 1;
                                        ix += ix;
                                }
                        } else
                                ix = iu | (ix & im);
                } else {
                        nd = k - ny;
                        ix = iu | (ix & im);
                        iy = iu | (iy & im);
                }
                /* INDENT OFF */
                /* fix point fmod for normalized ix and iy */
                /*
                 * while (nd--) {
                 *      iz = ix - iy;
                 *      if (iz < 0)
                 *              ix = ix + ix;
                 *      else if (iz == 0) {
                 *              *(int *) &w = is & hx;
                 *              return w;
                 *      } else
                 *              ix = iz + iz;
                 * }
                 */
                /* INDENT ON */
                /* unroll the above loop 4 times to gain performance */
```

```
126                 m = 0;
127                 k = nd >> 2;
128                 nd -= (k << 2);
129                 while (k--) {
130                         iz = ix - iy;
131                         if (iz >= 0) {
132                                 m += 1;
133                                 ix = iz + iz;
134                         } else
135                                 ix += ix;
136                         m += m;
137                         iz = ix - iy;
138                         if (iz >= 0) {
139                                 m += 1;
140                                 ix = iz + iz;
141                         } else
142                                 ix += ix;
143                         m += m;
144                         iz = ix - iy;
145                         if (iz >= 0) {
146                                 m += 1;
147                                 ix = iz + iz;
148                         } else
149                                 ix += ix;
150                         m += m;
151                         iz = ix - iy;
152                         if (iz >= 0) {
153                                 m += 1;
154                                 ix = iz + iz;
155                         } else
156                                 ix += ix;
157                         m += m;
158                         if (iz == 0) {
159                                 iz = (k << 2) + nd;
160                                 if (iz < 32)
161                                         m <<= iz;
162                                 else
163                                         m = 0;
164                                 m &= 0x7fffffff;
165                                 *quo = sq >= 0 ? m : -m;
166                                 *(int *) &w = is & hx;
167                                 return (w);
168                         }
169                 }
170                 while (nd--) {
171                         iz = ix - iy;
172                         if (iz >= 0) {
173                                 m += 1;
174                                 ix = iz + iz;
175                         } else
176                                 ix += ix;
177                         m += m;
178                 }
179                 /* end of unrolling */

181                 iz = ix - iy;
182                 if (iz >= 0) {
183                         m += 1;
184                         ix = iz;
185                 }
186                 m &= 0x7fffffff;
187                 *quo = sq >= 0 ? m : -m;

189                 /* convert back to floating value and restore the sign */
190                 if (ix == 0) {
191                         *(int *) &w = is & hx;
```



```
192                         return (w);
193                 }
194                 while (ix < iu) {
195                         ix += ix;
196                         ny -= 1;
197                 }
198                 while (ix > (iu + iu)) {
199                         ny += 1;
200                         ix >>= 1;
201                 }
202                 if (ny > 0)
203                         *(int *) &w = (is & hx) | (ix & im) | (ny << 23);
204                 else {          /* subnormal output */
205                         k = -ny + 1;
206                         ix >>= k;
207                         *(int *) &w = (is & hx) | ix;
208                 }
209         }
210         return (w);
211 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6845 Tue Nov  4 19:01:43 2014
new/usr/src/lib/libm/common/m9x/remquol.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __remquol = remquol
#pragma weak remquol = __remquol

#include "libm.h"
#include "libm_synonyms.h"
#if defined(__SUNPRO_C)
#include <sunmath.h>                    /* fabsl */
#endif
/* INDENT OFF */
static const int
        is = -0x7fffffff - 1,
        im = 0x0000ffff,
        iu = 0x00010000;

static const long double zero = 0.0L, one = 1.0L;
/* INDENT ON */

#if defined(__sparc)
#define __H0(x) ((int *) &x)[0]
#define __H1(x) ((int *) &x)[1]
#define __H2(x) ((int *) &x)[2]
#define __H3(x) ((int *) &x)[3]
#else
#error Unsupported architecture
#endif

/*
 * On entrance: *quo is initialized to 0, x finite and y non-zero & ordered
 */
static long double
fmodquol(long double x, long double y, int *quo) {
        long double a, b;
```



```
        int n, ix, iy, k, sx, sq, m;
        int hx;
        int x0, y0, z0, carry;
        unsigned x1, x2, x3, y1, y2, y3, z1, z2, z3;

        hx = __H0(x);
        x1 = __H1(x);
        x2 = __H2(x);
        x3 = __H3(x);
        y0 = __H0(y);
        y1 = __H1(y);
        y2 = __H2(y);
        y3 = __H3(y);

        sx = hx & is;
        sq = (hx ^ y0) & is;
        x0 = hx ^ sx;
        y0 &= ~0x80000000;

        a = fabsl(x);
        b = fabsl(y);
        if (a <= b) {
                if (a < b)
                        return (x);
                else {
                        *quo = 1 + (sq >> 30);
                        return (zero * x);
                }
        }
        /* determine ix = ilogbl(x) */
        if (x0 < iu) {          /* subnormal x */
                ix = 0;
                ix = -16382;
                while (x0 == 0) {
                        ix -= 16;
                        x0 = x1 >> 16;
                        x1 = (x1 << 16) | (x2 >> 16);
                        x2 = (x2 << 16) | (x3 >> 16);
                        x3 = (x3 << 16);
                }
                while (x0 < iu) {
                        ix -= 1;
                        x0 = (x0 << 1) | (x1 >> 31);
                        x1 = (x1 << 1) | (x2 >> 31);
                        x2 = (x2 << 1) | (x3 >> 31);
                        x3 <<= 1;
                }
        } else {
                ix = (x0 >> 16) - 16383;
                x0 = iu | (x0 & im);
        }

        /* determine iy = ilogbl(y) */
        if (y0 < iu) {          /* subnormal y */
                iy = -16382;
                while (y0 == 0) {
                        iy -= 16;
                        y0 = y1 >> 16;
                        y1 = (y1 << 16) | (y2 >> 16);
                        y2 = (y2 << 16) | (y3 >> 16);
                        y3 = (y3 << 16);
                }
                while (y0 < iu) {
                        iy -= 1;
                        y0 = (y0 << 1) | (y1 >> 31);
                        y1 = (y1 << 1) | (y2 >> 31);
```

```
126                         y2 = (y2 << 1) | (y3 >> 31);
127                         y3 <<= 1;
128                 }
129         } else {
130                 iy = (y0 >> 16) - 16383;
131                 y0 = iu | (y0 & im);
132         }


135         /* fix point fmod */
136         n = ix - iy;
137         m = 0;
138         while (n--) {
139                 while (x0 == 0 && n >= 16) {
140                         m <<= 16;
141                         n -= 16;
142                         x0 = x1 >> 16;
143                         x1 = (x1 << 16) | (x2 >> 16);
144                         x2 = (x2 << 16) | (x3 >> 16);
145                         x3 = (x3 << 16);
146                 }
147                 while (x0 < iu && n >= 1) {
148                         m += m;
149                         n -= 1;
150                         x0 = (x0 << 1) | (x1 >> 31);
151                         x1 = (x1 << 1) | (x2 >> 31);
152                         x2 = (x2 << 1) | (x3 >> 31);
153                         x3 = (x3 << 1);
154                 }
155                 carry = 0;
156                 z3 = x3 - y3;
157                 carry = z3 > x3;
158                 if (carry == 0) {
159                         z2 = x2 - y2;
160                         carry = z2 > x2;
161                 } else {
162                         z2 = x2 - y2 - 1;
163                         carry = z2 >= x2;
164                 }
165                 if (carry == 0) {
166                         z1 = x1 - y1;
167                         carry = z1 > x1;
168                 } else {
169                         z1 = x1 - y1 - 1;
170                         carry = z1 >= x1;
171                 }
172                 z0 = x0 - y0 - carry;
173                 if (z0 < 0) {   /* double x */
174                         x0 = x0 + x0 + ((x1 & is) != 0);
175                         x1 = x1 + x1 + ((x2 & is) != 0);
176                         x2 = x2 + x2 + ((x3 & is) != 0);
177                         x3 = x3 + x3;
178                         m += m;
179                 } else {
180                         m += 1;
181                         if (z0 == 0) {
182                                 if ((z1 | z2 | z3) == 0) {
183                                         /* 0: we are done */
184                                         if (n < 31)
185                                                 m <<= (1 + n);
186                                         else
187                                                 m = 0;
188                                         m &= ~0x80000000;
189                                         *quo = sq >= 0 ? m : -m;
190                                         __H0(a) = hx & is;
191                                         __H1(a) = __H2(a) = __H3(a) = 0;
```



```
192                                         return (a);
193                                 }
194                         }
195                         /* x = z << 1 */
196                         z0 = z0 + z0 + ((z1 & is) != 0);
197                         z1 = z1 + z1 + ((z2 & is) != 0);
198                         z2 = z2 + z2 + ((z3 & is) != 0);
199                         z3 = z3 + z3;
200                         x0 = z0;
201                         x1 = z1;
202                         x2 = z2;
203                         x3 = z3;
204                         m += m;
205                 }
206         }
207         carry = 0;
208         z3 = x3 - y3;
209         carry = z3 > x3;
210         if (carry == 0) {
211                 z2 = x2 - y2;
212                 carry = z2 > x2;
213         } else {
214                 z2 = x2 - y2 - 1;
215                 carry = z2 >= x2;
216         }
217         if (carry == 0) {
218                 z1 = x1 - y1;
219                 carry = z1 > x1;
220         } else {
221                 z1 = x1 - y1 - 1;
222                 carry = z1 >= x1;
223         }
224         z0 = x0 - y0 - carry;
225         if (z0 >= 0) {
226                 x0 = z0;
227                 x1 = z1;
228                 x2 = z2;
229                 x3 = z3;
230                 m += 1;
231         }
232         m &= ~0x80000000;
233         *quo = sq >= 0 ? m : -m;

235         /* convert back to floating value and restore the sign */
236         if ((x0 | x1 | x2 | x3) == 0) {
237                 __H0(a) = hx & is;
238                 __H1(a) = __H2(a) = __H3(a) = 0;
239                 return (a);
240         }
241         while (x0 < iu) {
242                 if (x0 == 0) {
243                         iy -= 16;
244                         x0 = x1 >> 16;
245                         x1 = (x1 << 16) | (x2 >> 16);
246                         x2 = (x2 << 16) | (x3 >> 16);
247                         x3 = (x3 << 16);
248                 } else {
249                         x0 = x0 + x0 + ((x1 & is) != 0);
250                         x1 = x1 + x1 + ((x2 & is) != 0);
251                         x2 = x2 + x2 + ((x3 & is) != 0);
252                         x3 = x3 + x3;
253                         iy -= 1;
254                 }
255         }

257         /* normalize output */
```

```
258         if (iy >= -16382) {
259                 __H0(a) = sx | (x0 - iu) | ((iy + 16383) << 16);
260                 __H1(a) = x1;
261                 __H2(a) = x2;
262                 __H3(a) = x3;
263         } else {                /* subnormal output */
264                 n = -16382 - iy;
265                 k = n & 31;
266                 if (k <= 16) {
267                         x3 = (x2 << (32 - k)) | (x3 >> k);
268                         x2 = (x1 << (32 - k)) | (x2 >> k);
269                         x1 = (x0 << (32 - k)) | (x1 >> k);
270                         x0 >>= k;
271                 } else {
272                         x3 = (x2 << (32 - k)) | (x3 >> k);
273                         x2 = (x1 << (32 - k)) | (x2 >> k);
274                         x1 = (x0 << (32 - k)) | (x1 >> k);
275                         x0 = 0;
276                 }
277                 while (n >= 32) {
278                         n -= 32;
279                         x3 = x2;
280                         x2 = x1;
281                         x1 = x0;
282                         x0 = 0;
283                 }
284                 __H0(a) = x0 | sx;
285                 __H1(a) = x1;
286                 __H2(a) = x2;
287                 __H3(a) = x3;
288                 a *= one;
289         }
290         return (a);
291 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1921 Tue Nov  4 19:01:44 2014
new/usr/src/lib/libm/common/m9x/round.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __round = round
#pragma weak round = __round

#include "libm.h"

double
round(double x) {
        union {
                unsigned i[2];
                double d;
        } xx;
        unsigned hx, sx, i;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;
        sx = xx.i[HIWORD] & 0x80000000;
        if (hx < 0x43300000) {  /* |x| < 2^52 */
                if (hx < 0x3ff00000) {  /* |x| < 1 */
                        if (hx >= 0x3fe00000)
                                return (sx ? -1.0 : 1.0);
                        return (sx ? -0.0 : 0.0);
                }

                /* round x at the integer bit */
                if (hx < 0x41300000) {
                        i = 1 << (0x412 - (hx >> 20));
                        xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
                        xx.i[LOWORD] = 0;
                } else {
                        i = 1 << (0x432 - (hx >> 20));
                        xx.i[LOWORD] += i;
                        if (xx.i[LOWORD] < i)
```



```
                                xx.i[HIWORD]++;
                        xx.i[LOWORD] &= ~(i | (i - 1));
                }
                return (xx.d);
        } else if (hx < 0x7ff00000)
                return (x);
        else
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (hx >= 0x7ff80000 ? x : x + x);
                /* assumes sparc-like QNaN */
#else
                return (x + x);
#endif
}
```

*unchanged_portion_omitted_*

```
**********************************************************
    1686 Tue Nov  4 19:01:44 2014
new/usr/src/lib/libm/common/m9x/roundf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __roundf = roundf
#pragma weak roundf = __roundf

#include "libm.h"

float
roundf(float x) {
        union {
                unsigned i;
                float f;
        } xx;
        unsigned hx, sx, i;

        xx.f = x;
        hx = xx.i & ~0x80000000;
        sx = xx.i & 0x80000000;
        if (hx < 0x4b000000) {          /* |x| < 2^23 */
                if (hx < 0x3f800000) {          /* |x| < 1 */
                        if (hx >= 0x3f000000)
                                return (sx ? -1.0F : 1.0F);
                        return (sx ? -0.0F : 0.0F);
                }

                /* round x at the integer bit */
                i = 1 << (0x95 - (hx >> 23));
                xx.i = (xx.i + i) & ~((i << 1) - 1);
                return (xx.f);
        } else if (hx < 0x7f800000)     /* |x| is integral */
                return (x);
        else
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (hx > 0x7f800000 ? x * x : x + x);
```



```
#else
                return (x + x);
#endif
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3658 Tue Nov  4 19:01:45 2014
new/usr/src/lib/libm/common/m9x/roundl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __roundl = roundl
#pragma weak roundl = __roundl

#include "libm.h"

#if defined(__sparc)
long double
roundl(long double x) {
        union {
                unsigned i[4];
                long double q;
        } xx;
        unsigned hx, sx, v;
        int j;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;

        /* handle trivial cases */
        if (hx >= 0x406f0000) /* |x| >= 2^112 + ... or x is nan */
                return (hx >= 0x7fff0000 ? x + x : x);

        /* handle |x| < 1 */
        if (hx < 0x3fff0000) {
                if (hx >= 0x3ffe0000)
                        return (sx ? -1.0L : 1.0L);
                return (sx ? -0.0L : 0.0L);
        }

        xx.i[0] = hx;
        j = 0x406f - (hx >> 16);                /* 1 <= j <= 112 */
```



```
        if (j >= 96) {                          /* 96 <= j <= 112 */
                v = (1U << (j - 96)) >> 1;
                if (v) {
                        if (xx.i[0] & v)
                                xx.i[0] += v;
                        xx.i[0] &= ~(v - 1);
                } else if (xx.i[1] & 0x80000000)
                                ++xx.i[0];
                xx.i[1] = xx.i[2] = xx.i[3] = 0;
        } else if (j >= 64) {                   /* 64 <= j <= 95 */
                v = (1U << (j - 64)) >> 1;
                if (v) {
                        if (xx.i[1] & v) {
                                xx.i[1] += v;
                                if (xx.i[1] < v)
                                        ++xx.i[0];
                        }
                        xx.i[1] &= ~(v - 1);
                } else if (xx.i[2] & 0x80000000) {
                                if (++xx.i[1] == 0)
                                        ++xx.i[0];
                }
                xx.i[2] = xx.i[3] = 0;
        } else if (j >= 32) {                   /* 32 <= j <= 63 */
                v = (1U << (j - 32)) >> 1;
                if (v) {
                        if (xx.i[2] & v) {
                                xx.i[2] += v;
                                if (xx.i[2] < v) {
                                        if (++xx.i[1] == 0)
                                                ++xx.i[0];
                                }
                        }
                        xx.i[2] &= ~(v - 1);
                } else if (xx.i[3] & 0x80000000) {
                                if (++xx.i[2] == 0) {
                                        if (++xx.i[1] == 0)
                                                ++xx.i[0];
                                }
                }
                xx.i[3] = 0;
        } else {                                /* 1 <= j <= 31 */
                v = 1U << (j - 1);
                if (xx.i[3] & v) {
                        xx.i[3] += v;
                        if (xx.i[3] < v) {
                                if (++xx.i[2] == 0) {
                                        if (++xx.i[1] == 0)
                                                ++xx.i[0];
                                }
                        }
                }
                xx.i[3] &= ~(v - 1);
        }

        /* negate result if need be */
        if (sx)
                xx.i[0] |= 0x80000000;
        return (xx.q);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2659 Tue Nov  4 19:01:45 2014
new/usr/src/lib/libm/common/m9x/scalbln.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalbln = scalbln
#pragma weak scalbln = __scalbln

#include "libm.h"
#include <float.h>              /* DBL_MAX, DBL_MIN */

static const double twom54 = 5.5511151231257827021181583404541015625e-17;
#if defined(__x86)
static const double two52 = 4503599627370496.0;
#else
/*
 * Normalize non-zero subnormal x and return biased exponent of x in [-51,0]
 */
static int
ilogb_biased(unsigned *px) {
        int s = 52;
        unsigned v = px[HIWORD] & ~0x80000000, w = px[LOWORD], t = v;

        if (t)
                s -= 32;
        else
                t = w;
        if (t & 0xffff0000)
                s -= 16, t >>= 16;
        if (t & 0xff00)
                s -= 8, t >>= 8;
        if (t & 0xf0)
                s -= 4, t >>= 4;
        t <<= 1;
        s -= (0xffffaa50 >> t) & 0x3;
        if (s < 32) {
                v = (v << s) | w >> (32 - s);
```



```
                w <<= s;
        } else {
                v = w << (s - 32);
                w = 0;
        }
        px[HIWORD] = (px[HIWORD] & 0x80000000) | v;
        px[LOWORD] = w;
        return (1 - s);
}
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2273 Tue Nov  4 19:01:45 2014
new/usr/src/lib/libm/common/m9x/scalblnf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalblnf = scalblnf
#pragma weak scalblnf = __scalblnf

#include "libm.h"
#include <float.h>              /* FLT_MAX, FLT_MIN */

static const float twom25f = 2.98023223876953125e-8F;
#if defined(__x86)
static const float two23f = 8388608.0F;
#else
/*
 * v: a non-zero subnormal |x|; returns [-22, 0]
 */
static int
ilogbf_biased(unsigned v) {
        int r = -22;

        if (v & 0xffff0000)
                r += 16, v >>= 16;
        if (v & 0xff00)
                r += 8, v >>= 8;
        if (v & 0xf0)
                r += 4, v >>= 4;
        v <<= 1;
        return (r + ((0xffffaa50 >> v) & 0x3));
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    2404 Tue Nov  4 19:01:46 2014
new/usr/src/lib/libm/common/m9x/scalblnl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __scalblnl = scalblnl
#pragma weak scalblnl = __scalblnl

#include "libm.h"
#include <float.h>              /* LDBL_MAX, LDBL_MIN */

#if defined(__sparc)
#define XSET_EXP(k, x)  ((int *) &x)[0] = (((int *) &x)[0] & ~0x7fff0000) | \
                        (k << 16)
#define ISINFNANL(k, x) (k == 0x7fff)
#define XTWOT_OFFSET    113
static const long double xtwot = 10384593717069655257060992658440192.0L,
                                                                /* 2^113 */
        twomtm1 = 4.81482486096808963263994485646231829 6E-35L;  /* 2^-114 */
#elif defined(__x86)
#define XSET_EXP(k, x)  ((int *) &x)[2] = (((int *) &x)[2] & ~0x7fff) | k
#if defined(HANDLE_UNSUPPORTED)
#define      ISINFNANL(k, x) (k == 0x7fff || \
                        (k != 0 && (((int *) &x)[1] & 0x80000000) == 0))
#else
#define ISINFNANL(k, x) (k == 0x7fff)
#endif
#define XTWOT_OFFSET    64
static const long double xtwot = 18446744073709551616.0L,       /* 2^64 */
        twomtm1 = 2.71050543121376108501 86E-20L;               /* 2^-65 */
#endif

long double
scalblnl(long double x, long n) {
        int k = XBIASED_EXP(x);

        if (ISINFNANL(k, x))
```



```
                return (x + x);
        if (ISZEROL(x) || n == 0)
                return (x);
        if (k == 0) {
                x *= xtwot;
                k = XBIASED_EXP(x) - XTWOT_OFFSET;
        }
        k += (int) n;
        if (n > 50000 || k > 0x7ffe)
                return (LDBL_MAX * copysignl(LDBL_MAX, x));
        if (n < -50000 || k <= -XTWOT_OFFSET - 1)
                return (LDBL_MIN * copysignl(LDBL_MIN, x));
        if (k > 0) {
                XSET_EXP(k, x);
                return (x);
        }
        k += XTWOT_OFFSET + 1;
        XSET_EXP(k, x);
        return (x * twomtm1);
}
```

_____unchanged_portion_omitted_

```
**********************************************************
   68964 Tue Nov  4 19:01:46 2014
new/usr/src/lib/libm/common/m9x/tgamma.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tgamma = tgamma
#pragma weak tgamma = __tgamma

/* INDENT OFF */
/*
 * True gamma function
 * double tgamma(double x)
 *
 * Error:
 * ------
 *      Less that one ulp for both positive and negative arguments.
 *
 * Algorithm:
 * ----------
 *      A: For negative argument
 *              (1) gamma(-n or -inf) is NaN
 *              (2) Underflow Threshold
 *              (3) Reduction to gamma(1+x)
 *      B: For x between 1 and 2
 *      C: For x between 0 and 1
 *      D: For x between 2 and 8
 *      E: Overflow thresold {see over.c}
 *      F: For overflow_threshold >= x >= 8
 *
 * Implementation details
 * ----------------------
 *                                                      -pi
 * (A) For negative argument, use gamma(-x) = -------------------------.
 *                                          (sin(pi*x)*gamma(1+x))
 *
 *   (1) gamma(-n or -inf) is NaN with invalid signal by SUSv3 spec.
 *       (Ideally, gamma(-n) = 1/sinpi(n) = (-1)**(n+1) * inf.)
```



```
 *
 *   (2) Underflow Threshold. For each precision, there is a value T
 *      such that when x>T and when x is not an integer, gamma(-x) will
 *       always underflow. A table of the underflow threshold value is given
 *      below. For proof, see file "under.c".
 *
 *      Precision       underflow threshold T =
 *      ---------------------------------------------------------------------
 *      single  41.000041962                                    = 41  + 11 ULP
 *              (machine format) 4224000B
 *      double  183.000000000000312639                          = 183 + 11 ULP
 *              (machine format) 4066E000 0000000B
 *      quad    1774.00000000000000000000000000017749370        = 1774 + 9 ULP
 *              (machine format) 4009BB8000000000000000000000009
 *      ---------------------------------------------------------------------
 *
 *   (3) Reduction to gamma(1+x).
 *      Because of (1) and (2), we need only consider non-integral x
 *      such that 0<x<T. Let k = [x] and z = x-[x]. Define
 *                   sin(x*pi)               cos(x*pi)
 *      kpsin(x) = --------- and kpcos(x) = --------- . Then
 *                      pi                      pi
 *                                  1
 *              gamma(-x) = --------------------.
 *                          -kpsin(x)*gamma(1+x)
 *      Since x = k+z,
 *                                                      k+1
 *              -sin(x*pi) = -sin(k*pi+z*pi) = (-1)   *sin(z*pi),
 *                                  k+1
 *      we have -kpsin(x) = (-1)   * kpsin(z).  We can further
 *      reduce z to t by
 *         (I)   t = z       when 0.00000     <= z < 0.31830...
 *         (II)  t = 0.5-z   when 0.31830...  <= z < 0.681690...
 *         (III) t = 1-z     when 0.681690... <= z < 1.00000
 *      and correspondingly
 *         (I)   kpsin(z) = kpsin(t)    ... 0<= z < 0.3184
 *         (II)  kpsin(z) = kpcos(t)    ... |t|   < 0.182
 *         (III) kpsin(z) = kpsin(t)    ... 0<= t < 0.3184
 *
 *      Using a special Remez algorithm, we obtain the following polynomial
 *      approximation for kpsin(t) for 0<=t<0.3184:
 *
 *      Computation note: in simulating higher precision arithmetic, kcpsin
 *      return head = t and tail = ks[0]*t^3 + (...) to maintain extra bits.
 *
 *      Quad precision, remez error <= 2**(-129.74)
 *                                    3           5                   27
 *          kpsin(t) = t + ks[0] * t  + ks[1] * t  + ... + ks[12] * t
 *
 *       ks[ 0] =  -1.64493406684822643647241516664602518705158902870e+0000
 *       ks[ 1] =   8.11742425283353643637002772405874238094995726160e-0001
 *       ks[ 2] =  -1.90751824122084213696472111835337366232282723933e-0001
 *       ks[ 3] =   2.61478478176548005046532613563241288115395517084e-0002
 *       ks[ 4] =  -2.34608103545582363750893072647117829448016479971e-0003
 *       ks[ 5] =   1.48428793031071003684606647212534027556262040158e-0004
 *       ks[ 6] =  -6.97587366165638046518462722252768122615952898698e-0006
 *       ks[ 7] =   2.53121740413702536928659271747187500934840057929e-0007
 *       ks[ 8] =  -7.30471182221385990397683641695766121301933621956e-0009
 *       ks[ 9] =   1.71653847451163495739958249695549313987973589884e-0010
 *       ks[10] =  -3.34813314714560776122245796929054813458341420565e-0012
 *       ks[11] =   5.50724992262622033449487808306969135431411753047e-0014
 *       ks[12] =  -7.67678132753577998601234393215802221104236979928e-0016
 *
 *      Double precision, Remez error <= 2**(-62.9)
 *                                  3           5                 15
 *          kpsin(t) = t + ks[0] * t  + ks[1] * t  + ... + ks[6] * t
```

```
 127  *
 128  *       ks[0] =  -1.64493406684822640606569l   (0x3ffa51a6 625307d3)
 129  *       ks[1] =   8.11742425283341655883668741874008920850698590621e-0001
 130  *       ks[2] =  -1.90751824120862873825597279118304943994042258291e-0001
 131  *       ks[3] =   2.61478477632554278317289628332654539353521911570e-0002
 132  *       ks[4] =  -2.34607978510202710377617190278735525354347705866e-0003
 133  *       ks[5] =   1.48413292290051695897242899977121846763824221705e-0004
 134  *       ks[6] =  -6.87730769637543488108688726777687262485357072242e-0006
 135  *
 136  *      Single precision, Remez error <= 2**(-34.09)
 137  *                                3           5                 9
 138  *          kpsin(t) = t + ks[0] * t  + ks[1] * t  + ... + ks[3] * t
 139  *
 140  *       ks[0] =  -1.64493404985645811354476665052005342839447790544e+0000
 141  *       ks[1] =   8.11740794458351064092797249069438269367389272270e-0001
 142  *       ks[2] =  -1.90703144603551216933075809162889536878854055202e-0001
 143  *       ks[3] =   2.55742333994264563281155312271481108635575331201e-0002
 144  *
 145  *      Computation note: in simulating higher precision arithmetic, kcpsin
 146  *      return head = t and tail = kc[0]*t^3 + (...) to maintain extra bits
 147  *      precision.
 148  *
 149  *      And for kpcos(t) for |t|< 0.183:
 150  *
 151  *      Quad precision, remez <= 2**(-122.48)
 152  *                                  2           4                   22
 153  *          kpcos(t) = 1/pi +  pi/2 * t  + kc[2] * t + ... + kc[11] * t
 154  *
 155  *       kc[2] =   1.29192819501249250731151312779548918765320728489e+0000
 156  *       kc[3] =  -4.25027339979557573976029596929319207009444090366e-0001
 157  *       kc[4] =   7.49080661650990096109672954618317623888421628613e-0002
 158  *       kc[5] =  -8.21458866111282287985539464173976555436050215120e-0003
 159  *       kc[6] =   6.14202578809529228503205255165761204750211603402e-0004
 160  *       kc[7] =  -3.33073432691149607007217330302595267179545908740e-0005
 161  *       kc[8] =   1.36970959047832085796809745461530865597993680204e-0006
 162  *       kc[9] =  -4.41780774262583514450246512727201806217271097336e-0008
 163  *       kc[10]=   1.14741409212381858820016567664488123478660705759e-0009
 164  *       kc[11]=  -2.44261236114707374558437500654381006300502749632e-0011
 165  *
 166  *      Double precision, remez < 2**(61.91)
 167  *                                 2           4                  12
 168  *          kpcos(t) = 1/pi + pi/2 *t +  kc[2] * t  + ... + kc[6] * t
 169  *
 170  *       kc[2] =   1.29192819501230224953283586722575766189551966008e+0000
 171  *       kc[3] =  -4.25027339940149518500158850753393173519732149213e-0001
 172  *       kc[4] =   7.49080625187015312373925142219429422375556727752e-0002
 173  *       kc[5] =  -8.21442040906099210866977352284054849051348692715e-0003
 174  *       kc[6] =   6.10411356829515414575566564733632532333904115968e-0004
 175  *
 176  *      Single precision, remez < 2**(-30.13)
 177  *                                2                6
 178  *          kpcos(t) = kc[0] +  kc[1] * t  + ... + kc[3] * t
 179  *
 180  *       kc[0] =   3.18309886183790671537767526745028724068919291480e-0001
 181  *       kc[1] =  -1.57079581447762568199467875065854538626594937791e+0000
 182  *       kc[2] =   1.29183528092558692844073004029568674027807393862e+0000
 183  *       kc[3] =  -4.20232949771307685981015914425195471602739075537e-0001
 184  *
 185  *      Computation note: in simulating higher precision arithmetic, kcpcos
 186  *      return head = 1/pi chopped, and tail = pi/2 *t^2 + (tail part of 1/pi
 187  *      + ...) to maintain extra bits precision. In particular, pi/2 * t^2
 188  *      is calculated with great care.
 189  *
 190  *      Thus, the computation of gamma(-x), x>0, is:
 191  *      Let k = int(x), z = x-k.
 192  *      For z in (I)
```



```
 193  *                                    k+1
 194  *                                (-1)
 195  *              gamma(-x) = ---------------------- ;
 196  *                          kpsin(z)*gamma(1+x)
 197  *
 198  *      otherwise, for z in (II),
 199  *                                    k+1
 200  *                                (-1)
 201  *              gamma(-x) = ---------------------- ;
 202  *                          kpcos(0.5-z)*gamma(1+x)
 203  *
 204  *      otherwise, for z in (III),
 205  *                                    k+1
 206  *                                (-1)
 207  *              gamma(-x) = ---------------------- .
 208  *                          kpsin(1-z)*gamma(1+x)
 209  *
 210  *      Thus, the computation of gamma(-x) reduced to the computation of
 211  *      gamma(1+x) and kpsin(), kpcos().
 212  *
 213  * (B) For x between 1 and 2.  We break [1,2] into three parts:
 214  *      GT1 = [1.0000, 1.2845]
 215  *      GT2 = [1.2844, 1.6374]
 216  *      GT3 = [1.6373, 2.0000]
 217  *
 218  *    For x in GTi, i=1,2,3, let
 219  *      z1  =  1.134861805732790769689793935774652917006
 220  *      gz1 = gamma(z1)  =   0.938204627909682449409753561580326957698 8
 221  *      tz1 = gamma'(z1) =  -0.351721435785293579101562500000000000000 0
 222  *
 223  *      z2  =  1.461632144968362341262659542325721328468e+0000
 224  *      gz2 = gamma(z2)  = 0.885603194410888700278815900582588733208 0
 225  *      tz2 = gamma'(z2) = 0.00
 226  *
 227  *      z3  =  1.819773101100500601787868704921606996312e+0000
 228  *      gz3 = gamma(z3)  = 0.936781411463652321618846897080837849742 6
 229  *      tz3 = gamma'(z3) = 0.280530631542205810546875000000000000000 0
 230  *
 231  *    and
 232  *      y = x-zi        ... for extra precision, write y = y.h + y.l
 233  *    Then
 234  *      gamma(x) = gzi + tzi*(y.h+y.l) + y*y*Ri(y),
 235  *               = gzi.h + (tzi*y.h + ((tzi*y.l+gzi.l) +  y*y*Ri(y)))
 236  *               = gy.h + gy.l
 237  *    where
 238  *      (I) For double precision
 239  *
 240  *              Ri(y) = Pi(y)/Qi(y), i=1,2,3;
 241  *
 242  *              P1(y) = p1[0] + p1[1]*y + ... + p1[4]*y^4
 243  *              Q1(y) = q1[0] + q1[1]*y + ... + q1[5]*y^5
 244  *
 245  *              P2(y) = p2[0] + p2[1]*y + ... + p2[3]*y^3
 246  *              Q2(y) = q2[0] + q2[1]*y + ... + q2[6]*y^6
 247  *
 248  *              P3(y) = p3[0] + p3[1]*y + ... + p3[4]*y^4
 249  *              Q3(y) = q3[0] + q3[1]*y + ... + q3[5]*y^5
 250  *
 251  *              Remez precision of Ri(y):
 252  *              |gamma(x)-(gzi+tzi*y) - y*y*Ri(y)|  <= 2**-62.3 ... for i = 1
 253  *                                                  <= 2**-59.4 ... for i = 2
 254  *                                                  <= 2**-62.1 ... for i = 3
 255  *
 256  *      (II) For quad precision
 257  *
 258  *              Ri(y) = Pi(y)/Qi(y), i=1,2,3;
```

```
 259  *
 260  *              P1(y) = p1[0] + p1[1]*y + ... + p1[9]*y^9
 261  *              Q1(y) = q1[0] + q1[1]*y + ... + q1[8]*y^8
 262  *
 263  *              P2(y) = p2[0] + p2[1]*y + ... + p2[9]*y^9
 264  *              Q2(y) = q2[0] + q2[1]*y + ... + q2[9]*y^9
 265  *
 266  *              P3(y) = p3[0] + p3[1]*y + ... + p3[9]*y^9
 267  *              Q3(y) = q3[0] + q3[1]*y + ... + q3[9]*y^9
 268  *
 269  *              Remez precision of Ri(y):
 270  *              |gamma(x)-(gzi+tzi*y) - y*y*Ri(y)|  <= 2**-118.2 ... for i = 1
 271  *                                                  <= 2**-126.8 ... for i = 2
 272  *                                                  <= 2**-119.5 ... for i = 3
 273  *
 274  *      (III) For single precision
 275  *
 276  *              Ri(y) = Pi(y), i=1,2,3;
 277  *
 278  *              P1(y) = p1[0] + p1[1]*y + ... + p1[5]*y^5
 279  *
 280  *              P2(y) = p2[0] + p2[1]*y + ... + p2[5]*y^5
 281  *
 282  *              P3(y) = p3[0] + p3[1]*y + ... + p3[4]*y^4
 283  *
 284  *              Remez precision of Ri(y):
 285  *              |gamma(x)-(gzi+tzi*y) - y*y*Ri(y)|  <= 2**-30.8 ... for i = 1
 286  *                                                  <= 2**-31.6 ... for i = 2
 287  *                                                  <= 2**-29.5 ... for i = 3
 288  *
 289  *    Notes. (1) GTi and zi are choosen to balance the interval width and
 290  *               minimize the distant between gamma(x) and the tangent line at
 291  *               zi. In particular, we have
 292  *               |gamma(x)-(gzi+tzi*(x-zi))|  <=   0.01436... for x in [1,z2]
 293  *                                            <=   0.01265... for x in [z2,2]
 294  *
 295  *            (2) zi are slightly adjusted so that tzi=gamma'(zi) is very
 296  *               close to a single precision value.
 297  *
 298  *    Coefficents: Single precision
 299  *      i= 1:
 300  *       P1[0] =   7.0908725343508836027145161339801928007756127944 3e-0001
 301  *       P1[1] =  -5.1722956078865210854514197823870179010524176108 9e-0001
 302  *       P1[2] =   5.2340339452815078940582522323770647162337764327e-0001
 303  *       P1[3] =  -4.5458630871707501078404156606948041173263481489 9e-0001
 304  *       P1[4] =   4.2059649091523908545996459055925691349819095523 3e-0001
 305  *      P1[5] =  -3.5730758971237752097833218583824145864214218578 9e-0001
 306  *
 307  *      i = 2:
 308  *       p2[0] =   4.2848698398029519816605611922398428443426434457 8e-0001
 309  *       p2[1] =  -1.3070453948770913852868012162789973538665010391 4e-0001
 310  *       p2[2] =   1.6085628503805195507286121935265585154295543087 1e-0001
 311  *       p2[3] =  -9.2228516134601058377445880206737118215893794350 7e-0002
 312  *       p2[4] =   7.1924051176722526074089029260507059556062617935 7e-0002
 313  *       p2[5] =  -4.8815826559335509370311223853448463619326045957 4e-0002
 314  *
 315  *      i = 3
 316  *       p3[0] =   3.8240953111880775908112147978609213481480888728 0e-0001
 317  *       p3[1] =   2.6530988818018864795640040301349575936516785342 6e-0002
 318  *       p3[2] =   8.0681510977507917192356116941537030937629673983 5e-0002
 319  *       p3[3] =  -1.5482159166613761392884089083517435167400776479 9e-0002
 320  *       p3[4] =   1.7630823924271726853049831341689918815716518340 5e-0002
 321  *
 322  *    Coefficents: Double precision
 323  *      i = 1:
 324  *       p1[0]   =   0.709086836199777970080049271928146481513977050781250 00
```



```
 325  *       p1[1]   =   1.719870613930485580895795133843564416683517200 61e-0001
 326  *       p1[2]   =  -3.192733457919909702933203161228139605277054506 71e-0002
 327  *       p1[3]   =   8.361726454191100362671696003905499735635344769 89e-0003
 328  *       p1[4]   =   1.137453366485728383331522134742779712446297581 01e-0003
 329  *       q1[0]   =   1.0
 330  *       q1[1]   =   9.719802178260329375264607317784723897913219680 82e-0001
 331  *       q1[2]   =  -7.435767433267561765940841372560426534970876660 30e-0002
 332  *       q1[3]   =  -1.193459449322655597697194705151020122469952553 72e-0001
 333  *       q1[4]   =   1.599134457514250026209351204707813822150502847 62e-0002
 334  *       q1[5]   =   1.126011368533749845665726913064023219115475507 83e-0003
 335  *      i = 2:
 336  *       p2[0]   =   0.428486815855586011814182256784988567233085632324218 75
 337  *       p2[1]   =   6.535967626689708160237188451056674184831221036290 e-0002
 338  *       p2[2]   =  -6.972808296312129313210507709251282642727689367 31e-0003
 339  *       p2[3]   =   6.463423590219817189472086056748132601661166328 99e-0003
 340  *       q2[0]   =   1.0
 341  *       q2[1]   =   4.575726205605060470625539574540620123275195139 36e-0001
 342  *       q2[2]   =  -2.521825948860754528596550034077961030834225720 36e-0001
 343  *       q2[3]   =  -1.829709454077785946813481660401031971787115528 27e-0002
 344  *       q2[4]   =   2.435747269931695664752276421288301413049538405 02e-0002
 345  *       q2[5]   =  -5.203904064669425253586459575648974112586670855 01e-0003
 346  *       q2[6]   =   4.795202513832798376355524319880232560319511338 85e-0004
 347  *      i = 3:
 348  *       p3[0]   =   0.382409479734567459008331979930517263710498809814453125
 349  *       p3[1]   =   1.428760486976681615990698140434493015729280341 40e-0001
 350  *       p3[2]   =   3.421575710522505368179238660135617607857488990 71e-0003
 351  *       p3[3]   =  -5.015426217100675214050878878569917009877097229 37e-0004
 352  *       p3[4]   =   8.892858148667409101238346881638382876183321226 70e-0004
 353  *       q3[0]   =   1.0
 354  *       q3[1]   =   3.042530866294442010022156409489578979062996331 68e-0001
 355  *       q3[2]   =  -2.231624073799994772825556728348812138731855200 06e-0001
 356  *       q3[3]   =  -1.050608677419520659218098119336701314275529036 36e-0002
 357  *       q3[4]   =   1.705117639161869824733018619808563520059266693 20e-0002
 358  *       q3[5]   =  -2.129502016836091879278994167000946307641824774 64e-0003
 359  *
 360  *    Note that all pi0 are exact in double, which is obtained by a
 361  *    special Remez Algorithm.
 362  *
 363  *    Coefficents: Quad precision
 364  *      i = 1:
 365  *       p1[0] =   0.709086836199777919037185741507610124611513720557
 366  *       p1[1] =   4.457547812064890358279159693673548356673910609 51e-0001
 367  *       p1[2] =   3.210492987358323823116622738826322100629181538 52e-0002
 368  *       p1[3] =  -5.712967963421066176517652458582891973696888643 50e-0003
 369  *       p1[4] =   6.046668928919989770816191749698558316069653527 73e-0003
 370  *       p1[5] =   8.991061869688871193962781217476525882265864516 8e-0004
 371  *       p1[6] =  -6.964968461444077414312070085270184418101755689 49e-0005
 372  *       p1[7] =   1.525970461189840208142254093001314450702138824 29e-0005
 373  *       p1[8] =   5.685210761684956738447114654074321891906815415 47e-0007
 374  *       p1[9] =   3.307496735196348952205820625202865656104189529 79e-0008
 375  *       q1[0] =   1.0+0000
 376  *       q1[1] =   1.358065117216710704085708535372570795794906506 68e+0000
 377  *       q1[2] =   2.975678101534295534053271400960630869948072952961e-0001
 378  *       q1[3] =  -1.529568359825885715029543728216818516811180978 70e-0001
 379  *       q1[4] =  -2.882485195614201097687816152890820535979545212 18e-0002
 380  *       q1[5] =   1.034753117199374052197899484563139363023783959 55e-0002
 381  *       q1[6] =   4.123102032438912223689653601243912973748227423 13e-0004
 382  *       q1[7] =  -3.126537081522908672489319251203807295183325073 88e-0004
 383  *       q1[8] =   2.366721708504097452373581056677577605270143324 58e-0005
 384  *
 385  *      i = 2:
 386  *       p2[0] =   0.428486815855585429730209907810650616737756697477
 387  *       p2[1] =   2.636221240678852229191926511515815419433626173 52e-0001
 388  *       p2[2] =   3.855206836700288657318772767413904217449714468 55e-0002
 389  *       p2[3] =   3.050659782781285499588971331902953252582352586 2e-0003
 390  *       p2[4] =   2.482329349517231288920804150540843391524504450 81e-0003
```

```
 391  *       p2[5] =   3.67092777065632360693313762221411547741550105407e-0004
 392  *       p2[6] =   3.81228045616085789674530902563145250532194518946e-0006
 393  *       p2[7] =   4.61677225867087554059531455133839175822537617677e-0006
 394  *       p2[8] =   2.18209052385703200438239200991201916609364872993e-0007
 395  *       p2[9] =   1.00490538985245846460006244065624754421022542454e-0008
 396  *       q2[0] =   1.0
 397  *       q2[1] =   9.20276350207639290567783725273128544224570775056e-0001
 398  *       q2[2] =  -4.79533683654165107448020515733883781138947771495e-0003
 399  *       q2[3] =  -1.24538337585899300494444600248687901947684291683e-0001
 400  *       q2[4] =   4.49866050763472358547524708431719114204535491412e-0003
 401  *       q2[5] =   7.20715455697920560621638325356292640604078591907e-0003
 402  *       q2[6] =  -8.68513169029126780280798337091982780598228096116e-0004
 403  *       q2[7] =  -1.25104431629401181525027098222745544809974229874e-0004
 404  *       q2[8] =   3.10558344839000038489191304550998047521253437464e-0005
 405  *       q2[9] =  -1.76829227852852176018537139573609433652506765712e-0006
 406  *
 407  *      i = 3
 408  *       p3[0] =   0.38240947973456750485027476610753556400704393889O2
 409  *       p3[1] =   3.42198093076618495415854906335908427159833377774e-0001
 410  *       p3[2] =   9.63828189500585568303961406863153237440702754858e-0002
 411  *       p3[3] =   8.76069421042696384852462044188520252156846768667e-0003
 412  *       p3[4] =   1.86477890389161491224872014149309015261897537488e-0003
 413  *       p3[5] =   8.16871354540309895879974742853701311541286944191e-0004
 414  *       p3[6] =   6.83783483674600322518695090864659381650125625216e-0005
 415  *       p3[7] =  -1.10168269719261574708565935172719209272190828456e-0006
 416  *       p3[8] =   9.66243228508380420159234853278906717065629721016e-0007
 417  *       p3[9] =   2.31858885579177250541163820671121664974334728142e-0008
 418  *       q3[0] =   1.0
 419  *       q3[1] =   8.25479821168813634632437430090376252512793067339e-0001
 420  *       q3[2] =  -1.62251363073937769739639623669295110346015576320e-0002
 421  *       q3[3] =  -1.10621286905916732758745130629426559691187579852e-0001
 422  *       q3[4] =   3.48309693970985612644464157892300155153652914590e-0003
 423  *       q3[5] =   6.73553737487488333032431261131289672347043401328e-0003
 424  *       q3[6] =  -7.63222008393372630162743587811004613050245128051e-0004
 425  *       q3[7] =  -1.35792670669190631476784768961953711773073251336e-0004
 426  *       q3[8] =   3.19610150954223587006220730065608156460205690618e-0005
 427  *       q3[9] =  -1.82096553862822346610109522015129585693354348322e-0006
 428  *
 429  * (C) For x between 0 and 1.
 430  *     Let P stand for the number of significant bits in the working precision.
 431  *                      -P                                  1
 432  *    (1)For 0 <= x <= 2   , gamma(x) is computed by --- rounded to nearest.
 433  *                                                         x
 434  *       The error is bound by 0.739 ulp(gamma(x)) in IEEE double precision.
 435  *      Proof.
 436  *                1                          2
 437  *      Since  --------  ~  x + 0.577...*x  - ...,  we have, for small x,
 438  *             gamma(x)
 439  *          1                    1
 440  *      ----------- < gamma(x) < --- and
 441  *      x(1+0.578x)              x
 442  *             1                 1          1
 443  *        0 <  --- - gamma(x) <= ---  -  ----------- < 0.578
 444  *             x                 x      x(1+0.578x)
 445  *                                  1       1                          -P
 446  *      The error is thus bounded by --- ulp(---) + 0.578. Since x <= 2   ,
 447  *                                  2       x
 448  *       1      P       1           P                                      1
 449  *      --- >= 2 , ulp(---) >= ulp(2  ) >= 2. Thus 0.578=0.289*2<=0.289ulp(-)
 450  *       x             x                                                 x
 451  *       Thus
 452  *                           1                                    1
 453  *              | gamma(x) - [---] rounded | <= (0.5+0.289)*ulp(---).
 454  *                           x                                    x
 455  *                        -P                                 1
 456  *      Note that for x<= 2  , it is easy to see that ulp(---)=ulp(gamma(x))
```



```
 457  *                                                          x
 458  *                              n                           1
 459  *      except only when x = 2 , (n<= -53). In such cases, --- is exact
 460  *                                                          x
 461  *      and therefore the error is bounded by
 462  *                             1
 463  *              0.298*ulp(---) = 0.298*2*ulp(gamma(x)) = 0.578ulp(gamma(x)).
 464  *                             x
 465  *      Thus we conclude that the error in gamma is less than 0.739 ulp.
 466  *
 467  *    (2)Otherwise, for x in GTi-1 (see B), let y = x-(zi-1). From (B) we obtain
 468  *                                                          gamma(1+x)
 469  *      gamma(1+x) = gy.h + gy.l,  then compute gamma(x) by -----------.
 470  *                                                               x
 471  *                                                            gy.h
 472  *      Implementaion note. Write x = x.h+x.l, and Let th = ----- chopped to
 473  *                                                             x
 474  *      20 bits, then
 475  *                                gy.h+gy.l
 476  *              gamma(x) = th + (----------  - th )
 477  *                                   x
 478  *                              1
 479  *                     = th + ---*(gy.h-th*x.h+gy.l-th*x.l)
 480  *                              x
 481  *
 482  * (D) For x between 2 and 8. Let n = 1+x chopped to an integer. Then
 483  *
 484  *               gamma(x)=(x-1)*(x-2)*...*(x-n)*gamma(x-n)
 485  *
 486  *     Since x-n is between 1 and 2, we can apply (B) to compute gamma(x).
 487  *
 488  *     Implementation detail. The computation of (x-1)(x-2)...(x-n) in simulated
 489  *     higher precision arithmetic can be somewhat optimized.  For example, in
 490  *     computing (x-1)*(x-2)*(x-3)*(x-4), if we compute (x-1)*(x-4) = z.h+z.l,
 491  *     then (x-2)(x-3) = z.h+2+z.l readily. In below, we list the expression
 492  *     of the formula to compute gamma(x).
 493  *
 494  *     Assume x-n is in GTi (i=1,2, or 3, see B for detail). Let y = x - n - zi.
 495  *     By (B) we have gamma(x-n) = gy.h+gy.l. If x = x.h+x.l, then we have
 496  *      n=1 (x in [2,3]):
 497  *       gamma(x) = (x-1)*gamma(x-1) = (x-1)*(gy.h+gy.l)
 498  *                = [(x.h-1)+x.l]*(gy.h+gy.l)
 499  *      n=2 (x in [3,4]):
 500  *        gamma(x) = (x-1)(x-2)*gamma(x-2) = (x-1)*(x-2)*(gy.h+gy.l)
 501  *                 = ((x.h-2)+x.l)*((x.h-1)+x.l)*(gy.h+gy.l)
 502  *                 = [x.h*(x.h-3)+2+x.l*(x+(x.h-3))]*(gy.h+gy.l)
 503  *      n=3 (x in [4,5])
 504  *       gamma(x) = (x-1)(x-2)(x-3)*(gy.h+gy.l)
 505  *                = (x.h*(x.h-3)+2+x.l*(x+(x.h-3)))*[((x.h-3)+x.l)(gy.h+gy.l)]
 506  *      n=4 (x in [5,6])
 507  *       gamma(x) = [(x-1)(x-4)]*[(x-2)(x-3)]*(gy.h+gy.l)
 508  *                = [(x.h*(x.h-5)+4+x.l(x+(x.h-5)))]*[(x-2)*(x-3)]*(gy.h+gy.l)
 509  *                = (y.h+y.l)*(y.h+1+y.l)*(gy.h+gy.l)
 510  *      n=5 (x in [6,7])
 511  *       gamma(x) = [(x-1)(x-4)]*[(x-2)(x-3)]*[(x-5)*(gy.h+gy.l)]
 512  *      n=6 (x in [7,8])
 513  *       gamma(x) = [(x-1)(x-6)]*[(x-2)(x-5)]*[(x-3)(x-4)]*(gy.h+gy.l)]
 514  *                = [(y.h+y.l)(y.h+4+y.l)]][(y.h+6+y.l)(gy.h+gy.l)]
 515  *
 516  * (E)Overflow Thresold. For x > Overflow thresold of gamma,
 517  *    return huge*huge (overflow).
 518  *
 519  *    By checking whether lgamma(x) >= 2**{128,1024,16384}, one can
 520  *    determine the overflow threshold for x in single, double, and
 521  *    quad precision. See over.c for details.
 522  *
```

```
 523  *    The overflow threshold of gamma(x) are
 524  *
 525  *    single: x = 3.5040096283e+01
 526  *              = 0x420C290F (IEEE single)
 527  *    double: x = 1.71624376956302711505e+02
 528  *              = 0x406573FAE561F647 (IEEE double)
 529  *    quad:   x = 1.75554834290446291700388921607020320341 77e+03
 530  *              = 0x4009B6E3180CD66A5C4206F128BA77F4  (quad)
 531  *
 532  * (F)For overflow_threshold >= x >= 8, we use asymptotic approximation.
 533  *    (1) Stirling's formula
 534  *
 535  *      log(G(x)) ~= (x-.5)*(log(x)-1) + .5(log(2*pi)-1) + (1/x)*P(1/(x*x))
 536  *                 = L1 + L2 + L3,
 537  *    where
 538  *              L1(x) = (x-.5)*(log(x)-1),
 539  *              L2    = .5(log(2pi)-1) = 0.41893853....,
 540  *              L3(x) = (1/x)P(1/(x*x)),
 541  *
 542  *    The range of L1,L2, and L3 are as follows:
 543  *
 544  *      ------------------------------------------------------------------
 545  *      Range(L1) =  (single) [8.09..,88.30..]   =[2** 3.01..,2**  6.46..]
 546  *                   (double) [8.09..,709.3..]   =[2** 3.01..,2**  9.47..]
 547  *                   (quad)   [8.09..,11356.10..]=[2** 3.01..,2** 13.47..]
 548  *      Range(L2) = 0.41893853.....
 549  *      Range(L3) = [0.0104...., 0.00048....]    =[2**-6.58..,2**-11.02..]
 550  *      ------------------------------------------------------------------
 551  *
 552  *    Gamma(x) is then computed by exp(L1+L2+L3).
 553  *
 554  *    (2) Error analysis of (F):
 555  *    --------------------------
 556  *    The error in Gamma(x) depends on the error inherited in the computation
 557  *    of L= L1+L2+L3. Let L' be the computed value of L. The absolute error
 558  *    in L' is t = L-L'. Since exp(L') = exp(L-t) = exp(L)*exp(t) ~
 559  *    (1+t)*exp(L), the relative error in exp(L') is approximately t.
 560  *
 561  *    To guarantee the relatively accuracy in exp(L'), we would like
 562  *    |t| < 2**(-P-5) where P denotes for the number of significant bits
 563  *    of the working precision. Consequently, each of the L1,L2, and L3
 564  *    must be computed with absolute error bounded by 2**(-P-5) in absolute
 565  *    value.
 566  *
 567  *    Since L2 is a constant, it can be pre-computed to the desired accuracy.
 568  *    Also |L3| < 2**-6; therefore, it suffices to compute L3 with the
 569  *    working precision.  That is,
 570  *      L3(x) approxmiate log(G(x))-(x-.5)(log(x)-1)-.5(log(2pi)-1)
 571  *    to a precision bounded by 2**(-P-5).
 572  *
 573  *                                  2**(-6)
 574  *                               _________V___________________
 575  *              L1(x):     |_____|_____|_____________________|
 576  *                            __ _____________________________
 577  *              L2:              |__|_____________________________|
 578  *                                  _____________________________
 579  *         +    L3(x):              |_____________________________|
 580  *                      ----------------------------------------
 581  *                        [leading] + [Trailing]
 582  *
 583  *    For L1(x)=(x-0.5)*(log(x)-1), we need ilogb(L1(x))+5 extra bits for
 584  *    both multiplicants to guarantee L1(x)'s absolute error is bounded by
 585  *    2**(-P-5) in absolute value. Here ilogb(y) is defined to be the unbias
 586  *    binary exponent of y in IEEE format.  We can get x-0.5 to the desire
 587  *    accuracy easily. It remains to compute log(x)-1 with ilogb(L1(x))+5
 588  *    extra bits accracy. Note that the range of L1 is 88.30.., 709.3.., and
```



```
 589  *    11356.10... for single, double, and quadruple precision, we have
 590  *
 591  *                            single     double      quadruple
 592  *                           ------------------------------------
 593  *      ilogb(L1(x))+5 <=     11          14           18
 594  *                           ------------------------------------
 595  *
 596  *    (3) Table Driven Method for log(x)-1:
 597  *    -------------------------------------
 598  *    Let x = 2**n * y, where 1 <= y < 2. Let Z={z(i),i=1,...,m}
 599  *    be a set of predetermined evenly distributed floating point numbers
 600  *    in [1, 2]. Let z(j) be the closest one to y, then
 601  *      log(x)-1 = n*log(2)-1  +  log(y)
 602  *               = n*log(2)-1  +  log(z(j)*y/z(j))
 603  *               = n*log(2)-1  +  log(z(j))  +  log(y/z(j))
 604  *               = T1(n)       +  T2(j)      +  T3,
 605  *
 606  *    where T1(n) = n*log(2)-1 and T2(j) = log(z(j)). Both T1 and T2 can be
 607  *    pre-calculated and be looked-up in a table. Note that 8 <= x < 1756
 608  *    implies 3<=n<=10 implies 1.079.. < T1(n) < 6.931.
 609  *
 610  *
 611  *                        y-z(i)          y       1+s
 612  *    For T3, let s = --------; then ----- =  ----- and
 613  *                        y+z(i)        z(i)     1-s
 614  *                  1+s             2   3    2   5
 615  *      T3 = log(-----) = 2s + --- s  + --- s  + ....
 616  *                  1-s             3        5
 617  *
 618  *    Suppose the first term 2s is compute in extra precision. The
 619  *    dominating error in T3 would then be the rounding error of the
 620  *    second term 2/3*s**3. To force the rounding bounded by
 621  *    the required accuracy, we have
 622  *        single:  |2/3*s**3| < 2**-11   == > |s|<0.09014...
 623  *        double:  |2/3*s**3| < 2**-14   == > |s|<0.04507...
 624  *        quad  :  |2/3*s**3| < 2**-18   == > |s|<0.01788... = 2**(-5.80..)
 625  *
 626  *    Base on this analysis, we choose Z = {z(i)|z(i)=1+i/64+1/128, 0<=i<=63}.
 627  *    For any y in [1,2), let j = [64*y] chopped to integer, then z(j) is
 628  *    the closest to y, and it is not difficult to see that |s| < 2**(-8).
 629  *    Please note that the polynomial approximation of T3 must be accurate
 630  *        -24-11   -35    -53-14    -67          -113-18   -131
 631  *    to 2       =2    ,  2       = 2    ,  and  2        =2
 632  *    for single, double, and quadruple precision respectively.
 633  *
 634  *    Inplementation notes.
 635  *    (1) Table look-up entries for T1(n) and T2(j), as well as the calculation
 636  *        of the leading term 2s in T3,  are broken up into leading and trailing
 637  *        part such that (leading part)* 2**24 will always be an integer. That
 638  *        will guarantee the addition of the leading parts will be exact.
 639  *
 640  *                                  2**(-24)
 641  *                               _________V___________________
 642  *              T1(n):     |_____|_____|_____________________|
 643  *                            _______ _____________________________
 644  *              T2(j):       |_______|_____________________________|
 645  *                              ____ _____________________________
 646  *              2s:            |____|_____________________________|
 647  *                                  _____________________________
 648  *         +    T3(s)-2s:           |_____________________________|
 649  *                      ----------------------------------------
 650  *                        [leading] + [Trailing]
 651  *
 652  *    (2) How to compute 2s accurately.
 653  *        (A) Compute v = 2s to the working precision. If |v| < 2**(-18),
 654  *            stop.
```

```
 655  *        (B) chopped v to 2**(-24): v = ((int)(v*2**24))/2**24
 656  *       (C) 2s = v + (2s - v), where
 657  *                     1
 658  *              2s - v = --- * (2(y-z) - v*(y+z) )
 659  *                     y+z
 660  *                       1
 661  *                    = --- * ( [2(y-z) - v*(y+z)_h ]  - v*(y+z)_l  )
 662  *                     y+z
 663  *          where (y+z)_h = (y+z) rounded to 24 bits by (double)(float),
 664  *          and (y+z)_l = ((z+z)-(y+z)_h)+(y-z).  Note the the quantity
 665  *          in [] is exact.
 666  *                                                        2         4
 667  *    (3) Remez approximation for (T3(s)-2s)/s = T3[0]*s + T3[1]*s + ...:
 668  *       Single precision: 1 term (compute in double precision arithmetic)
 669  *          T3(s) = 2s + S1*s^3, S1 = 0.66667172318485180546936236975392 30
 670  *          Remez error: |T3(s)/s - (2s+S1*s^3)| < 2**(-35.87)
 671  *       Double precision: 3 terms, Remez error is bounded by 2**(-72.40),
 672  *          see "tgamma_log"
 673  *       Quad precision: 7 terms, Remez error is bounded by 2**(-136.54),
 674  *          see "tgammal_log"
 675  *
 676  *   The computation of 0.5*(ln(2pi)-1):
 677  *      0.5*(ln(2pi)-1) =  0.41893853320467274178032973640561763986 14...
 678  *      split 0.5*(ln(2pi)-1) to hln2pi_h + hln2pi_l, where hln2pi_h is the
 679  *      leading 21 bits of the constant.
 680  *          hln2pi_h= 0.418938398361206054687 5
 681  *          hln2pi_l= 1.348434666870928297364056176398612173648e-07
 682  *
 683  *   The computation of 1/x*P(1/x^2) = log(G(x))-(x-.5)(ln(x)-1)-(.5ln(2pi)-1):
 684  *      Let s = 1/x <= 1/8 < 0.125. We have
 685  *      quad precision
 686  *          |GP(s) - s*P(s^2)| <= 2**(-120.6), where
 687  *                               3      5                39
 688  *          GP(s) = GP0*s+GP1*s +GP2*s +... +GP19*s     ,
 689  *       GP0  =   0.0833333333333333333333333333333333172839171301
 690  *                      hex 0x3ffe5555 55555555 55555555 55555548
 691  *       GP1  =  -2.7777777777777777777777777749250121199939942 4104e-0003
 692  *       GP2  =   7.9365079365079365079365655054163823635002088 3243e-0004
 693  *       GP3  =  -5.9523809523809523805729977267932450333924196 1704e-0004
 694  *       GP4  =   8.4175084175084169613842298797768352492614260 0321e-0004
 695  *       GP5  =  -1.9175269175268668282503254782369966217884212 3308e-0003
 696  *       GP6  =   6.4102564102240348092189155935647345116127935 9322e-0003
 697  *       GP7  =  -2.9550653579841401918981958745557700373280818 5071e-0002
 698  *       GP8  =   1.7964436722997003148607918006092307347656873 2136e-0001
 699  *       GP9  =  -1.3924308648727466217456287256705720025564929 0646e+0000
 700  *       GP10 =   1.3402587404441796218867781647784226525960826 9775e+0001
 701  *       GP11 =  -1.5680371348012746941449554539998250870074827 4318e+0002
 702  *       GP12 =   2.1873984165620156169492763033509931396892 4493891e+0003
 703  *       GP13 =  -3.5524984864410033841918703809092541097623792 1269e+0004
 704  *       GP14 =   6.4346488043783528621676895943948437644917957 6452e+0005
 705  *       GP15 =  -1.2045915438557701499260034278282138960589390 4624e+0007
 706  *       GP16 =   2.0926324963735129856393949423497497184910710 93210e+0008
 707  *       GP17 =  -2.9624748318316921934374531643389959983468570 3457e+0009
 708  *       GP18 =   2.8898493360589603315472762608650675697232 7292981e+0010
 709  *       GP19 =  -1.4096043414603000773283838241623061030267806 3984e+0011
 710  *
 711  *       double precision
 712  *          |GP(s) - s*P(s^2)| <= 2**(-63.5), where
 713  *                               3      5      7      9      11      13      15
 714  *          GP(s) = GP0*s+GP1*s +GP2*s +GP3*s +GP4*s +GP5*s  +GP6*s  +GP7*s  ,
 715  *
 716  *              GP0=  0.083333333333333328707404 0640618477 (3FB55555 55555555)
 717  *              GP1= -2.777777777766493552005656111146276700891307728 43e-0003
 718  *              GP2=  7.936507874860837248054761941702117757841585515 09e-0004
 719  *              GP3= -5.952366285583149287578114195802812945939035829 71e-0004
 720  *              GP4=  8.415664739998534519831371627804278127811789325 40e-0004
```



```
 721  *              GP5= -1.904247766704413735645129420389261681759213032 12e-0003
 722  *              GP6=  5.849331615309496663123339495344823030075429917 8e-0003
 723  *              GP7= -1.594532289310820302621248325061443924965616945 50e-0002
 724  *       single precision
 725  *          |GP(s) - s*P(s^2)| <= 2**(-37.78), where
 726  *                               3      5
 727  *          GP(s) = GP0*s+GP1*s +GP2*s
 728  *        GP0 =   8.33333330959694065245736888749042811909994573178e-0002
 729  *        GP1 =  -2.77765545601667179767706600890361535225507762168e-0003
 730  *        GP2 =   7.77830853479775281781085278324621033523037489883e-0004
 731  *
 732  *
 733  *      Implementation note:
 734  *      z = (1/x), z2 = z*z, z4 = z2*z2;
 735  *      p = z*(GP0+z2*(GP1+....+z2*GP7))
 736  *        = z*(GP0+(z4*(GP2+z4*(GP4+z4*GP6))+z2*(GP1+z4*(GP3+z4*(GP5+z4*GP7)))))
 737  *
 738  *   Adding everything up:
 739  *      t = rr.h*ww.h+hln2pi_h                  ... exact
 740  *      w = (hln2pi_l + ((x-0.5)*ww.l+rr.l*ww.h)) + p
 741  *
 742  *   Computing exp(t+w):
 743  *      s = t+w; write s = (n+j/32)*ln2+r, |r|<=(1/64)*ln2, then
 744  *      exp(s) = 2**n * (2**(j/32) + 2**(j/32)*expm1(r)), where
 745  *      expm1(r) = r + Et1*r^2 + Et2*r^3 + ... + Et5*r^6, and
 746  *      2**(j/32) is obtained by table look-up S[j]+S_trail[j].
 747  *      Remez error bound:
 748  *      |exp(r) - (1+r+Et1*r^2+...+Et5*r^6)| <= 2^(-63).
 749  */

 751 #include "libm.h"

 753 #define __HI(x) ((int *) &x)[HIWORD]
 754 #define __LO(x) ((unsigned *) &x)[LOWORD]

 756 struct Double {
 757         double h;
 758         double l;
 759 };
_____unchanged_portion_omitted_
```

```
**********************************************************
   15234 Tue Nov  4 19:01:47 2014
new/usr/src/lib/libm/common/m9x/tgammaf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #pragma weak __tgammaf = tgammaf
  30 #pragma weak tgammaf = __tgammaf

  32 /*
  33  * True gamma function
  34  *
  35  * float tgammaf(float x)
  36  *
  37  * Algorithm: see tgamma.c
  38  *
  39  * Maximum error observed: 0.87ulp (both positive and negative arguments)
  40  */

  42 #include "libm.h"
  43 #include "libm_synonyms.h"
  43 #include <math.h>
  44 #if defined(__SUNPRO_C)
  45 #include <sunmath.h>
  46 #endif
  47 #include <sys/isa_defs.h>

  49 #if defined(_BIG_ENDIAN)
  50 #define HIWORD  0
  51 #define LOWORD  1
  52 #else
  53 #define HIWORD  1
  54 #define LOWORD  0
  55 #endif
  56 #define __HI(x) ((int *) &x)[HIWORD]
  57 #define __LO(x) ((unsigned *) &x)[LOWORD]

  59 /* Coefficients for primary intervals GTi() */
```



```
  60 static const double cr[] = {
  61         /* p1 */
  62         +7.09087253435088360271451613398019280077561279443e-0001,
  63         -5.17229560788652108545141978238701790105241761089e-0001,
  64         +5.23403394528150789405825222323770647162337764327e-0001,
  65         -4.54586308717075010784041566069480411732634814899e-0001,
  66         +4.20596490915239085459964590559256913498190955233e-0001,
  67         -3.57307589712377520978332185838241458642142185789e-0001,

  69         /* p2 */
  70         +4.28486983980295198166056119223984284434264344578e-0001,
  71         -1.30704539487709138528680121627899735386650103914e-0001,
  72         +1.60856285038051955072861219352655851542955430871e-0001,
  73         -9.22285161346010583774458802067371182158937943507e-0002,
  74         +7.19240511767225260740890292605070595560626179357e-0002,
  75         -4.88158265593355093703112238534484636193260459574e-0002,

  77         /* p3 */
  78         +3.82409531118807759081121479786092134814808872880e-0001,
  79         +2.65309888180188647956400403013495759365167853426e-0002,
  80         +8.06815109775079171923561169415370309376296739835e-0002,
  81         -1.54821591666137613928840890835174351674007764799e-0002,
  82         +1.76308239242717268530498313416899188157165183405e-0002,

  84         /* GZi and TZi */
  85         +0.93820462790968244940975356158032695769 88,    /* GZ1 */
  86         +0.88560319441088870027881590058258873320 80,    /* GZ2 */
  87         +0.93678141146365232161884689708083784974 26,    /* GZ3 */
  88         -0.35172143578529357910156 25,    /* TZ1 */
  89         +0.28053063154220581054687 5,    /* TZ3 */
  90 };
______unchanged_portion_omitted_
```

```
**********************************************************
   40060 Tue Nov  4 19:01:47 2014
new/usr/src/lib/libm/common/m9x/tgammal.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __tgammal = tgammal
#pragma weak tgammal = __tgammal

#include "libm.h"
#include <sys/isa_defs.h>

#if defined(_BIG_ENDIAN)
#define H0_WORD(x)      ((unsigned *) &x)[0]
#define H3_WORD(x)      ((unsigned *) &x)[3]
#define CHOPPED(x)      (long double) ((double) (x))
#else
#define H0_WORD(x)      ((((int *) &x)[2] << 16) | \
                        (0x0000ffff & (((unsigned *) &x)[1] >> 15)))
#define H3_WORD(x)      ((unsigned *) &x)[0]
#define CHOPPED(x)      (long double) ((float) (x))
#endif

struct LDouble {
        long double h, l;
};
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    1772 Tue Nov  4 19:01:48 2014
new/usr/src/lib/libm/common/m9x/trunc.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __trunc = trunc
#pragma weak trunc = __trunc

#include "libm.h"

double
trunc(double x) {
        union {
                unsigned i[2];
                double d;
        } xx;
        unsigned hx, sx, i;

        xx.d = x;
        hx = xx.i[HIWORD] & ~0x80000000;
        sx = xx.i[HIWORD] & 0x80000000;
        if (hx < 0x43300000) {  /* |x| < 2^52 */
                if (hx < 0x3ff00000)    /* |x| < 1 */
                        return (sx ? -0.0 : 0.0);

                /* chop x at the integer bit */
                if (hx < 0x41300000) {
                        i = 1 << (0x412 - (hx >> 20));
                        xx.i[HIWORD] &= ~(i | (i - 1));
                        xx.i[LOWORD] = 0;
                } else {
                        i = 1 << (0x432 - (hx >> 20));
                        xx.i[LOWORD] &= ~(i | (i - 1));
                }
                return (xx.d);
        } else if (hx < 0x7ff00000)
                return (x);
```



```
        else
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (hx >= 0x7ff80000 ? x : x + x);
                /* assumes sparc-like QNaN */
#else
                return (x + x);
#endif
}
```

_____unchanged_portion_omitted_

```
**********************************************************
    1610 Tue Nov  4 19:01:48 2014
new/usr/src/lib/libm/common/m9x/truncf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __truncf = truncf
#pragma weak truncf = __truncf

#include "libm.h"

float
truncf(float x) {
        union {
                unsigned i;
                float f;
        } xx;
        unsigned hx, sx, i;

        xx.f = x;
        hx = xx.i & ~0x80000000;
        sx = xx.i & 0x80000000;
        if (hx < 0x4b000000) {          /* |x| < 2^23 */
                if (hx < 0x3f800000)            /* |x| < 1 */
                        return (sx ? -0.0F : 0.0F);

                /* chop x at the integer bit */
                i = 1 << (0x95 - (hx >> 23));
                xx.i &= ~((i << 1) - 1);
                return (xx.f);
        } else if (hx < 0x7f800000)     /* |x| is integral */
                return (x);
        else
#if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
                return (hx > 0x7f800000 ? x * x : x + x);
#else
                return (x + x);
#endif
```



```
}
______unchanged_portion_omitted_
```

```
**********************************************************
    2668 Tue Nov  4 19:01:49 2014
new/usr/src/lib/libm/common/m9x/truncl.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma weak __truncl = truncl
#pragma weak truncl = __truncl

#include "libm.h"

#if defined(__sparc)
long double
truncl(long double x) {
        union {
                unsigned i[4];
                long double q;
        } xx;
        unsigned hx, sx;
        int j;

        xx.q = x;
        sx = xx.i[0] & 0x80000000;
        hx = xx.i[0] & ~0x80000000;

        /* handle trivial cases */
        if (hx >= 0x406f0000) /* |x| >= 2^112 + ... or x is nan */
                return (hx >= 0x7fff0000 ? x + x : x);

        /* handle |x| < 1 */
        if (hx < 0x3fff0000)
                return (sx ? -0.0L : 0.0L);

        j = 0x406f - (hx >> 16);                /* 1 <= j <= 112 */
        xx.i[0] = hx;
        if (j >= 96) {                          /* 96 <= j <= 112 */
                xx.i[0] &= ~((1 << (j - 96)) - 1);
                xx.i[1] = xx.i[2] = xx.i[3] = 0;
```



```
        } else if (j >= 64) {                   /* 64 <= j <= 95 */
                xx.i[1] &= ~((1 << (j - 64)) - 1);
                xx.i[2] = xx.i[3] = 0;
        } else if (j >= 32) {                   /* 32 <= j <= 63 */
                xx.i[2] &= ~((1 << (j - 32)) - 1);
                xx.i[3] = 0;
        } else                                  /* 1 <= j <= 31 */
                xx.i[3] &= ~((1 << j) - 1);

        /* negate result if need be */
        if (sx)
                xx.i[0] |= 0x80000000;
        return (xx.q);
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   12923 Tue Nov  4 19:01:49 2014
new/usr/src/lib/libm/common/mapfile-vers
5271 matherr(3M) is designed for interposition, should not be direct bound
5261 libm should stop using synonyms.h
**********************************************************
______unchanged_portion_omitted_

  90 SYMBOL_VERSION SUNW_1.2 {
  91         global:
  92                 __acoshf;
  93                 __acoshl;
  94                 __asinhf;
  95                 __asinhl;
  96                 __atanhf;
  97                 __atanhl;
  98                 __cabs;
  99                 __cabsf;
 100                 __cabsl;
 101                 __cacos;
 102                 __cacosf;
 103                 __cacosh;
 104                 __cacoshf;
 105                 __cacoshl;
 106                 __cacosl;
 107                 __carg;
 108                 __cargf;
 109                 __cargl;
 110                 __casin;
 111                 __casinf;
 112                 __casinh;
 113                 __casinhf;
 114                 __casinhl;
 115                 __casinl;
 116                 __catan;
 117                 __catanf;
 118                 __catanh;
 119                 __catanhf;
 120                 __catanhl;
 121                 __catanl;
 122                 __cbrtf;
 123                 __cbrtl;
 124                 __ccos;
 125                 __ccosf;
 126                 __ccosh;
 127                 __ccoshf;
 128                 __ccoshl;
 129                 __ccosl;
 130                 __cexp;
 131                 __cexpf;
 132                 __cexpl;
 133                 __cimag;
 134                 __cimagf;
 135                 __cimagl;
 136                 __clog;
 137                 __clogf;
 138                 __clogl;
 139                 __conj;
 140                 __conjf;
 141                 __conjl;
 142                 __copysignf;
 143                 __copysignl;
 144                 __cpow;
 145                 __cpowf;
 146                 __cpowl;
 147                 __cproj;
```



```
 148                 __cprojf;
 149                 __cprojl;
 150                 __creal;
 151                 __crealf;
 152                 __creall;
 153                 __csin;
 154                 __csinf;
 155                 __csinh;
 156                 __csinhf;
 157                 __csinhl;
 158                 __csinl;
 159                 __csqrt;
 160                 __csqrtf;
 161                 __csqrtl;
 162                 __ctan;
 163                 __ctanf;
 164                 __ctanh;
 165                 __ctanhf;
 166                 __ctanhl;
 167                 __ctanl;
 168                 __erfcf;
 169                 __erfcl;
 170                 __erff;
 171                 __erfl;
 172                 __exp2;
 173                 __exp2f;
 174                 __exp2l;
 175                 __expm1f;
 176                 __expm1l;
 177                 __fdim;
 178                 __fdimf;
 179                 __fdiml;
 180                 __feclearexcept;
 181                 __fegetenv;
 182                 __fegetexceptflag;
 183 $if _x86
 184                 __fegetprec;                    #LSARC/1996/175
 185 $endif
 186                 __fegetround;
 187                 __feholdexcept;
 188                 __fenv_dfl_env;                 #LSARC/1996/175
 189                 __feraiseexcept;
 190                 __fesetenv;
 191                 __fesetexceptflag;
 192 $if _x86
 193                 __fesetprec;                    #LSARC/1996/175
 194 $endif
 195                 __fesetround;
 196                 __fetestexcept;
 197                 __feupdateenv;
 198                 __fex_get_handling;             #LSARC/1996/175
 199                 __fex_get_log;                  #LSARC/1996/175
 200                 __fex_get_log_depth;            #LSARC/1996/175
 201                 __fex_getexcepthandler;         #LSARC/1996/175
 202                 __fex_log_entry;                #LSARC/1996/175
 203                 __fex_merge_flags;              #LSARC/1996/175
 204                 __fex_set_handling;             #LSARC/1996/175
 205                 __fex_set_log;                  #LSARC/1996/175
 206                 __fex_set_log_depth;            #LSARC/1996/175
 207                 __fex_setexcepthandler;         #LSARC/1996/175
 208                 __fma;
 209                 __fmaf;
 210                 __fmal;
 211                 __fmax;
 212                 __fmaxf;
 213                 __fmaxl;
```

```
214                 __fmin;
215                 __fminf;
216                 __fminl;
217                 __frexp;
218                 __gammaf;                       #LSARC/2003/279
219                 __gammaf_r;                     #LSARC/2003/279
220                 __gammal;                       #LSARC/2003/279
221                 __gammal_r;                     #LSARC/2003/279
222                 __hypotf;
223                 __hypotl;
224                 __ilogbf;
225                 __ilogbl;
226                 __j0f;                          #LSARC/2003/279
227                 __j0l;                          #LSARC/2003/279
228                 __j1f;                          #LSARC/2003/279
229                 __j1l;                          #LSARC/2003/279
230                 __jnf;                          #LSARC/2003/279
231                 __jnl;                          #LSARC/2003/279
232                 __ldexp;
233                 __lgammaf;
234                 __lgammaf_r;                    #LSARC/2003/279
235                 __lgammal;
236                 __lgammal_r;                    #LSARC/2003/279
237 $if amd64 || sparcv9
238                 __llrint        { FLAGS = NODYNSORT };
239                 __llrintf       { FLAGS = NODYNSORT };
240                 __llrintl       { FLAGS = NODYNSORT };
241                 __llround       { FLAGS = NODYNSORT };
242                 __llroundf      { FLAGS = NODYNSORT };
243                 __llroundl      { FLAGS = NODYNSORT };
244 $else
245 #endif /* ! codereview */
246                 __llrint;
247                 __llrintf;
248                 __llrintl;
249                 __llround;
250                 __llroundf;
251                 __llroundl;
252 $endif
253 #endif /* ! codereview */
254                 __log1pf;
255                 __log1pl;
256                 __log2;
257                 __log2f;
258                 __log2l;
259                 __logbf;
260                 __logbl;
261 $if amd64 || sparcv9
262                 __lrint         { FLAGS = NODYNSORT };
263                 __lrintf        { FLAGS = NODYNSORT };
264                 __lrintl        { FLAGS = NODYNSORT };
265                 __lround        { FLAGS = NODYNSORT };
266                 __lroundf       { FLAGS = NODYNSORT };
267                 __lroundl       { FLAGS = NODYNSORT };
268 $else
269                 __lrint;
270                 __lrintf;
271                 __lrintl;
272                 __lround;
273                 __lroundf;
274                 __lroundl;
275 $endif
276                 __modf;
277                 __nan;
278                 __nanf;
279                 __nanl;
```



```
280                 __nearbyint;
281                 __nearbyintf;
282                 __nearbyintl;
283                 __nextafterf;
284                 __nextafterl;
285                 __nexttoward;
286                 __nexttowardf;
287                 __nexttowardl;
288                 __remainderf;
289                 __remainderl;
290                 __remquo;
291                 __remquof;
292                 __remquol;
293                 __rintf;
294                 __rintl;
295                 __round;
296                 __roundf;
297                 __roundl;
298                 __scalbf;                       #LSARC/2003/279
299                 __scalbl;                       #LSARC/2003/279
300                 __scalbln;
301                 __scalblnf;
302                 __scalblnl;
303                 __scalbnf;
304                 __scalbnl;
305                 __signgamf;                     #LSARC/2003/279
306                 __signgaml;                     #LSARC/2003/279
307                 __significandf;                 #LSARC/2003/279
308                 __significandl;                 #LSARC/2003/279
309                 __sincos;                       #LSARC/2003/279
310                 __sincosf;                      #LSARC/2003/279
311                 __sincosl;                      #LSARC/2003/279
312                 __tgamma;
313                 __tgammaf;
314                 __tgammal;
315                 __trunc;
316                 __truncf;
317                 __truncl;
318                 __xpg6 { FLAGS = NODIRECT };    #private contract with libc grou
319                 __y0f;                          #LSARC/2003/279
320                 __y0l;                          #LSARC/2003/279
321                 __y1f;                          #LSARC/2003/279
322                 __y1l;                          #LSARC/2003/279
323                 __ynf;                          #LSARC/2003/279
324                 __ynl;                          #LSARC/2003/279
325                 acosf;
326                 acoshf;
327                 acoshl;
328                 acosl;
329                 asinf;
330                 asinhf;
331                 asinhl;
332                 asinl;
333                 atan2f;
334                 atan2l;
335                 atanf;
336                 atanhf;
337                 atanhl;
338                 atanl;
339                 cabs;
340                 cabsf;
341                 cabsl;
342                 cacos;
343                 cacosf;
344                 cacosh;
345                 cacoshf;
```

```
346                 cacoshl;
347                 cacosl;
348                 carg;
349                 cargf;
350                 cargl;
351                 casin;
352                 casinf;
353                 casinh;
354                 casinhf;
355                 casinhl;
356                 casinl;
357                 catan;
358                 catanf;
359                 catanh;
360                 catanhf;
361                 catanhl;
362                 catanl;
363                 cbrtf;
364                 cbrtl;
365                 ccos;
366                 ccosf;
367                 ccosh;
368                 ccoshf;
369                 ccoshl;
370                 ccosl;
371                 ceilf;
372                 ceill;
373                 cexp;
374                 cexpf;
375                 cexpl;
376                 cimag;
377                 cimagf;
378                 cimagl;
379                 clog;
380                 clogf;
381                 clogl;
382                 conj;
383                 conjf;
384                 conjl;
385                 copysignf;
386                 copysignl;
387                 cosf;
388                 coshf;
389                 coshl;
390                 cosl;
391                 cpow;
392                 cpowf;
393                 cpowl;
394                 cproj;
395                 cprojf;
396                 cprojl;
397                 creal;
398                 crealf;
399                 creall;
400                 csin;
401                 csinf;
402                 csinh;
403                 csinhf;
404                 csinhl;
405                 csinl;
406                 csqrt;
407                 csqrtf;
408                 csqrtl;
409                 ctan;
410                 ctanf;
411                 ctanh;
```



```
412                 ctanhf;
413                 ctanhl;
414                 ctanl;
415                 erfcf;
416                 erfcl;
417                 erff;
418                 erfl;
419                 exp2;
420                 exp2f;
421                 exp2l;
422                 expf;
423                 expl;
424                 expm1f;
425                 expm1l;
426                 fabsf;
427                 fabsl;
428                 fdim;
429                 fdimf;
430                 fdiml;
431                 feclearexcept;
432                 fegetenv;
433                 fegetexceptflag;
434 $if _x86
435                 fegetprec;                      #LSARC/1996/175
436 $endif
437                 fegetround;
438                 feholdexcept;
439                 feraiseexcept;
440                 fesetenv;
441                 fesetexceptflag;
442 $if _x86
443                 fesetprec;                      #LSARC/1996/175
444 $endif
445                 fesetround;
446                 fetestexcept;
447                 feupdateenv;
448                 fex_get_handling;               #LSARC/1996/175
449                 fex_get_log;                    #LSARC/1996/175
450                 fex_get_log_depth;              #LSARC/1996/175
451                 fex_getexcepthandler;           #LSARC/1996/175
452                 fex_log_entry;                  #LSARC/1996/175
453                 fex_merge_flags;                #LSARC/1996/175
454                 fex_set_handling;               #LSARC/1996/175
455                 fex_set_log;                    #LSARC/1996/175
456                 fex_set_log_depth;              #LSARC/1996/175
457                 fex_setexcepthandler;           #LSARC/1996/175
458                 floorf;
459                 floorl;
460                 fma;
461                 fmaf;
462                 fmal;
463                 fmax;
464                 fmaxf;
465                 fmaxl;
466                 fmin;
467                 fminf;
468                 fminl;
469                 fmodf;
470                 fmodl;
471                 frexp;
472                 frexpf;
473                 frexpl;
474                 gammaf;                         #LSARC/2003/279
475                 gammaf_r;                       #LSARC/2003/279
476                 gammal;                         #LSARC/2003/279
477                 gammal_r;                       #LSARC/2003/279
```

```
478                 hypotf;
479                 hypotl;
480                 ilogbf;
481                 ilogbl;
482                 j0f;                            #LSARC/2003/279
483                 j0l;                            #LSARC/2003/279
484                 j1f;                            #LSARC/2003/279
485                 j1l;                            #LSARC/2003/279
486                 jnf;                            #LSARC/2003/279
487                 jnl;                            #LSARC/2003/279
488                 ldexp;
489                 ldexpf;
490                 ldexpl;
491                 lgammaf;
492                 lgammaf_r;                      #LSARC/2003/279
493                 lgammal;
494                 lgammal_r;                      #LSARC/2003/279
495 $if amd64 || sparcv9
496                 llrint          { FLAGS = NODYNSORT };
497                 llrintf         { FLAGS = NODYNSORT };
498                 llrintl         { FLAGS = NODYNSORT };
499                 llround         { FLAGS = NODYNSORT };
500                 llroundf        { FLAGS = NODYNSORT };
501                 llroundl        { FLAGS = NODYNSORT };
502 $else
503                 llrint;
504                 llrintf;
505                 llrintl;
506                 llround;
507                 llroundf;
508                 llroundl;
509 $endif
510                 log10f;
511                 log10l;
512                 log1pf;
513                 log1pl;
514                 log2;
515                 log2f;
516                 log2l;
517                 logbf;
518                 logbl;
519                 logf;
520                 logl;
521                 lrint;
522                 lrintf;
523                 lrintl;
524                 lround;
525                 lroundf;
526                 lroundl;
527                 modf;
528                 modff;
529                 modfl;
530                 nan;
531                 nanf;
532                 nanl;
533                 nearbyint;
534                 nearbyintf;
535                 nearbyintl;
536                 nextafterf;
537                 nextafterl;
538                 nexttoward;
539                 nexttowardf;
540                 nexttowardl;
541                 powf;
542                 powl;
543                 remainderf;
```



```
544                 remainderl;
545                 remquo;
546                 remquof;
547                 remquol;
548                 rintf;
549                 rintl;
550                 round;
551                 roundf;
552                 roundl;
553                 scalbf;                         #LSARC/2003/279
554                 scalbl;                         #LSARC/2003/279
555                 scalbln;
556                 scalblnf;
557                 scalblnl;
558                 scalbnf;
559                 scalbnl;
560                 signgamf;                       #LSARC/2003/279
561                 signgaml;                       #LSARC/2003/279
562                 significandf;                   #LSARC/2003/279
563                 significandl;                   #LSARC/2003/279
564                 sincos;                         #LSARC/2003/279
565                 sincosf;                        #LSARC/2003/279
566                 sincosl;                        #LSARC/2003/279
567                 sinf;
568                 sinhf;
569                 sinhl;
570                 sinl;
571                 sqrtf;
572                 sqrtl;
573                 tanf;
574                 tanhf;
575                 tanhl;
576                 tanl;
577                 tgamma;
578                 tgammaf;
579                 tgammal;
580                 trunc;
581                 truncf;
582                 truncl;
583                 y0f;                            #LSARC/2003/279
584                 y0l;                            #LSARC/2003/279
585                 y1f;                            #LSARC/2003/279
586                 y1l;                            #LSARC/2003/279
587                 ynf;                            #LSARC/2003/279
588                 ynl;                            #LSARC/2003/279
589 } SUNW_1.1.1;

591 SYMBOL_VERSION SUNW_1.1.1 {
592         global:
593                 __acosf;
594                 __acosl;
595                 __asinf;
596                 __asinl;
597                 __atan2f;
598                 __atan2l;
599                 __atanf;
600                 __atanl;
601                 __ceilf;
602                 __ceill;
603                 __cosf;
604                 __coshf;
605                 __coshl;
606                 __cosl;
607                 __expf;
608                 __expl;
609                 __fabsf;
```

```
610                 __fabsl;
611                 __floorf;
612                 __floorl;
613                 __fmodf;
614                 __fmodl;
615                 __frexpf;
616                 __frexpl;
617                 __ldexpf;
618                 __ldexpl;
619                 __log10f;
620                 __log10l;
621                 __logf;
622                 __logl;
623                 __modff;
624                 __modfl;
625                 __powf;
626                 __powl;
627                 __sinf;
628                 __sinhf;
629                 __sinhl;
630                 __sinl;
631                 __sqrtf;
632                 __sqrtl;
633                 __tanf;
634                 __tanhf;
635                 __tanhl;
636                 __tanl;
637 } SUNW_1.1;

639 SYMBOL_VERSION SUNW_1.1 {
640         global:
641                 __acos;
642                 __acosh;
643                 __asin;
644                 __asinh;
645                 __atan;
646                 __atan2;
647                 __atanh;
648                 __cbrt;
649                 __ceil;
650                 __copysign;
651                 __cos;
652                 __cosh;
653                 __erf;
654                 __erfc;
655                 __exp;
656                 __expm1;
657                 __fabs;
658                 __floor;
659                 __fmod;
660                 __gamma;
661                 __gamma_r;
662                 __hypot;
663                 __ilogb;
664                 __isnan;
665                 __j0;
666                 __j1;
667                 __jn;
668                 __lgamma;
669                 __lgamma_r;
670                 __log;
671                 __log10;
672                 __log1p;
673                 __logb;
674                 __nextafter;
675                 __pow;
```



```
676                 __remainder;
677                 __rint;
678                 __scalb;
679                 __scalbn;
680                 __signgam;
681                 __significand;
682                 __sin;
683                 __sinh;
684                 __sqrt;
685                 __tan;
686                 __tanh;
687                 __y0;
688                 __y1;
689                 __yn;
690                 acos;
691                 acosh;
692                 asin;
693                 asinh;
694                 atan;
695                 atan2;
696                 atanh;
697                 cbrt;
698                 ceil;
699                 copysign;
700                 cos;
701                 cosh;
702                 erf;
703                 erfc;
704                 exp;
705                 expm1;
706                 fabs;
707                 floor;
708                 fmod;
709                 gamma;
710                 gamma_r;
711                 hypot;
712                 ilogb;
713                 isnan;
714                 j0;
715                 j1;
716                 jn;
717                 lgamma;
718                 lgamma_r;
719                 log;
720                 log10;
721                 log1p;
722                 logb;
723                 matherr { FLAGS = NODIRECT };
237                 matherr;
724                 nextafter;
725                 pow;
726                 remainder;
727                 rint;
728                 scalb;
729                 scalbn;
730                 signgam;
731                 significand;
732                 sin;
733                 sinh;
734                 sqrt;
735                 tan;
736                 tanh;
737                 y0;
738                 y1;
739                 yn;
740 };
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2397 Tue Nov  4 19:01:50 2014
new/usr/src/lib/libm/i386/src/__reduction.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "__reduction.s"

/
/    After argument reduction which returns n:
/       n mod 4     sin(x)      cos(x)        tan(x)
/     ------------------------------------------------------
/          0          S           C             S/C
/          1          C           -S            -C/S
/          2          -S          -C            S/C
/          3          -C          S             -C/S
/     ------------------------------------------------------

#include "libm.h"
#include "libm_synonyms.h"
#include "libm_protos.h"
#undef fabs

        ENTRY(__reduction)
#ifndef PIC
        movl    12(%esp),%eax           / load the high part of arg
#else
        movl    16(%esp),%eax           / load the high part of arg
#endif
        andl    $0x7fffffff,%eax        / clear sign
        cmpl    $0x3fe921fb,%eax        / Is |x| < pi/4 (= 0x3fe921fb54...) ?
        jbe     .L0
        cmpl    $0x7ff00000,%eax        / Is arg a NaN or an Inf ?
        jb      .L1
.L0:
#ifndef PIC
        fldl    8(%esp)                 / push arg
#else
        fldl    12(%esp)                / push arg
```



```
#endif
        fwait
        movl    $0,%eax                 / set n = 0
        ret
.L1:
        pushl   %ebp
        movl    %esp,%ebp
        subl    $16,%esp
        PIC_SETUP(1)
        leal    -16(%ebp),%eax          / address of y[0]
        pushl   %eax
#ifndef PIC
        pushl   16(%ebp)
        pushl   12(%ebp)
#else
        pushl   20(%ebp)
        pushl   16(%ebp)
#endif
        call    PIC_F(__rem_pio2)       / call __rem_pio2(x,&y)
        fldl    -8(%ebp)                / y[1]
        fldl    -16(%ebp)               / y[0], y[1]
        faddp   %st,%st(1)              / y[0]+y[1] round-to-extended
        addl    $28,%esp                / 16+4*3
        andl    $3,%eax
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(__reduction)
______unchanged_portion_omitted_
```

```
**********************************************************
    2005 Tue Nov  4 19:01:50 2014
new/usr/src/lib/libm/i386/src/acos.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "acos.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(acos,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        ENTRY(acos)
        fldl    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw   %ax
        sahf
        ja      .ERR
        fadd    %st(1),%st              / 1+x,x
        fldz
        fucomp
        fstsw   %ax
        sahf
        jp      .L1
        jne     .L1
        / x is -1
        fstp    %st(0)                  / -1
        fstp    %st(0)                  / empty NPX stack
        fldpi
        ret
.L1:
        fxch    %st(1)                  / x,1+x
        fld1                            / 1,x,1+x
```



```
        fsubp   %st,%st(1)              / 1-x,1+x
        fdivp   %st,%st(1)              / (1-x)/(1+x)
        fsqrt
        fld1                            / 1,sqrt((1-x)/(1+x))
        fpatan
        fadd    %st(0),%st
        ret

.ERR:
        / |x| > 1
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
        pushl   $1
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        call    PIC_F(_SVID_libm_err)   / report SVID result/error
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(acos)
______unchanged_portion_omitted_
```

```
**********************************************************
    1718 Tue Nov  4 19:01:50 2014
new/usr/src/lib/libm/i386/src/acosf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "acosf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(acosf,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        ENTRY(acosf)
        flds    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw   %ax
        sahf
        ja      .ERR
        fadd    %st(1),%st              / 1+x,x
        fldz
        fucomp
        fstsw   %ax
        sahf
        jp      .L1
        jne     .L1
        / x is -1
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
        fldpi
        ret
.L1:
        fxch    %st(1)                  / x,1+x
        fld1                            / 1,x,1+x
```



```
        fsubp   %st,%st(1)              / 1-x,1+x
        fdivp   %st,%st(1)              / (1-x)/(1+x)
        fsqrt
        fld1                            / 1,sqrt((1-x)/(1+x))
        fpatan
        fadd    %st(0),%st
        ret

.ERR:
        / |x| > 1
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
        fldz
        fdiv    %st(0),%st              / 0/0
        ret
        .align  4
        SET_SIZE(acosf)
______unchanged_portion_omitted_
```

```
**********************************************************
    1702 Tue Nov  4 19:01:51 2014
new/usr/src/lib/libm/i386/src/acosl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "acosl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(acosl,function)
#include "libm_synonyms.h"

#undef fabs

        ENTRY(acosl)
        fldt    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw   %ax
        sahf
        ja      9f
        fadd    %st(1),%st              / 1+x,x
        fldz
        fucomp
        fstsw   %ax
        sahf
        jp      .L1
        jne     .L1
        / x is -1
        fstp    %st(0)                  / -1
        fstp    %st(0)                  / empty NPX stack
        fldpi
        ret
.L1:
        fxch    %st(1)                  / x,1+x
        fld1                            / 1,x,1+x
        fsubp   %st,%st(1)              / 1-x,1+x
```



```
        fdivp   %st,%st(1)              / (1-x)/(1+x)
        fsqrt
        fld1                            / 1,sqrt((1-x)/(1+x))
        fpatan
        fadd    %st(0),%st
        ret
9:
        / |x| > 1
        fstp    %st(0)                  / x
        fsub    %st,%st(0)              / +/-0 or NaN+invalid
        fdiv    %st,%st(0)              / NaN+invalid or NaN
        ret
        .align  4
        SET_SIZE(acosl)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1866 Tue Nov  4 19:01:51 2014
new/usr/src/lib/libm/i386/src/asin.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "asin.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(asin,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        ENTRY(asin)
        fldl    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw  %ax
        sahf
        ja      .ERR
        fadd    %st(1),%st              / 1+x,x
        fld1                            / 1,1+x,x
        fsub    %st(2),%st              / 1-x,1+x,x
        fmulp   %st,%st(1)              / (1-x)*(1+x),x
        fsqrt                           / sqrt((1-x)/(1+x)),x
        fpatan                          / atan(x/sqrt((1-x)/(1+x)))
        ret

.ERR:
        / |x| > 1
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
```



```
        pushl   $2
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        call    PIC_F(_SVID_libm_err)   / report SVID result/error
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(asin)
______unchanged_portion_omitted_
```

```
**********************************************************
    1580 Tue Nov  4 19:01:52 2014
new/usr/src/lib/libm/i386/src/asinf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "asinf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(asinf,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        ENTRY(asinf)
        flds    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw  %ax
        sahf
        ja      .ERR
        fadd    %st(1),%st              / 1+x,x
        fld1                            / 1,1+x,x
        fsub    %st(2),%st              / 1-x,1+x,x
        fmulp   %st,%st(1)              / (1-x)*(1+x),x
        fsqrt                           / sqrt((1-x)*(1+x)),x
        fpatan                          / atan(x/sqrt((1-x)*(1+x)))
        ret

.ERR:
        / |x| > 1
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
        fldz
        fdiv    %st(0),%st              / 0/0
        ret
```



```
        .align  4
        SET_SIZE(asinf)
______unchanged_portion_omitted_
```

```
**********************************************************
    1552 Tue Nov  4 19:01:52 2014
new/usr/src/lib/libm/i386/src/asinl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "asinl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(asinl,function)
#include "libm_synonyms.h"

#undef fabs

        ENTRY(asinl)
        fldt    4(%esp)                 / push x
        fld1                            / push 1
        fld     %st(1)                  / x , 1 , x
        fabs                            / |x| , 1 , x
        fucomp
        fstsw   %ax
        sahf
        ja      9f
        fadd    %st(1),%st              / 1+x,x
        fld1                            / 1,1+x,x
        fsub    %st(2),%st              / 1-x,1+x,x
        fmulp   %st,%st(1)              / (1-x)*(1+x),x
        fsqrt                           / sqrt((1-x)*(1+x)),x
        fpatan                          / atan(x/sqrt((1-x)*(1+x)))
        ret
9:
        / |x| > 1
        fstp    %st(0)                  / x
        fsub    %st,%st(0)              / +/-0 or NaN+invalid
        fdiv    %st,%st(0)              / NaN+invalid or NaN
        ret
        .align  4
        SET_SIZE(asinl)
______unchanged_portion_omitted_
```

****
1187 Tue Nov 4 19:01:52 2014
new/usr/src/lib/libm/i386/src/atan.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "atan.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(atan,function)
#include "libm_synonyms.h"

        ENTRY(atan)
        fldl    4(%esp)                 / push arg
        fld1                            / push 1.0
        fpatan                          / atan(arg/1.0)
        ret
        .align  4
        SET_SIZE(atan)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1782 Tue Nov  4 19:01:53 2014
new/usr/src/lib/libm/i386/src/atan2.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file "atan2.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(atan2,function)
  33 #include "libm_synonyms.h"
  33 #include "libm_protos.h"

  35         ENTRY(atan2)
  36         movl    4(%esp),%eax            / low part of y
  37         movl    12(%esp),%ecx           / low part of x
  38         orl     %eax,%ecx
  39         jz      .maybe_0s

  41         / not both x and y are 0's
  42 1:
  43         fldl    4(%esp)                 / push y
  44         fldl    12(%esp)                / push x
  45         fpatan                          / return atan2(y,x)
  46         ret

  48 .maybe_0s:
  49         movl    8(%esp),%eax            / high part of y
  50         movl    16(%esp),%ecx           / high part of x
  51         orl     %eax,%ecx
  52         andl    $0x7fffffff,%ecx        / clear sign
  53         jnz     1b
  54         / both x and y are 0's
  55         pushl   %ebp
  56         movl    %esp,%ebp
  57         PIC_SETUP(1)
  58         pushl   $3
  59         pushl   12(%ebp)                / high y
  60         pushl   8(%ebp)                 / low y
```



```
  61         pushl   20(%ebp)                / high x
  62         pushl   16(%ebp)                / low x
  63         call    PIC_F(_SVID_libm_err)   / report SVID result/error
  64         addl    $20,%esp
  65         PIC_WRAPUP
  66         leave
  67         ret
  68         .align  4
  69         SET_SIZE(atan2)
______unchanged_portion_omitted_
```

```
**********************************************************
    1234 Tue Nov  4 19:01:53 2014
new/usr/src/lib/libm/i386/src/atan2f.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file "atan2f.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(atan2f,function)
  33 #include "libm_synonyms.h"
  33 #include "libm_protos.h"

  35         ENTRY(atan2f)
  36         flds    4(%esp)                 / push y
  37         flds    8(%esp)                 / push x
  38         fpatan                          / return atan2(y,x)
  39         ret
  40         .align  4
  41         SET_SIZE(atan2f)
______unchanged_portion_omitted_
```

****
1209 Tue Nov 4 19:01:54 2014
new/usr/src/lib/libm/i386/src/atan2l.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file "atan2l.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(atan2l,function)
33 #include "libm_synonyms.h"

34         ENTRY(atan2l)
35         fldt    4(%esp)                 / push y
36         fldt    16(%esp)                / push x
37         fpatan                          / return atan2(y,x)
38         ret
39         .align  4
40         SET_SIZE(atan2l)
```

_____***unchanged_portion_omitted_***

****
1191 Tue Nov 4 19:01:54 2014
new/usr/src/lib/libm/i386/src/atanl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "atanl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(atanl,function)
#include "libm_synonyms.h"

        ENTRY(atanl)
        fldt    4(%esp)                 / push arg
        fld1                            / push 1.0
        fpatan                          / atan(arg/1.0)
        ret
        .align  4
        SET_SIZE(atanl)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1440 Tue Nov  4 19:01:55 2014
new/usr/src/lib/libm/i386/src/ceil.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "ceil.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(ceil,function)
#include "libm_synonyms.h"

        ENTRY(ceil)
        subl    $8,%esp
        fstcw   (%esp)
        fldl    12(%esp)
        movw    (%esp),%cx
        orw     $0x0c00,%cx
        xorw    $0x0400,%cx
        movw    %cx,4(%esp)
        fldcw   4(%esp)                 / set RD = up
        frndint
        fstcw   4(%esp)                 / restore RD
        movw    4(%esp),%dx
        andw    $0xf3ff,%dx
        movw    (%esp),%cx
        andw    $0x0c00,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)                  / restore RD
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(ceil)
______unchanged_portion_omitted_
```

```
**********************************************************
    1733 Tue Nov  4 19:01:55 2014
new/usr/src/lib/libm/i386/src/copysign.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "copysign.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(copysign,function)
#include "libm_synonyms.h"

        ENTRY(copysign)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        movl    16(%esp),%ecx           / ecx <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(abs(x))
        andl    $0x80000000,%ecx        / ecx[31] <-- sign_bit(y)
        orl     %ecx,%eax               / eax <-- hi_32(copysign(x,y))
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
                                        /       = lo_32(copysign(x,y))
        subl    $8,%esp                 / set up loading dock for result
        movl    %ecx,(%esp)             / copy lo_32(result) to loading dock
        movl    %eax,4(%esp)            / copy hi_32(result) to loading dock
        fldl    (%esp)                  / load copysign(x,y)
        fwait                           / in case fldl causes exception
        addl    $8,%esp                 / restore stack-pointer for return
        ret
        .align  4
        SET_SIZE(copysign)
______unchanged_portion_omitted_
```

****
1572 Tue Nov 4 19:01:56 2014
new/usr/src/lib/libm/i386/src/copysignf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "copysignf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(copysignf,function)
#include "libm_synonyms.h"

        ENTRY(copysignf)
        movl    4(%esp),%eax            / eax <-- x
        movl    8(%esp),%ecx            / ecx <-- y
        andl    $0x7fffffff,%eax        / eax <-- abs(x)
        andl    $0x80000000,%ecx        / ecx[31] <-- sign_bit(y)
        orl     %ecx,%eax               / eax <-- copysign(x,y)
        subl    $4,%esp                 / set up loading dock for result
        movl    %eax,(%esp)             / copy result to loading dock
        flds    (%esp)                  / load copysign(x,y)
        fwait                           / in case fldl causes exception
        addl    $4,%esp                 / restore stack-pointer for return
        ret
        .align  4
        SET_SIZE(copysignf)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1821 Tue Nov  4 19:01:56 2014
new/usr/src/lib/libm/i386/src/copysignl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "copysignl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(copysignl,function)
#include "libm_synonyms.h"

        ENTRY(copysignl)
        movl    12(%esp),%eax           / sign and bexp of x
        movl    24(%esp),%ecx           / sign and bexp of y
        andl    $0x00007fff,%eax        / eax <-- bexp(x)
        andl    $0x00008000,%ecx        / ecx <-- sign(y)
        orl     %ecx,%eax               / eax <-- bexp(x) with sign(y)
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        movl    4(%esp),%edx            / edx <-- lo_32(sgnfcnd(x))
        subl    $12,%esp                / set up loading dock for result
        movl    %edx,(%esp)             / copy lo_32(result's sgnfcnd)
                                        / to loading dock
        movl    %ecx,4(%esp)            / copy hi_32(result's sgnfcnd)
                                        / to loading dock
        movl    %eax,8(%esp)            / copy sign&bexp(result)
                                        / to loading dock
        fldt    (%esp)                  / load copysign(x,y)
        addl    $12,%esp                / restore stack-pointer for return
        ret
        .align  4
        SET_SIZE(copysignl)
______unchanged_portion_omitted_
```

****
1322 Tue Nov 4 19:01:56 2014
new/usr/src/lib/libm/i386/src/cos.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "cos.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(cos,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(cos)
        PIC_SETUP(1)
        call    PIC_F(__reduction)
        PIC_WRAPUP
        cmpl    $1,%eax
        jl      .cos0
        je      .cos1
        cmpl    $2,%eax
        je      .cos2
        fsin
        ret
.cos2:
        fcos
        fchs
        ret
.cos1:
        fsin
        fchs
        ret
.cos0:
        fcos
        ret
        .align  4
        SET_SIZE(cos)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    4254 Tue Nov  4 19:01:57 2014
new/usr/src/lib/libm/i386/src/exp.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(exp)
        movl    8(%esp),%ecx            / ecx <-- hi_32(x)
        andl    $0x7fffffff,%ecx        / ecx <-- hi_32(|x|)
        cmpl    $0x3fe62e42,%ecx        / Is |x| < ln(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be only slightly < ln(2)
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) >= hi_32(INF)?
        jae     .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, ln(2) < |x| < INF
        fldl    4(%esp)                 / push x
        subl    $8,%esp
                                        /// overhead of RP save/restore; 63/15
        fstcw   (%esp)                  /// ; 15/3
        movw    (%esp),%ax              /// ; 4/1
        movw    %ax,4(%esp)             /// save old RP; 2/1
        orw     $0x0300,%ax             /// force 64-bit RP; 2/1
        movw    %ax,(%esp)              /// ; 2/1
        fldcw   (%esp)                  /// ; 19/4
        fldl2e                          / push log2e   }not for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2e  }not for xtndd_dbl
        fld     %st(0)                  / duplicate stack top
        frndint                         / [z],z
        fucom                           / This and the next 3 instructions
        fstsw  %ax                      / add 10 clocks to runtime of the
        sahf                            / main branch, but save about 265
        je      .z_integral             / upon detection of integral z.
```



```
        / [z] != z, compute exp(x)
        fxch                            / z,[z]
        fsub    %st(1),%st              / z-[z],[z]
        f2xm1                           / 2**(z-[z])-1,[z]
        fld1                            / 1,2**(z-[z])-1,[z]
        faddp   %st,%st(1)              /   2**(z-[z])  ,[z]
.merge:
        fscale                          /   exp(x)      ,[z]
        fstp    %st(1)
        fstcw   (%esp)                  / restore RD
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)                  /// restore old RP; 19/4
        fstpl   (%esp)                  / round to double
        fldl    (%esp)                  / exp(x) rounded to double
        fxam                            / determine class of exp(x)
        add     $8,%esp
        fstsw   %ax                     / store status in ax
        andw    $0x4500,%ax
        cmpw    $0x0500,%ax
        je      .overflow
        cmpw    $0x4000,%ax
        je      .underflow
        ret

.overflow:
        fstp    %st(0)                  / stack empty
        push    %ebp
        mov     %esp,%ebp
        PIC_SETUP(1)
        pushl   $6
        jmp     .error

.underflow:
        fstp    %st(0)                  / stack empty
        push    %ebp
        mov     %esp,%ebp
        PIC_SETUP(2)
        pushl   $7

.error:
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret

.z_integral:                            / here, z is integral
        fstp    %st(0)                  / ,z
        fld1                            / 1,z
        jmp     .merge

.check_tail:
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0xfefa39ef,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x*log2(e)| < 1,
```

```
 127         / whence z is in f2xm1's domain.
 128         fldl    4(%esp)                 / push x
 129         fldl2e                          / push log2e  }not for xtndd_dbl
 130         fmulp   %st,%st(1)              / z = x*log2e }not for xtndd_dbl
 131         f2xm1                           / 2**(x*log2(e))-1 = e**x - 1
 132         fld1                            / 1,2**(z)-1
 133         faddp   %st,%st(1)              /   2**(z) = e**x
 134         ret

 136 .not_finite:
 137         / Here, flags still have settings from execution of
 138         /       cmpl    $0x7ff00000,%ecx        / hi_32(|x|) > hi_32(INF)?
 139         ja      .NaN_or_pinf            / if not, x may be +/- INF
 140         movl    4(%esp),%edx            / edx <-- lo_32(x)
 141         cmpl    $0,%edx                 / lo_32(x) = 0?
 142         jne     .NaN_or_pinf            / if not, x is NaN
 143         movl    8(%esp),%eax            / eax <-- hi_32(x)
 144         andl    $0x80000000,%eax        / here, x is infinite, but +/-?
 145         jz      .NaN_or_pinf            / branch if x = +INF
 146         fldz                            / Here, x = -inf, so return 0
 147         ret

 149 .NaN_or_pinf:
 150         / Here, x = NaN or +inf, so load x and return immediately.
 151         fldl    4(%esp)
 152         fwait
 153         ret
 154         .align  4
 155         SET_SIZE(exp)
```

unchanged_portion_omitted_

```
**********************************************************
    3528 Tue Nov  4 19:01:57 2014
new/usr/src/lib/libm/i386/src/exp10.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp10.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp10,function)
#include "libm_synonyms.h"

        ENTRY(exp10)
        movl    8(%esp),%ecx            / ecx <-- hi_32(x)
        andl    $0x7fffffff,%ecx        / ecx <-- hi_32(|x|)
        cmpl    $0x3fd34413,%ecx        / Is |x| < log10(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / maybe |x| only slightly < log10(2)
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) >= hi_32(INF)?
        jae     .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, log10(2) < |x| < INF
        fldl    4(%esp)                 / push x (=arg)

        subl    $8,%esp                 / save RP and set round-to-64-bits
        fstcw   (%esp)
        movw    (%esp),%ax
        movw    %ax,4(%esp)
        orw     $0x0300,%ax
        movw    %ax,(%esp)
        fldcw   (%esp)

        fldl2t                          / push log2(10)  }NOT for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2(10) }NOT for xtndd_dbl
        fld     %st(0)                  / duplicate stack top
        frndint                         / [z],z
        fucom                           / z integral?
        fstsw  %ax
        sahf
        je      .z_integral             / branch if z integral
```



```
        fxch                            / z, [z]
        fsub    %st(1),%st              / z-[z], [z]
        f2xm1                           / 2**(z-[z])-1, [z]
        fld1                            / 1,2**(z-[z])-1, [z]
        faddp   %st,%st(1)              / 2**(z-[z]), [z]
        fscale                          / 2**z = 10**(arg), [z]
        fstp    %st(1)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.z_integral:                            / here, z is integral
        fstp    %st(0)                  / ,z
        fld1                            / 1 = 2**0, z
        fscale                          / 2**(0 + z) = 2**z = 10**(arg), z
        fstp    %st(1)                  / 10**(arg)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.check_tail:
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0x509f79fe,%edx        / Is |x| slightly > log10(2)?
        ja      .finite_non_special     / branch if |x| slightly > log10(2)
.shortcut:
        / Here, |x| < log10(2), so |z| = |x*log2(10)| < 1
        / whence z is in f2xm1's domain.
        fldl    4(%esp)                 / push x (=arg)
        fldl2t                          / push log2(10)  }NOT for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2(10) }NOT for xtndd_dbl
        f2xm1                           / 2**z - 1
        fld1                            / 1,2**z - 1
        faddp   %st,%st(1)              / 2**z = 10**x
        ret

.not_finite:
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) > hi_32(INF)?
        ja      .NaN_or_pinf            / if so, x is NaN
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x80000000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
```

```
126         / Here, x = NaN or +inf, so load x and return immediately.
127         fldl    4(%esp)
128         fwait
129         ret
130         .align  4
131         SET_SIZE(exp10)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3069 Tue Nov  4 19:01:58 2014
new/usr/src/lib/libm/i386/src/exp10f.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp10f.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp10f,function)
#include "libm_synonyms.h"

        ENTRY(exp10f)
        movl    4(%esp),%ecx            / ecx <-- x
        andl    $0x7fffffff,%ecx        / ecx <-- |x|
        cmpl    $0x3e9a209a,%ecx        / Is |x| < log10(2)?
        jbe     .shortcut               / If so, take a shortcut.
        cmpl    $0x7f800000,%ecx        / |x| >= INF?
        jae     .not_finite             / if so, x is not finite
        flds    4(%esp)                 / push x (=arg)

        subl    $8,%esp                 / save RP and set round-to-64-bits
        fstcw   (%esp)
        movw    (%esp),%ax
        movw    %ax,4(%esp)
        orw     $0x0300,%ax
        movw    %ax,(%esp)
        fldcw   (%esp)

        fldl2t                          / push log2(10)  }NOT for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2(10) }NOT for xtndd_dbl
        fld     %st(0)                  / duplicate stack top
        frndint                         / [z],z
        fucom                           / z integral?
        fstsw   %ax
        sahf
        je      .z_integral             / branch if z integral
        fxch                            / z, [z]
        fsub    %st(1),%st              / z-[z], [z]
```



```
        f2xm1                           / 2**(z-[z])-1, [z]
        fld1                            / 1,2**(z-[z])-1, [z]
        faddp   %st,%st(1)              / 2**(z-[z]), [z]
        fscale                          / 2**z = 10**(arg), [z]
        fstp    %st(1)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.z_integral:                            / here, z is integral
        fstp    %st(0)                  / ,z
        fld1                            / 1 = 2**0, z
        fscale                          / 2**(0 + z) = 2**z = 10**(arg), z
        fstp    %st(1)                  / 10**(arg)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.shortcut:
        / Here, |x| < log10(2), so |z| = |x*log2(10)| < 1
        / whence z is in f2xm1's domain.
        flds    4(%esp)                 / push x (=arg)
        fldl2t                          / push log2(10)  }NOT for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2(10) }NOT for xtndd_dbl
        f2xm1                           / 2**z - 1
        fld1                            / 1,2**z - 1
        faddp   %st,%st(1)              / 2**z = 10**x
        ret

.not_finite:
        ja      .NaN_or_pinf            / branch if x is NaN
        movl    4(%esp),%eax            / eax <-- x
        andl    $0x80000000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        flds    4(%esp)
        fwait
        ret
        .align  4
        SET_SIZE(exp10f)
______unchanged_portion_omitted_
```

```
**********************************************************
    3473 Tue Nov  4 19:01:58 2014
new/usr/src/lib/libm/i386/src/exp10l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp10l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp10l,function)
#include "libm_synonyms.h"

        .data
        .align  4
lt2_hi: .long   0xfbd00000, 0x9a209a84, 0x00003ffd
lt2_lo: .long   0x653f4837, 0x8677076a, 0x0000bfc9

        ENTRY(exp10l)
        movl    12(%esp),%ecx           / cx <--sign&bexp(x)
        andl    $0x00007fff,%ecx        / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x00003ffd,%ecx        / Is |x| < log10(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / maybe |x| only slightly < log10(2)
        cmpl    $0x00007fff,%ecx        / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        cmpl    $0x0000400e,%ecx        / |x| < 32768 = 2^15?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    4(%esp)                 / x
        fld1                            / 1, x
        jmp     1f
.finite_non_special:                    / Here, log10(2) < |x| < 2^15
        fldt    4(%esp)                 / x
        fld     %st(0)                  / x, x
        fldl2t                          / log2(10), x, x
        fmulp                           / z := x*log2(10), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
        fldt    PIC_L(lt2_hi)           / lt2_hi, [z], x, [z]
```



```
        fmulp                           / [z]*lt2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*lt2_hi, [z]
        fldt    PIC_L(lt2_lo)           / lt2_lo, x-[z]*lt2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*lt2_lo, x-[z]*lt2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*log10(2), [z]
        fldl2t                          / log2(10), r, [z]
        fmulp                           / f := r*log2(10), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
1:
        fscale                          / 10^x, [z]
        fstp    %st(1)
        ret

.check_tail:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x9a209a84,%ecx        / Is |x| < log10(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    4(%esp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0xfbcff798,%edx        / Is |x| slightly > log10(2)?
        ja      .finite_non_special     / branch if |x| slightly > log10(2)
.shortcut:
        / Here, |x| < log10(2), so |z| = |x/log10(2)| < 1
        / whence z is in f2xm1's domain.
        fldt    4(%esp)                 / x
        fldl2t                          / log2(10), x
        fmulp                           / z := x*log2(10)
        f2xm1                           / 2^z-1
        fld1                            / 1, 2^z-1
        faddp   %st,%st(1)              / 10^x
        ret

.not_finite:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(sgnfcnd(x)) = hi_32(sgnfcnd(INF)
        jne     .NaN_or_pinf            / if not, x is NaN or unsupp.
        movl    4(%esp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0,%edx                 / lo_32(sgnfcnd(x)) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    12(%esp),%eax           / ax <-- sign&bexp((x))
        andl    $0x00008000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    4(%esp)
        ret
        .align  4
        SET_SIZE(exp10l)
```

_____unchanged_portion_omitted_

```
**********************************************************
    2756 Tue Nov  4 19:01:59 2014
new/usr/src/lib/libm/i386/src/exp2.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp2.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp2,function)
#include "libm_synonyms.h"

        ENTRY(exp2)
        movl    8(%esp),%ecx            / ecx <-- hi_32(x)
        andl    $0x7fffffff,%ecx        / ecx <-- hi_32(|x|)
        cmpl    $0x3ff00000,%ecx        / Is |x| < 1?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be only slightly < ln(2)
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) >= hi_32(INF)?
        jae     .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, 1 < |x| < INF
        fldl    4(%esp)                 / push arg
        fld     %st(0)                  / duplicate stack top
        frndint                         / [x],x
        fucom                           / x integral?
        fstsw   %ax
        sahf
        je      .x_integral             / branch if x integral
        fxch                            / x, [x]
        fsub    %st(1),%st              / x-[x], [x]
        f2xm1                           / 2**(x-[x])-1, [x]
        fld1                            / 1,2**(x-[x])-1, [x]
        faddp   %st,%st(1)              / 2**(x-[x]), [x]
        fscale                          / 2**x = 2**(arg), [x]
        fstp    %st(1)
        ret

.x_integral:
        fstp    %st(0)                  / ,x
```



```
        fld1                            / 1 = 2**0, x
        fscale                          / 2**(0 + x) = 2**x, x
        fstp    %st(1)                  / 2**x
        ret

.check_tail:
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0x00000000,%edx        / Is |x| slightly > 1?
        ja      .finite_non_special     / branch if |x| slightly > 1
.shortcut:
        / Here, |x| <= 1,
        / whence x is in f2xm1's domain.
        fldl    4(%esp)                 / push x
        f2xm1                           / 2**x - 1
        fld1                            / 1,2**x - 1
        faddp   %st,%st(1)              / 2**x
        ret

.not_finite:
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) > hi_32(INF)?
        ja      .NaN_or_pinf            / if so, x is NaN
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x80000000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldl    4(%esp)
        fwait
        ret
        .align  4
        SET_SIZE(exp2)
______unchanged_portion_omitted_
```

```
**********************************************************
    2348 Tue Nov  4 19:01:59 2014
new/usr/src/lib/libm/i386/src/exp2f.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp2f.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp2f,function)
#include "libm_synonyms.h"

        ENTRY(exp2f)
        movl    4(%esp),%ecx            / ecx <-- x
        andl    $0x7fffffff,%ecx        / ecx <-- |x|
        cmpl    $0x3f800000,%ecx        / Is |x| <= 1?
        jbe     .shortcut               / If so, take a shortcut.
        cmpl    $0x7f800000,%ecx        / |x| >= INF?
        jae     .not_finite             / if so, x is not finite
        flds    4(%esp)                 / push arg
        fld     %st(0)                  / duplicate stack top
        frndint                         / [x],x
        fucom                           / x integral?
        fstsw   %ax
        sahf
        je      .x_integral             / branch if x integral
        fxch                            / x, [x]
        fsub    %st(1),%st              / x-[x], [x]
        f2xm1                           / 2**(x-[x])-1, [x]
        fld1                            / 1,2**(x-[x])-1, [x]
        faddp   %st,%st(1)              / 2**(x-[x]), [x]
        fscale                          / 2**x = 2**(arg), [x]
        fstp    %st(1)
        ret

.x_integral:                            / here, x is integral
        fstp    %st(0)                  / ,x
        fld1                            / 1 = 2**0, x
        fscale                          / 2**(0 + x) = 2**x, x
```



```
        fstp    %st(1)                  / 2**x
        ret

.shortcut:
        / Here, |x| <= 1,
        / whence x is in f2xm1's domain.
        flds    4(%esp)                 / push x
        f2xm1                           / 2**x - 1
        fld1                            / 1,2**x - 1
        faddp   %st,%st(1)              / 2**x
        ret

.not_finite:
        ja      .NaN_or_pinf            / branch if x is NaN
        movl    4(%esp),%eax            / eax <-- x
        andl    $0x80000000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        flds    4(%esp)
        fwait
        ret
        .align  4
        SET_SIZE(exp2f)
```

*unchanged_portion_omitted_*

```
**********************************************************
    2928 Tue Nov  4 19:02:00 2014
new/usr/src/lib/libm/i386/src/exp2l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "exp2l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(exp2l,function)
#include "libm_synonyms.h"

        ENTRY(exp2l)
        movl    12(%esp),%ecx           / cx <--sign&bexp(x)
        andl    $0x00007fff,%ecx        / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x00003fff,%ecx        / Is |x| <= 1?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be slightly > 1
        cmpl    $0x00007fff,%ecx        / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, 1 < |x| < INF
        fldt    4(%esp)                 / push arg
        fld     %st(0)                  / duplicate stack top
        frndint                         / [x],x
        fucom                           / x integral?
        fnstsw  %ax
        sahf
        je      .x_integral             / branch if x integral
        fxch                            / x, [x]
        fsub    %st(1),%st              / x-[x], [x]
        f2xm1                           / 2**(x-[x])-1, [x]
        fld1                            / 1,2**(x-[x])-1, [x]
        faddp   %st,%st(1)              / 2**(x-[x]), [x]
        fscale                          / 2**x = 2**(arg), [x]
        fstp    %st(1)
        ret

.x_integral:
        fstp    %st(0)                  / ,x
```



```
        fld1                            / 1 = 2**0, x
        fscale                          / 2**(0 + x) = 2**x, x
        fstp    %st(1)                  / 2**x
        ret

.check_tail:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / Is |x| <= 1?
        ja      .finite_non_special
        movl    4(%esp),%edx            / edx <-- lo_32(sgnfcnd(x))
        cmpl    $0x00000000,%edx        / Is |x| slightly > 1?
        ja      .finite_non_special     / branch if |x| slightly > 1
.shortcut:
        / Here, |x| < 1,
        / whence x is in f2xm1's domain.
        fldt    4(%esp)                 / push x
        f2xm1                           / 2**x - 1
        fld1                            / 1,2**x - 1
        faddp   %st,%st(1)              / 2**x
        ret

.not_finite:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    12(%esp),%eax           / ax <-- sign&bexp((x))
        andl    $0x00008000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    4(%esp)
        ret
        .align  4
        SET_SIZE(exp2l)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3663 Tue Nov  4 19:02:00 2014
new/usr/src/lib/libm/i386/src/expl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expl,function)
#include "libm_synonyms.h"

        .data
        .align  4
ln2_hi: .long   0xd1d00000, 0xb17217f7, 0x00003ffe
ln2_lo: .long   0x4c67fc0d, 0x8654361c, 0x0000bfce

        ENTRY(expl)
        movl    12(%esp),%ecx           / cx <--sign&bexp(x)
        andl    $0x7fff,%ecx            / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x3ffe,%ecx            / Is |x| < 0.5?
        jb      2f                      / If so, see which shortcut to take
        je      .check_tail             / More checking if 0.5 <= |x| < 1
        cmpl    $0x00007fff,%ecx        / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        cmpl    $0x0000400e,%ecx        / |x| < 32768 = 2^15?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    4(%esp)                 / x
        fld1                            / 1, x
        jmp     1f
.finite_non_special:                    / Here, ln(2) < |x| < 2^15
        fldt    4(%esp)                 / x
        fld     %st(0)                  / x, x
        fldl2e                          / log2(e), x, x
        fmulp                           / z := x*log2(e), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
        fldt    PIC_L(ln2_hi)           / ln2_hi, [z], x, [z]
```



```
        fmulp                           / [z]*ln2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*ln2_hi, [z]
        fldt    PIC_L(ln2_lo)           / ln2_lo, x-[z]*ln2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*ln2_lo, x-[z]*ln2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*ln(2), [z]
        fldl2e                          / log2(e), r, [z]
        fmulp                           / f := r*log2(e), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
1:
        fscale                          / e^x, [z]
        fstp    %st(1)
        ret

2:                                      / Here, |x| < 0.5
        cmpl    $0x3fbe,%ecx            / Is |x| >= 2^-65?
        jae     .shortcut               / If so, take a shortcut
        fldt    4(%esp)                 / x
        fld1                            / 1, x
        faddp   %st,%st(1)              / 1+x (for inexact & directed rounding)
        ret

.check_tail:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0xb17217f7,%ecx        / Is |x| < ln(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0xd1cf79ab,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x/ln(2)| < 1,
        / whence z is in f2xm1's domain.
        fldt    4(%esp)                 / x
        fldl2e                          / log2(e), x
        fmulp                           / x*log2(e)
        f2xm1                           / 2^(x*log2(e))-1 = e^x-1
        fld1                            / 1, e^x-1
        faddp   %st,%st(1)              / e^x
        ret

.not_finite:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    12(%esp),%eax           / ax <-- sign&bexp((x))
        andl    $0x00008000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fldz                            / Here, x = -inf, so return 0
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    4(%esp)
        fadd    %st(0),%st              / quiet SNaN
        ret
        .align  4
        SET_SIZE(expl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    3708 Tue Nov  4 19:02:01 2014
new/usr/src/lib/libm/i386/src/expm1.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expm1.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expm1,function)
#include "libm_synonyms.h"

        .data
        .align  4
.mhundred:      .float  -100.0

        ENTRY(expm1)
        movl    8(%esp),%ecx            / ecx <-- hi_32(x)
        andl    $0x7fffffff,%ecx        / ecx <-- hi_32(|x|)
        cmpl    $0x3fe62e42,%ecx        / Is |x| < ln(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be only slightly < ln(2)
        cmpl    $0x7ff00000,%ecx        / hi_32(|x|) >= hi_32(INF)?
        jae     .not_finite             / if so, x is not finite
.finite_non_special:                    / Here, ln(2) < |x| < INF
        fldl    4(%esp)                 / push x

        subl    $8,%esp                 / save RP and set round-to-64-bits
        fstcw   (%esp)
        movw    (%esp),%ax
        movw    %ax,4(%esp)
        orw     $0x0300,%ax
        movw    %ax,(%esp)
        fldcw   (%esp)

        fldl2e                          / push log2e   }not for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2e  }not for xtndd_dbl
        fld     %st(0)                  / duplicate stack top
        frndint                         / [z],z
```



```
        / [z] != 0, compute exp(x) and then subtract one to get expm1(x)
        fxch                            / z,[z]
        fsub    %st(1),%st              / z-[z],[z]
        f2xm1                           / 2**(z-[z])-1,[z]
        / avoid spurious underflow when scaling to compute exp(x)
        PIC_SETUP(1)
        flds    PIC_L(.mhundred)
        PIC_WRAPUP
        fucom   %st(2)                  / if -100 !< [z], then use -100
        fstsw   %ax
        sahf
        jb      .got_int_part
        fxch    %st(2)
.got_int_part:
        fstp    %st(0)                  /   2**(z-[z])-1,max([z],-100)
        fld1                            / 1,2**(z-[z])-1,max([z],-100)
        faddp   %st,%st(1)              /   2**(z-[z])  ,max([z],-100)
        fscale                          /   exp(x)      ,max([z],-100)
        fld1                            / 1,exp(x)      ,max([z],-100)
        fxch                            / exp(x),1      ,max([z],-100)
        fsubp   %st,%st(1)              /   exp(x)-1    ,max([z],-100)
        fstp    %st(1)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.check_tail:
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0xfefa39ef,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x*log2(e)| < 1,
        / whence z is in f2xm1's domain.
        fldl    4(%esp)                 / push x
        fldl2e                          / push log2e  }not for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2e }not for xtndd_dbl
        f2xm1                           / 2**(x*log2(e))-1 = e**x - 1
        ret

.not_finite:
        / Here, flags still have settings from execution of
        /       cmpl    $0x7ff00000,%ecx        / hi_32(|x|) > hi_32(INF)?
        ja      .NaN_or_pinf            / if not, x may be +/- INF
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x80000000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
        fld1                            / Here, x = -inf, so return -1
        fchs
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldl    4(%esp)
        fwait
```

```
 127         ret
 128         .align  4
 129         SET_SIZE(expm1)
______unchanged_portion_omitted_
```

```
**********************************************************
    3982 Tue Nov  4 19:02:01 2014
new/usr/src/lib/libm/i386/src/expm1f.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expm1f.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expm1f,function)
#include "libm_synonyms.h"

        .data
        .align  4
.mhundred:      .float  -100.0

        ENTRY(expm1f)
        movl    4(%esp),%ecx            / ecx <-- x
        andl    $0x7fffffff,%ecx        / ecx <-- |x|
        cmpl    $0x3f317217,%ecx        / Is |x| < ln(2)?
        jbe     .shortcut               / If so, take a shortcut.
        cmpl    $0x7f800000,%ecx        / |x| >= INF?
        jae     .not_finite             / if so, x is not finite
        flds    4(%esp)                 / push x

        subl    $8,%esp                 / save RP and set round-to-64-bits
        fstcw   (%esp)
        movw    (%esp),%ax
        movw    %ax,4(%esp)
        orw     $0x0300,%ax
        movw    %ax,(%esp)
        fldcw   (%esp)

        fldl2e                          / push log2e   }not for xtndd_dbl
        fmulp   %st,%st(1)              / z = x*log2e  }not for xtndd_dbl
        fld     %st(0)                  / duplicate stack top
        frndint                         / [z],z
        fucom                           / This and the next 3 instructions
        fstsw   %ax                     / add 10 clocks to runtime of the
```



```
        sahf                            / main branch, but save about 265
        je      .z_integral             / upon detection of integral z.
        / [z] != 0, compute exp(x) and then subtract one to get expm1(x)
        fxch                            / z,[z]
        fsub    %st(1),%st              / z-[z],[z]
        f2xm1                           / 2**(z-[z])-1,[z]
        / avoid spurious underflow when scaling to compute exp(x)
        PIC_SETUP(1)
        flds    PIC_L(.mhundred)
        PIC_WRAPUP
        fucom   %st(2)                  / if -100 !< [z], then use -100
        fstsw   %ax
        sahf
        jb      .got_int_part
        fxch    %st(2)
.got_int_part:
        fstp    %st(0)                  /   2**(z-[z])-1,max([z],-100)
        fld1                            / 1,2**(z-[z])-1,max([z],-100)
        faddp   %st,%st(1)              /   2**(z-[z])  ,max([z],-100)
        fscale                          /   exp(x)      ,max([z],-100)
        fld1                            / 1,exp(x)      ,max([z],-100)
        fsubrp  %st,%st(1)              /   exp(x)-1    ,max([z],-100)
        fstp    %st(1)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret

.z_integral:                            / here, z is integral
        fstp    %st(0)                  / ,z
        / avoid spurious underflow when scaling to compute exp(x)
        PIC_SETUP(2)
        flds    PIC_L(.mhundred)
        PIC_WRAPUP
        fucom   %st(1)                  / if -100 !< [z], then use -100
        fstsw   %ax
        sahf
        jb      .scale_wont_ovfl
        fxch    %st(1)
.scale_wont_ovfl:
        fstp    %st(0)                  /   max([z],-100)
        fld1                            / 1,max([z],-100)
        fscale                          /   exp(x)      ,max([z],-100)
        fld1                            / 1,exp(x)      ,max([z],-100)
        fsubrp  %st,%st(1)              /   exp(x)-1    ,max([z],-100)
        fstp    %st(1)

        fstcw   (%esp)                  / restore old RP
        movw    (%esp),%dx
        andw    $0xfcff,%dx
        movw    4(%esp),%cx
        andw    $0x0300,%cx
        orw     %dx,%cx
        movw    %cx,(%esp)
        fldcw   (%esp)
        add     $8,%esp

        ret
```

```
128 .shortcut:
129         / Here, |x| < ln(2), so |z| = |x*log2(e)| < 1,
130         / whence z is in f2xm1's domain.
131         flds    4(%esp)                 / push x
132         fldl2e                          / push log2e  }not for xtndd_dbl
133         fmulp   %st,%st(1)              / z = x*log2e }not for xtndd_dbl
134         f2xm1                           / 2**(x*log2(e))-1 = e**x - 1
135         ret

137 .not_finite:
138         ja      .NaN_or_pinf            / branch if x is NaN
139         movl    4(%esp),%eax            / eax <-- x
140         andl    $0x80000000,%eax        / here, x is infinite, but +/-?
141         jz      .NaN_or_pinf            / branch if x = +INF
142         fld1                            / Here, x = -inf, so return -1
143         fchs
144         ret

146 .NaN_or_pinf:
147         / Here, x = NaN or +inf, so load x and return immediately.
148         flds    4(%esp)
149         fwait
150         ret
151         .align  4
152         SET_SIZE(expm1f)
```

*unchanged_portion_omitted_*

```
**********************************************************
    3682 Tue Nov  4 19:02:01 2014
new/usr/src/lib/libm/i386/src/expm1l.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "expm1l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(expm1l,function)
#include "libm_synonyms.h"

        .data
        .align  4
ln2_hi: .long   0xd1d00000, 0xb17217f7, 0x00003ffe
ln2_lo: .long   0x4c67fc0d, 0x8654361c, 0x0000bfce

        ENTRY(expm1l)
        movl    12(%esp),%ecx           / cx <--sign&bexp(x)
        movl    %ecx,%eax               / ax <--sign&bexp(x)
        andl    $0x00007fff,%ecx        / ecx <-- zero_xtnd(bexp(x))
        cmpl    $0x00003ffe,%ecx        / Is |x| < ln(2)?
        jb      .shortcut               / If so, take a shortcut.
        je      .check_tail             / |x| may be only slightly < ln(2)
        cmpl    $0x00007fff,%ecx        / bexp(|x|) = bexp(INF)?
        je      .not_finite             / if so, x is not finite
        andl    $0x0000ffff,%eax        / eax <-- sign&bexp(x)
        cmpl    $0x0000c006,%eax        / x <= -128?
        jae     1f                      / if so, simply return -1
        cmpl    $0x0000400d,%ecx        / |x| < 16384 = 2^14?
        jb      .finite_non_special     / if so, proceed with argument reduction
        fldt    4(%esp)                 / x >= 16384; x
        fld1                            / 1, x
        fscale                          / +Inf, x
        fstp    %st(1)                  / +Inf
        ret

.finite_non_special:                    / -128 < x < -ln(2) || ln(2) < x < 2^14
        fldt    4(%esp)                 / x
```



```
        fld     %st(0)                  / x, x
        fldl2e                          / log2(e), x, x
        fmulp                           / z := x*log2(e), x
        frndint                         / [z], x
        fst     %st(2)                  / [z], x, [z]
        PIC_SETUP(1)
        fldt    PIC_L(ln2_hi)           / ln2_hi, [z], x, [z]
        fmulp                           / [z]*ln2_hi, x, [z]
        fsubrp  %st,%st(1)              / x-[z]*ln2_hi, [z]
        fldt    PIC_L(ln2_lo)           / ln2_lo, x-[z]*ln2_hi, [z]
        PIC_WRAPUP
        fmul    %st(2),%st              / [z]*ln2_lo, x-[z]*ln2_hi, [z]
        fsubrp  %st,%st(1)              / r := x-[z]*ln(2), [z]
        fldl2e                          / log2(e), r, [z]
        fmulp                           / f := r*log2(e), [z]
        f2xm1                           / 2^f-1,[z]
        fld1                            / 1, 2^f-1, [z]
        faddp   %st,%st(1)              / 2^f, [z]
        fscale                          / e^x, [z]
        fstp    %st(1)                  / e^x
        fld1                            / 1, e^x
        fsubrp  %st,%st(1)              / e^x-1
        ret

.check_tail:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0xb17217f7,%ecx        / Is |x| < ln(2)?
        ja      .finite_non_special
        jb      .shortcut
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0xd1cf79ab,%edx        / Is |x| slightly < ln(2)?
        ja      .finite_non_special     / branch if |x| slightly > ln(2)
.shortcut:
        / Here, |x| < ln(2), so |z| = |x/ln(2)| < 1,
        / whence z is in f2xm1's domain.
        fldt    4(%esp)                 / x
        fldl2e                          / log2(e), x
        fmulp                           / z := x*log2(e)
        f2xm1                           / 2^(x*log2(e))-1 = e^x-1
        ret

.not_finite:
        movl    8(%esp),%ecx            / ecx <-- hi_32(sgnfcnd(x))
        cmpl    $0x80000000,%ecx        / hi_32(|x|) = hi_32(INF)?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    4(%esp),%edx            / edx <-- lo_32(x)
        cmpl    $0,%edx                 / lo_32(x) = 0?
        jne     .NaN_or_pinf            / if not, x is NaN
        movl    12(%esp),%eax           / ax <-- sign&bexp((x))
        andl    $0x00008000,%eax        / here, x is infinite, but +/-?
        jz      .NaN_or_pinf            / branch if x = +INF
1:
        fld1                            / Here, x = -inf, so return -1
        fchs
        ret

.NaN_or_pinf:
        / Here, x = NaN or +inf, so load x and return immediately.
        fldt    4(%esp)
        ret
        .align  4
        SET_SIZE(expm1l)
```

_____***unchanged_portion_omitted_***

****
1148 Tue Nov 4 19:02:02 2014
new/usr/src/lib/libm/i386/src/fabs.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "fabs.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(fabs,function)
33 #include "libm_synonyms.h"

34         ENTRY(fabs)
35         fldl    4(%esp)
36 #undef  fabs
37         fabs
38         ret
39         .align  4
40         SET_SIZE(fabs)
```

*_____unchanged_portion_omitted_*

****
1153 Tue Nov 4 19:02:02 2014
new/usr/src/lib/libm/i386/src/fabsf.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "fabsf.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(fabsf,function)
33 #include "libm_synonyms.h"

34         ENTRY(fabsf)
35         flds    4(%esp)
36 #undef  fabs
37         fabs
38         ret
39         .align  4
40         SET_SIZE(fabsf)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1153 Tue Nov  4 19:02:03 2014
new/usr/src/lib/libm/i386/src/fabsl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "fabsl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fabsl,function)
#include "libm_synonyms.h"

        ENTRY(fabsl)
        fldt    4(%esp)
#undef  fabs
        fabs
        ret
        .align  4
        SET_SIZE(fabsl)
______unchanged_portion_omitted_
```

****
1373 Tue Nov 4 19:02:03 2014
new/usr/src/lib/libm/i386/src/finitef.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "finitef.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(finitef,function)
#include "libm_synonyms.h"

        ENTRY(finitef)
        movl    4(%esp),%eax            / eax <-- x
        notl    %eax                    / not(bexp) = 0 iff bexp = all 1's
        andl    $0x7f800000,%eax        / ZF <-- 1      iff not(bexp) = 0
        jz      .done                   / no jump if arg. is finite
        movl    $1,%eax                 / %ax was 0; ansi needs %eax = 1
.done:
        ret
        .align  4
        SET_SIZE(finitef)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1705 Tue Nov  4 19:02:04 2014
new/usr/src/lib/libm/i386/src/finitel.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "finitel.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(finitel,function)
#include "libm_synonyms.h"

        ENTRY(finitel)
        movl    12(%esp),%eax           / %ax <-- sign&bexp(x)
        testl   $0x80000000,8(%esp)     / ZF = 1 iff hi_32(sgnfcnd(x))'s msb = 0
        jz      .chk_denormal_or_0
        notl    %eax                    / not(bexp) = 0 iff bexp = all 1's
        andl    $0x00007fff,%eax        / ZF <-- 1      iff not(bexp) = 0
        jz      .done                   / no jump if arg. is finite
        movl    $1,%eax                 / ansi needs %eax = 1
.done:
        ret

.chk_denormal_or_0:
        andl    $0x00007fff,%eax        / ZF <-- 1 iff bexp = 0 iff denormal or
        jnz     .unsupported            / jump if arg has unsupported format
        movl    $1,%eax                 / ansi needs %eax = 1
        ret

.unsupported:
        movl    $0,%eax                 / unsupported format does not represent
        ret                             / a finite number
        .align  4
        SET_SIZE(finitel)
______unchanged_portion_omitted_
```

****
1446 Tue Nov 4 19:02:05 2014
new/usr/src/lib/libm/i386/src/floor.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "floor.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(floor,function)
33 #include "libm_synonyms.h"

34         ENTRY(floor)
35         subl    $8,%esp
36         fstcw   (%esp)
37         fldl    12(%esp)
38         movw    (%esp),%cx
39         orw     $0x0c00,%cx
40         xorw    $0x0800,%cx
41         movw    %cx,4(%esp)
42         fldcw   4(%esp)                 / set RD = down
43         frndint
44         fstcw   4(%esp)                 / restore RD
45         movw    4(%esp),%dx
46         andw    $0xf3ff,%dx
47         movw    (%esp),%cx
48         andw    $0x0c00,%cx
49         orw     %dx,%cx
50         movw    %cx,(%esp)
51         fldcw   (%esp)                  / restore RD
52         addl    $8,%esp
53         ret
54         .align  4
55         SET_SIZE(floor)
```

_____***unchanged_portion_omitted_***

****
1863 Tue Nov 4 19:02:05 2014
new/usr/src/lib/libm/i386/src/floorl.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "floorl.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(ceill,function)
33 LIBM_ANSI_PRAGMA_WEAK(floorl,function)
34 #include "libm_synonyms.h"

35         ENTRY(ceill)
36         subl    $8,%esp
37         fstcw   (%esp)
38         fldt    12(%esp)
39         movw    (%esp),%cx
40         orw     $0x0c00,%cx
41         xorw    $0x0400,%cx
42         movw    %cx,4(%esp)
43         fldcw   4(%esp)                 / set RD = up
44         frndint
45         fstcw   4(%esp)                 / restore RD
46         movw    4(%esp),%dx
47         andw    $0xf3ff,%dx
48         movw    (%esp),%cx
49         andw    $0x0c00,%cx
50         orw     %dx,%cx
51         movw    %cx,(%esp)
52         fldcw   (%esp)                  / restore RD
53         addl    $8,%esp
54         ret
55         .align  4
56         SET_SIZE(ceill)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1772 Tue Nov  4 19:02:06 2014
new/usr/src/lib/libm/i386/src/fmod.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmod.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmod,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(fmod)
        movl    16(%esp),%eax           / eax <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|y|)
        orl     12(%esp),%eax           / eax <-- lo_32(y)|hi_32(|y|)
        je      .zero

        fldl    12(%esp)                / load arg y
        fldl    4(%esp)                 / load arg x
.mod_loop:
        fprem                           / partial fmod
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .mod_loop               / while incomplete, do fprem again
        fstp    %st(1)
        ret
.zero:
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)
        pushl   $27                     / case 27 in _SVID_libm_err
        pushl   20(%ebp)                / pass x
        pushl   16(%ebp)
        pushl   12(%ebp)                / pass y
        pushl   8(%ebp)
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
```



```
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(fmod)
______unchanged_portion_omitted_
```

```
**********************************************************
    1366 Tue Nov  4 19:02:06 2014
new/usr/src/lib/libm/i386/src/fmodf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmodf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmodf,function)
#include "libm_synonyms.h"

        ENTRY(fmodf)
        flds    8(%esp)                 / load arg y
        flds    4(%esp)                 / load arg x
.mod_loop:
        fprem                           / partial fmod
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .mod_loop               / while incomplete, do fprem again
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(fmodf)
```

*_____unchanged_portion_omitted_*

****
1366 Tue Nov 4 19:02:06 2014
new/usr/src/lib/libm/i386/src/fmodl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "fmodl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(fmodl,function)
#include "libm_synonyms.h"

        ENTRY(fmodl)
        fldt    16(%esp)                / load arg y
        fldt    4(%esp)                 / load arg x
.mod_loop:
        fprem                           / partial fmod
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
        jne     .mod_loop               / while incomplete, do fprem again
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(fmodl)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3208 Tue Nov  4 19:02:07 2014
new/usr/src/lib/libm/i386/src/hypot.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "hypot.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(hypot,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        .data
        .align  4
inf:
        .long   0x7f800000

        ENTRY(hypot)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|x|)
        jz      .x_maybe_0              / if x = +/-0, return |y|
        subl    $0x7ff00000,%eax        / eax <-- hi_32(|x|) - hi_32(INF)
        jz      .x_maybe_inf
.check_y:
        movl    16(%esp),%eax           / eax <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|y|)
        jz      .y_maybe_0              / if y = +/-0, return |x|
        subl    $0x7ff00000,%eax        / eax <-- hi_32(|y|) - hi_32(INF)
        jz      .y_maybe_inf
.do_hypot:
        fldl    12(%esp)                / ,y
        fmul    %st(0),%st              / ,y*y
        fldl    4(%esp)                 / x,y*y
        fmul    %st(0),%st              / x*x,y*y
        faddp   %st,%st(1)              / x*x+y*y
        fsqrt                           / sqrt(x*x+y*y)
```



```
        subl    $8,%esp
        fstpl   (%esp)                  / round to double
        fldl    (%esp)                  / sqrt(x*x+y*y) rounded to double
        PIC_SETUP(1)
        flds    PIC_L(inf)              / inf , sqrt(x*x+y*y)
        PIC_WRAPUP
        addl    $8,%esp
        fucomp
        fstsw   %ax                     / store status in %ax
        sahf                            / 80387 flags in %ah to 80386 flags
        jz      .maybe_ovflw
        ret

.maybe_ovflw:
        jnp     .ovflw
        ret

.ovflw:
        /       overflow occurred
        fstp    %st(0)                  / stack empty
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(2)
        pushl   $4
        pushl   20(%ebp)                / high y
        pushl   16(%ebp)                / low y
        pushl   12(%ebp)                / high x
        pushl   8(%ebp)                 / low x
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret

.x_maybe_0:
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        orl     %ecx,%eax               / is x = +/-0?
        jnz     .check_y                / branch if x is denormal
        /  x = +/-0, so return |y|
        fldl    12(%esp)
        fabs
        ret

.x_maybe_inf:
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        orl     %ecx,%eax               / is x = +/-INF?
        jnz     .check_y                / branch if x is NaN
        / push&pop y in case y is a SNaN
        fldl    12(%esp)
        fstp    %st(0)
        / x = +/-INF, so return |x|
        fldl    4(%esp)
        fabs
        ret

.y_maybe_0:
        movl    12(%esp),%ecx           / ecx <-- lo_32(y)
        orl     %ecx,%eax               / is y = +/-0?
        jnz     .do_hypot               / branch if y is denormal
        /  y = +/-0, so return |x|
        fldl    4(%esp)
        fabs
        ret

.y_maybe_inf:
        movl    12(%esp),%ecx           / ecx <-- lo_32(y)
```

```
 127         orl     %ecx,%eax               / is y = +/-INF?
 128         jnz     .do_hypot               / branch if y is NaN
 129         / push&pop x in case x is a SNaN
 130         fldl    4(%esp)
 131         fstp    %st(0)
 132         /  y = +/-INF, so return |y|
 133         fldl    12(%esp)
 134         fabs
 135         ret
 136         .align  4
 137         SET_SIZE(hypot)
```

unchanged_portion_omitted_

```
**********************************************************
    1886 Tue Nov  4 19:02:07 2014
new/usr/src/lib/libm/i386/src/hypotf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "hypotf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(hypotf,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

#undef fabs

        ENTRY(hypotf)
        movl    4(%esp),%eax            / eax <-- x
        andl    $0x7fffffff,%eax        / eax <-- |x|
        jz      .return_abs_y           / if x = +/-0, return |y|
        subl    $0x7f800000,%eax        / eax <-- |x| - INF
        jz      .return_abs_x           / if x = +/-INF, return |x|
        movl    8(%esp),%eax            / eax <-- y
        andl    $0x7fffffff,%eax        / eax <-- |y|
        jz      .return_abs_x           / if y = +/-0, return |x|
        subl    $0x7f800000,%eax        / eax <-- |y| - INF
.return_abs_y:
        flds    8(%esp)                 / y
        jz      .take_abs               / if y = +/-INF, return |y|
        fmul    %st(0),%st              / y*y
        flds    4(%esp)                 / x,y*y
        fmul    %st(0),%st              / x*x,y*y
        faddp   %st,%st(1)              / x*x+y*y
        fsqrt                           / sqrt(x*x+y*y)
        subl    $4,%esp
        fstps   (%esp)                  / round to single
        flds    (%esp)
        fwait
        addl    $4,%esp
        ret
```



```
.return_abs_x:
        / returns |x|
        flds    4(%esp)
.take_abs:
        fabs
        ret
        .align  4
        SET_SIZE(hypotf)
______unchanged_portion_omitted_
```

****
3188 Tue Nov 4 19:02:08 2014
new/usr/src/lib/libm/i386/src/ieee_funcl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "ieee_funcl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(isinfl,function)
LIBM_ANSI_PRAGMA_WEAK(isnormall,function)
LIBM_ANSI_PRAGMA_WEAK(issubnormall,function)
LIBM_ANSI_PRAGMA_WEAK(iszerol,function)
LIBM_ANSI_PRAGMA_WEAK(signbitl,function)
#include "libm_synonyms.h"

        ENTRY(isinfl)
        movl    12(%esp),%eax           / ax <-- sign and bexp of x
        notl    %eax
        andl    $0x00007fff,%eax
        jz      .L6
        movl    $0,%eax
.not_inf:
        ret

.L6:                                    / here, (eax) = 0.0
        movl    8(%esp),%ecx
        xorl    $0x80000000,%ecx        / handle unsupported implicitly
        orl     4(%esp), %ecx
        jnz     .not_inf
        movl    $1,%eax
        ret
        .align  4
        SET_SIZE(isinfl)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2337 Tue Nov  4 19:02:08 2014
new/usr/src/lib/libm/i386/src/ilogb.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "ilogb.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(ilogb,function)
#include "libm_synonyms.h"
#include "xpg6.h"

        .data
        .align  8
two52:  .long   0x0,0x43300000  / 2**52

        ENTRY(ilogb)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(abs(x))
        testl   $0x7ff00000,%eax        / is bexp(x) 0?
        jz      .bexp_0                 / jump if x is 0 or subnormal
                                        / biased exponent is non-zero
        cmpl    $0x7ff00000,%eax        / is bexp(x) 0x7ff?
        jae     .bexp_all_1             / jump if x is NaN or Inf
        shrl    $20,%eax                / eax <-- bexp(x)
        subl    $1023,%eax              / unbias exponent by 1023
        ret

.bexp_all_1:
        movl    $0x7fffffff,%eax        / x is NaN or inf, so return 0x7fffffff
        jmp     0f

.bexp_0:
        orl     4(%esp),%eax            / test whether x is 0
        jnz     .ilogb_subnorm
        movl    $0x80000001,%eax        / x is +/-0, so return 0x80000001
0:
        PIC_SETUP(0)
```



```
        PIC_G_LOAD(movzwl,__xpg6,ecx)
        PIC_WRAPUP
        andl    $_C99SUSv3_ilogb_0InfNaN_raises_invalid,%ecx
        cmpl    $0,%ecx
        je      1f
        fldz
        fdivp   %st,%st(0)              / raise invalid as per SUSv3
1:
        ret

.ilogb_subnorm:                         / subnormal input
        fldl    4(%esp)                 / push x
        PIC_SETUP(1)
        fmull   PIC_L(two52)            / x*2**52
        PIC_WRAPUP
        subl    $8,%esp                 / set up storage area
        fstpl   (%esp)                  / store x*2**52 in storage are
        movl    $0x7ff00000,%eax
        andl    4(%esp),%eax
        shrl    $20,%eax                / extract exponent of x*2**52
        subl    $1075,%eax              / unbias it by 1075 (= 1023 + 52)
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(ilogb)
```

*unchanged_portion_omitted_*

```
**********************************************************
    2607 Tue Nov  4 19:02:09 2014
new/usr/src/lib/libm/i386/src/ilogbf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file "ilogbf.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(ilogbf,function)
  33 #include "libm_synonyms.h"
  33 #include "xpg6.h"

  35         .data
  36         .align  8
  37 two23:  .long   0x4b000000              / 2**23

  39         ENTRY(ilogbf)
  40         movl    4(%esp),%eax            / eax <-- x
  41         testl   $0x7f800000,%eax        / is bexp(x) 0?
  42         jz      .bexp_0                 / jump if x is 0 or subnormal
  43                                         / here, biased exponent is non-zero
  44         andl    $0x7fffffff,%eax        / eax <-- abs(x)
  45         cmpl    $0x7f800000,%eax        / is bexp(x) 0xff?
  46         jae     .bexp_all_1             / jump if x is NaN or Inf
  47         shrl    $23,%eax                / eax <-- zero_xtnd(bexp(x))
  48         subl    $127,%eax               / unbias exponent by 127
  49         ret

  51 .bexp_all_1:
  52         movl    $0x7fffffff,%eax        / x is NaN or inf, so return 0x7fffffff
  53         jmp     0f

  55 .bexp_0:
  56         andl    $0x7fffffff,%eax        / eax <-- abs(x), and
  57                                         / ZF = 1 iff x = 0.0
  58         jnz     .ilogb_subnorm
  59         movl    $0x80000001,%eax        / x is +/-0, so return 0x80000001
  60 0:
```



```
  61         PIC_SETUP(0)
  62         PIC_G_LOAD(movzwl,__xpg6,ecx)
  63         PIC_WRAPUP
  64         andl    $_C99SUSv3_ilogb_0InfNaN_raises_invalid,%ecx
  65         cmpl    $0,%ecx
  66         je      1f
  67         fldz
  68         fdivp   %st,%st(0)              / raise invalid as per SUSv3
  69 1:
  70         ret

  72 .ilogb_subnorm:                         / subnormal input
  73         flds    4(%esp)                 / push x
  74         PIC_SETUP(1)
  75         fmuls   PIC_L(two23)            / x*2**23; rebias x by 127+23,
  76                                         /          instead of 127
  77         PIC_WRAPUP
  78         subl    $4,%esp                 / set up storage area
  79         fstps   (%esp)                  / store x*2**23 in storage area
  80         fwait                           / (shouldn't raise exception, but
  81                                         / just in case)
  82         movl    $0x7f800000,%eax        / eax <-- single_bexp_mask
  83         andl    (%esp),%eax             / eax[23..30]  <-- bexp(x*2**23),
  84                                         / rest_of(eax) <-- 0
  85         shrl    $23,%eax                / eax <-- zero_xtnd(bexp(x*2**23))
  86         subl    $150,%eax               / unbias rebiased x by 150 (= 127 + 23)
  87         addl    $4,%esp                 / restore stack for caller
  88         ret
  89         .align  4
  90         SET_SIZE(ilogbf)
______unchanged_portion_omitted_
```

```
**********************************************************
    2421 Tue Nov  4 19:02:09 2014
new/usr/src/lib/libm/i386/src/ilogbl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file "ilogbl.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(ilogbl,function)
  33 #include "libm_synonyms.h"
  33 #include "xpg6.h"

  35         .data
  36         .align  8
  37 two63:  .long   0x0,0x43d00000          / 2**63

  39         ENTRY(ilogbl)
  40         movl    12(%esp),%eax           / eax <-- sign and bexp of x
  41         andl    $0x00007fff,%eax        / eax <-- bexp(x)
  42         jz      .bexp_0                 / jump iff x is 0 or subnormal
  43                                         / here, biased exponent is non-zero
  44         testl   $0x80000000,8(%esp)     / test msb of hi_32(sgnfcnd(x))
  45         jz      .ilogbl_not_finite      / jump if unsupported format
  46         cmpl    $0x00007fff,%eax
  47         je      .ilogbl_not_finite
  48         subl    $16383,%eax             / unbias exponent by 16383 = 0x3fff
  49         ret

  51 .ilogbl_not_finite:
  52         movl    $0x7fffffff,%eax        / x is NaN/inf/unsup
  53         jmp     0f

  55 .bexp_0:
  56         movl    8(%esp),%eax            / eax <-- hi_32(sgnfcnd(x))
  57         orl     4(%esp),%eax            / test whether x is 0
  58         jnz     .ilogbl_subnorm         / jump iff x is subnormal
  59         movl    $0x80000001,%eax        / x is +/-0, so return 0x80000001
  60 0:
```



```
  61         PIC_SETUP(0)
  62         PIC_G_LOAD(movzwl,__xpg6,ecx)
  63         PIC_WRAPUP
  64         andl    $_C99SUSv3_ilogb_0InfNaN_raises_invalid,%ecx
  65         cmpl    $0,%ecx
  66         je      1f
  67         fldz
  68         fdivp   %st,%st(0)              / raise invalid as per SUSv3
  69 1:
  70         ret


  73 .ilogbl_subnorm:                        / subnormal or pseudo-denormal input
  74         fldt    4(%esp)                 / push x, setting D-flag
  75         PIC_SETUP(1)
  76         fmull   PIC_L(two63)            / x*2**63
  77         PIC_WRAPUP
  78         subl    $12,%esp
  79         fstpt   (%esp)
  80         movl    $0x00007fff,%eax
  81         andl    8(%esp),%eax            / eax <-- sign and bexp of x*2**63
  82         subl    $16445,%eax             / unbias it by (16,383 + 63)
  83         addl    $12,%esp
  84         ret
  85         .align  4
  86         SET_SIZE(ilogbl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1852 Tue Nov  4 19:02:10 2014
new/usr/src/lib/libm/i386/src/isnan.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "isnan.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(isnan,function)
        .weak _isnan
        .type _isnan,@function
_isnan  = __isnan
        .weak _isnand
        .type _isnand,@function
_isnand = __isnan
        .weak isnand
        .type isnand,@function
isnand  = __isnan
#include "libm_synonyms.h"

        ENTRY(isnan)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(abs(x))
        subl    $0x7ff00000,%eax        / weed out finite values
        jae     .nan_or_inf             / no jump if arg. is finite
        movl    $0,%eax                 / ansi needs (eax) = 0
        ret
.nan_or_inf:
        ja      .got_nan                / no jump if arg. may be infinite;
                                        / let nan waste time
                                        / (eax) = 0 here
        testl   $0xffffffff,4(%esp)     / ZF <-- 1 iff lo_frac. = 0
                                        /          iff arg. is infinite
        jnz     .got_nan                / no jump if arg. is infinite;
        ret
.got_nan:
        movl    $1,%eax                 / %eax was 0, must be made 1 to
                                        / indicate TRUE
```



```
        ret
        .align  4
        SET_SIZE(isnan)
______unchanged_portion_omitted_
```

```
**********************************************************
    1575 Tue Nov  4 19:02:10 2014
new/usr/src/lib/libm/i386/src/isnanf.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "isnanf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(isnanf,function)
        .weak _isnanf
        .type _isnanf,@function
_isnanf = __isnanf
#include "libm_synonyms.h"

        ENTRY(isnanf)
        movl    4(%esp),%eax            / eax <-- x
        andl    $0x7fffffff,%eax        / eax <-- abs(x)
        subl    $0x7f800000,%eax        / ZF <-- 1      iff x is infinite
        jae     .nan_or_inf             / no jump iff arg. is finite
        movl    $0,%eax
        ret
.nan_or_inf:
        jnz     .got_nan                / no jump if arg. infinite;
                                        / let nan waste time
        ret                             / %eax = 0 here
.got_nan:
        movl    $1,%eax                 / %eax was 0, must be made 1 to
                                        / indicate TRUE
        ret
        .align  4
        SET_SIZE(isnanf)
______unchanged_portion_omitted_
```

****
1705 Tue Nov 4 19:02:11 2014
new/usr/src/lib/libm/i386/src/isnanl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "isnanl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(isnanl,function)
#include "libm_synonyms.h"

        ENTRY(isnanl)
        movl    12(%esp),%eax           / ax <-- sign bit and exp
        andl    $0x00007fff,%eax
        jz      .not_nan                / jump if exp is all 0
        xorl    $0x00007fff,%eax
        jz      .nan_or_inf             / jump if exp is all 1
        testl   $0x80000000,8(%esp)
        jz      .got_nan                / jump if leading bit is 0
        movl    $0,%eax
.not_nan:
        ret
.nan_or_inf:                            / note that %eax = 0 from before
        cmpl    $0x80000000,8(%esp)     / what is first half of significand?
        jnz     .got_nan                / jump if not equal to 0x80000000
        testl   $0xffffffff,4(%esp)     / is second half of significand 0?
        jnz     .got_nan                / jump if not equal to 0
        ret
.got_nan:
        movl    $1,%eax
        ret
        .align  4
        SET_SIZE(isnanl)
```

_____***unchanged_portion_omitted_***

****
1262 Tue Nov 4 19:02:11 2014
new/usr/src/lib/libm/i386/src/llrint.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "llrint.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(llrint,function)
#include "libm_synonyms.h"

        ENTRY(llrint)
        movl    %esp,%ecx
        subl    $8,%esp
        fldl    4(%ecx)                 / load x
        fistpll -8(%ecx)                / [x]
        fwait
        movl    -8(%ecx),%eax
        movl    -4(%ecx),%edx
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(llrint)
```

_____unchanged_portion_omitted_

****
1266 Tue Nov 4 19:02:11 2014
new/usr/src/lib/libm/i386/src/llrintf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "llrintf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(llrintf,function)
#include "libm_synonyms.h"

        ENTRY(llrintf)
        movl    %esp,%ecx
        subl    $8,%esp
        flds    4(%ecx)                 / load x
        fistpll -8(%ecx)                / [x]
        fwait
        movl    -8(%ecx),%eax
        movl    -4(%ecx),%edx
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(llrintf)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1266 Tue Nov  4 19:02:12 2014
new/usr/src/lib/libm/i386/src/llrintl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "llrintl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(llrintl,function)
#include "libm_synonyms.h"

        ENTRY(llrintl)
        movl    %esp,%ecx
        subl    $8,%esp
        fldt    4(%ecx)                 / load x
        fistpll -8(%ecx)                / [x]
        fwait
        movl    -8(%ecx),%eax
        movl    -4(%ecx),%edx
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(llrintl)
______unchanged_portion_omitted_
```

```
**********************************************************
    2282 Tue Nov  4 19:02:12 2014
new/usr/src/lib/libm/i386/src/log.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "log.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(log)
        fldln2                          / loge(2)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        testl   $0x80000000,%eax
        jnz     .maybe_0_or_less
        testl   $0x7fffffff,%eax
        jz      .maybe_0
        fldl    4(%esp)                 / arg, loge(2)
        fyl2x                           / loge(2)*log2(arg); ln(arg)
        ret

.maybe_0:
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx
        je      .zero                   / no branch if x is +denormal
.neg_nan_reentry:
        fldl    4(%esp)                 / arg, loge(2)
        fyl2x                           / loge(2)*log2(arg); ln(arg)
        ret

.zero_or_less:
        / x =< 0
        testl   $0x7fffffff,%eax
        jnz     .less_than_0
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx
```



```
        jne     .less_than_0            / branch if x is -denormal
.zero:
        / x = +/-0
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)
        pushl   $16
        jmp     .merge

.maybe_0_or_less:
        cmpl    $0xfff00000,%eax        / -INF below hi_32(x)?
        ja      .neg_nan_reentry
        jb      .zero_or_less
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx                 / is x NaN or -INF?
        jne     .neg_nan_reentry        / branch if x is NaN with signbit = 1
        / x = -INF
.less_than_0:
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(2)
        pushl   $17
.merge:
        fstp    %st(0)                  / stack empty
        pushl   12(%ebp)
        pushl   8(%ebp)
        pushl   12(%ebp)
        pushl   8(%ebp)
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(log)
______unchanged_portion_omitted_
```

```
**********************************************************
    2300 Tue Nov  4 19:02:13 2014
new/usr/src/lib/libm/i386/src/log10.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "log10.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log10,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(log10)
        fldlg2                          / log10(2)
        movl    8(%esp),%eax            / eax <-- hi_32(x)
        testl   $0x80000000,%eax
        jnz     .maybe_0_or_less
        testl   $0x7fffffff,%eax
        jz      .maybe_0
        fldl    4(%esp)                 / arg, log10(2)
        fyl2x                           / log10(2)*log2(arg); log10(arg)
        ret

.maybe_0:
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx
        je      .zero                   / no branch if x is +denormal
.neg_nan_reentry:
        fldl    4(%esp)                 / arg, log10(2)
        fyl2x                           / log10(2)*log2(arg); log10(arg)
        ret

.zero_or_less:
        / x =< 0
        testl   $0x7fffffff,%eax
        jnz     .less_than_0
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx
```



```
        jne     .less_than_0            / branch if x is -denormal
.zero:
        / x = +/-0
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)
        pushl   $18
        jmp     .merge

.maybe_0_or_less:
        cmpl    $0xfff00000,%eax        / -INF below hi_32(x)?
        ja      .neg_nan_reentry
        jb      .zero_or_less
        movl    4(%esp),%ecx            / ecx <-- lo_32(x)
        cmpl    $0,%ecx                 / is x NaN or -INF?
        jne     .neg_nan_reentry        / branch if x is NaN with signbit = 1
        / x = -INF
.less_than_0:
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(2)
        pushl   $19
.merge:
        fstp    %st(0)                  / stack empty
        pushl   12(%ebp)
        pushl   8(%ebp)
        pushl   12(%ebp)
        pushl   8(%ebp)
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(log10)
```

*_____unchanged_portion_omitted_*

****
1256 Tue Nov 4 19:02:13 2014
new/usr/src/lib/libm/i386/src/log10f.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "log10f.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log10f,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(log10f)
        fldlg2
        flds    4(%esp)                 / st = arg, st(1) = log10(2)
        fyl2x                           / st = log10(arg) = log10(2)*log2(arg)
        ret
        .align  4
        SET_SIZE(log10f)
```

_____unchanged_portion_omitted_

****
1231 Tue Nov 4 19:02:14 2014
new/usr/src/lib/libm/i386/src/log10l.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file "log10l.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(log10l,function)
33 #include "libm_synonyms.h"

34         ENTRY(log10l)
35         fldlg2
36         fldt    4(%esp)                 / st = arg, st(1) = log10(2)
37         fyl2x                           / st = log10(arg) = log10(2)*log2(arg)
38         ret
39         .align  4
40         SET_SIZE(log10l)
```

*unchanged_portion_omitted_*

```
**********************************************************
    1186 Tue Nov  4 19:02:14 2014
new/usr/src/lib/libm/i386/src/log2.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "log2.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log2,function)
#include "libm_synonyms.h"

        ENTRY(log2)
        fld1                    / push 1.0
        fldl    4(%esp)         / push x
        fyl2x                   / st = 1.0*log2(arg)
        ret
        .align  4
        SET_SIZE(log2)
______unchanged_portion_omitted_
```

```
**********************************************************
    1190 Tue Nov  4 19:02:15 2014
new/usr/src/lib/libm/i386/src/log2f.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "log2f.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log2f,function)
#include "libm_synonyms.h"

        ENTRY(log2f)
        fld1                    / push 1.0
        flds    4(%esp)         / push x
        fyl2x                   / st = 1.0*log2(arg)
        ret
        .align  4
        SET_SIZE(log2f)
______unchanged_portion_omitted_
```

```
**********************************************************
    1190 Tue Nov  4 19:02:15 2014
new/usr/src/lib/libm/i386/src/log2l.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "log2l.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(log2l,function)
#include "libm_synonyms.h"

        ENTRY(log2l)
        fld1                    / push 1.0
        fldt    4(%esp)         / push x
        fyl2x                   / st = 1.0*log2(arg)
        ret
        .align  4
        SET_SIZE(log2l)
______unchanged_portion_omitted_
```

```
**********************************************************
    1218 Tue Nov  4 19:02:16 2014
new/usr/src/lib/libm/i386/src/logl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "logl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(logl,function)
#include "libm_synonyms.h"

        ENTRY(logl)
        fldln2
        fldt    4(%esp)                 / st = arg, st(1) = loge(2)
        fyl2x                           / st = ln(arg) = loge(2)*log2(arg)
        ret
        .align  4
        SET_SIZE(logl)
```

_____***unchanged_portion_omitted_***

****
1237 Tue Nov 4 19:02:16 2014
new/usr/src/lib/libm/i386/src/lrint.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "lrint.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(lrint,function)
#include "libm_synonyms.h"

        ENTRY(lrint)
        movl    %esp,%ecx
        subl    $8,%esp
        fldl    4(%ecx)                 / load x
        fistpl  -8(%ecx)                / [x]
        fwait
        movl    -8(%ecx),%eax
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(lrint)
```

_____unchanged_portion_omitted_

****
1241 Tue Nov 4 19:02:16 2014
new/usr/src/lib/libm/i386/src/lrintf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "lrintf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(lrintf,function)
#include "libm_synonyms.h"

        ENTRY(lrintf)
        movl    %esp,%ecx
        subl    $8,%esp
        flds    4(%ecx)                 / load x
        fistpl  -8(%ecx)                / [x]
        fwait
        movl    -8(%ecx),%eax
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(lrintf)
```

*_____unchanged_portion_omitted_*

****
1241 Tue Nov 4 19:02:17 2014
new/usr/src/lib/libm/i386/src/lrintl.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "lrintl.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(lrintl,function)
33 #include "libm_synonyms.h"

34         ENTRY(lrintl)
35         movl    %esp,%ecx
36         subl    $8,%esp
37         fldt    4(%ecx)                 / load x
38         fistpl  -8(%ecx)                / [x]
39         fwait
40         movl    -8(%ecx),%eax
41         addl    $8,%esp
42         ret
43         .align  4
44         SET_SIZE(lrintl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    2170 Tue Nov  4 19:02:17 2014
new/usr/src/lib/libm/i386/src/lround.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "lround.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(lround,function)
#include "libm_synonyms.h"
#undef fabs

        .section .rodata
        .align  4
.Lhalf: .float  0.5

        ENTRY(lround)
        movl    %esp,%ecx
        subl    $8,%esp
        fstcw   -8(%ecx)
        fldl    4(%ecx)
        movw    -8(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    %dx,-4(%ecx)
        fldcw   -4(%ecx)                / set RD = to_nearest
        fld     %st(0)
        frndint                         / [x],x
        fstcw   -4(%ecx)
        movw    -4(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    -8(%ecx),%ax
        andw    $0x0c00,%ax
        orw     %dx,%ax
        movw    %ax,-8(%ecx)
        fldcw   -8(%ecx)                / restore RD
        fucom                           / check if x is already an integer
        fstsw   %ax
        sahf
```



```
        jp      0f
        je      0f
        fxch                            / x,[x]
        fsub    %st(1),%st              / x-[x],[x]
        fabs                            / |x-[x]|,[x]
        PIC_SETUP(1)
        fcoms   PIC_L(.Lhalf)
        PIC_WRAPUP
        fnstsw  %ax
        sahf
        jae     2f                      / if |x-[x]| = 0.5 goto halfway,
                                        / most cases will not take branch.
0:
        fstp    %st(0)
1:
        fistpl  -8(%ecx)
        fwait
        movl    -8(%ecx),%eax
        addl    $8,%esp
        ret
2:
    / x = n+0.5, recompute lround(x) as x+sign(x)*0.5
        fldl    4(%ecx)                 / x, 0.5, [x]
        movl    8(%ecx),%eax            / high part of x
        andl    $0x80000000,%eax
        jnz     3f
        faddp
        fstp    %st(1)
        jmp     1b
3:
        / here, x is negative, so return x-0.5
        fsubp   %st,%st(1)              / x-0.5,[x]
        fstp    %st(1)
        jmp     1b
        .align  4
        SET_SIZE(lround)
______unchanged_portion_omitted_
```

```
**********************************************************
    2194 Tue Nov  4 19:02:18 2014
new/usr/src/lib/libm/i386/src/lroundl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "lroundl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(lroundl,function)
#include "libm_synonyms.h"
#undef fabs

        .section .rodata
        .align  4
.Lhalf: .float  0.5

        ENTRY(lroundl)
        movl    %esp,%ecx
        subl    $8,%esp
        fstcw   -8(%ecx)
        fldt    4(%ecx)
        movw    -8(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    %dx,-4(%ecx)
        fldcw   -4(%ecx)                / set RD = to_nearest
        fld     %st(0)
        frndint                         / [x],x
        fstcw   -4(%ecx)
        movw    -4(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    -8(%ecx),%ax
        andw    $0x0c00,%ax
        orw     %dx,%ax
        movw    %ax,-8(%ecx)
        fldcw   -8(%ecx)                / restore RD
        fucom                           / check if x is already an integer
        fstsw   %ax
        sahf
```



```
        jp      0f
        je      0f
        fxch                            / x,[x]
        fsub    %st(1),%st              / x-[x],[x]
        fabs                            / |x-[x]|,[x]
        PIC_SETUP(1)
        fcoms   PIC_L(.Lhalf)
        PIC_WRAPUP
        fnstsw  %ax
        sahf
        jae     2f                      / if |x-[x]| = 0.5 goto halfway,
                                        / most cases will not take branch.
0:
        fstp    %st(0)
1:
        fistpl  -8(%ecx)
        fwait
        movl    -8(%ecx),%eax
        addl    $8,%esp
        ret
2:
    / x = n+0.5, recompute lroundl(x) as x+sign(x)*0.5
        fldt    4(%ecx)                 / x, 0.5, [x]
        movw    12(%ecx),%ax            / sign+exp part of x
        andw    $0x8000,%ax             / look at sign bit
        jnz     3f
        faddp
        fstp    %st(1)
        jmp     1b
3:
        / here, x is negative, so return x-0.5
        fsubp   %st,%st(1)              / x-0.5,[x]
        fstp    %st(1)
        jmp     1b
        .align  4
        SET_SIZE(lroundl)
______unchanged_portion_omitted_
```

```
**********************************************************
    2706 Tue Nov  4 19:02:18 2014
new/usr/src/lib/libm/i386/src/nextafter.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "nextafter.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(nextafter,function)
        .weak _nextafter
        .type _nextafter,@function
_nextafter      = __nextafter
#include "libm_synonyms.h"
#include "libm_protos.h"

        .data
        .align  8
Fmin:   .long   0x1,0x0
ftmp:   .long   0,0             /// WILL WRITE INTO


        ENTRY(nextafter)
        pushl   %ebp
        movl    %esp,%ebp
        fldl    16(%ebp)        / y
        subl    $8,%esp
        fldl    8(%ebp)         / load x
        fucom                   / x : y
        fstsw   %ax
        sahf
        jp      .NaN
        je      .equal
        fstp    %st(1)          / x
        ja      .bigger
        / x < y
        ftst
        movl    $1,%ecx         /// Fmin
        movl    %ecx,-8(%ebp)
```



```
        movl    $0,%ecx         /// Fmin+4
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
        je      .final
        ja      .addulp
        jb      .subulp
.bigger:
        / x > y
        ftst
        movl    $1,%ecx         /// Fmin
        movl    %ecx,-8(%ebp)
        movl    $0,%ecx         /// Fmin+4
        xorl    $0x80000000,%ecx
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
        je      .final
        jb      .addulp
.subulp:
        movl    8(%ebp),%eax    / low x
        movl    12(%ebp),%ecx   / high x
        subl    $1,%eax         / low x - ulp
        movl    %eax,-8(%ebp)
        sbbl    $0x0,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .final
.addulp:
        movl    8(%ebp),%eax    / low x
        movl    12(%ebp),%ecx   / high x
        addl    $1,%eax         / low x + ulp
        movl    %eax,-8(%ebp)
        adcl    $0x0,%ecx
        movl    %ecx,-4(%ebp)

.final:
        fstp    %st(0)
        fldl    -8(%ebp)
        andl    $0x7ff00000,%ecx
        jz      .underflow
        cmpl    $0x7ff00000,%ecx
        je      .overflow
        jmp     .return
.overflow:
        PIC_SETUP(1)
        pushl   $46
        fstp    %st(0)          / stack empty
        pushl   -4(%ebp)
        pushl   -8(%ebp)
        pushl   -4(%ebp)
        pushl   -8(%ebp)
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        jmp     .return
.underflow:
        PIC_SETUP(2)
        fldl    PIC_L(Fmin)
        fmul    %st(0),%st
        fstpl   PIC_L(ftmp)     / create underflow signal
        PIC_WRAPUP
        jmp     .return
.equal:
        fstp    %st(0)          / C99 says to return y when x == y
        jmp     .return
.NaN:
```

```
 127         faddp   %st,%st(1)      / x+y,x
 128 .return:
 129         fwait
 130         leave
 131         ret
 132         .align  4
 133         SET_SIZE(nextafter)
```

unchanged_portion_omitted_

```
**********************************************************
    2460 Tue Nov  4 19:02:19 2014
new/usr/src/lib/libm/i386/src/nextafterf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "nextafterf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(nextafterf,function)
#include "libm_synonyms.h"

        .data
        .align  4
Fmaxf:  .long   0x7f7fffff
Fminf:  .long   0x1
ftmpf:  .long   0


        ENTRY(nextafterf)
        pushl   %ebp
        movl    %esp,%ebp
        movl    $0,%eax         /// upper half of %eax must be initialized
        flds    12(%ebp)        / y
        subl    $4,%esp
        flds    8(%ebp)         / x, y
        fucom                   / x : y
        fstsw   %ax
        sahf
        jp      .NaN
        je      .equal
        fstp    %st(1)          / x
        ja      .bigger
        / x < y
        ftst                    / x : 0
        movl    $0x1,-4(%ebp)           / -4(%ebp) contains Fminf
        fnstsw  %ax
        sahf
        je      .final
```



```
        ja      .addulp
        jb      .subulp
.bigger:
        / x > y
        ftst                    / x : 0
        movl    $0x80000001,-4(%ebp)    / -4(%ebp) contains -Fminf
        fnstsw  %ax
        sahf
        je      .final
        jb      .addulp
.subulp:
        movl    8(%ebp),%eax    / x
        subl    $1,%eax         / x - ulp
        movl    %eax,-4(%ebp)
        jmp     .final
.addulp:
        movl    8(%ebp),%eax    / x
        addl    $1,%eax         / x + ulp
        movl    %eax,-4(%ebp)

.final:
        fstp    %st(0)          / empty
        flds    -4(%ebp)        / z
        andl    $0x7f800000,%eax
        jz      .underflow
        cmpl    $0x7f800000,%eax
        je      .overflow
        jmp     .return
.overflow:
        PIC_SETUP(1)
        flds    PIC_L(Fmaxf)    / Fmaxf, z
        fmul    %st(0),%st      / overflow-to-Inf, z
        fstps   PIC_L(ftmpf)    / z & create overflow signal
        PIC_WRAPUP
        jmp     .return
.underflow:
        PIC_SETUP(2)
        flds    PIC_L(Fminf)    / Fminf, z
        fmul    %st(0),%st      / underflow-to-0, z
        fstps   PIC_L(ftmpf)    / z & create underflow signal
        PIC_WRAPUP
        jmp     .return
.equal:
        fstp    %st(0)          / C99 says to return y when x == y
        jmp     .return
.NaN:
        faddp   %st,%st(1)      / x+y
.return:
        fwait
        leave
        ret
        .align  4
        SET_SIZE(nextafterf)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    4201 Tue Nov  4 19:02:19 2014
new/usr/src/lib/libm/i386/src/nextafterl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "nextafterl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(nextafterl,function)
#include "libm_synonyms.h"

        .section .rodata
        .align  4
.LFmaxl:        .long   0xffffffff,0xffffffff,0x00007ffe
.LFminl:        .long   0x1,0x0,0x0


        ENTRY(nextafterl)
        pushl   %ebp
        movl    %esp,%ebp
        fldt    20(%ebp)        / y
        subl    $12,%esp
        fldt    8(%ebp)         / load x
        fucom                   / x : y
        fstsw   %ax
        sahf
        jp      .LNaN
        je      .Lequal
        fstp    %st(1)          / x
        ja      .Lbigger
        / x < y
        ftst
        movl    $1,-12(%ebp)    /// -12(%ebp) contains Fminl
        movl    $0,-8(%ebp)
        movl    $0,%ecx                 /// final needs this
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
```



```
        je      .Lfinal
        ja      .Laddulp
        jb      .Lsubulp
.Lbigger:
        / x > y
        ftst
        movl    $1,-12(%ebp)    /// -12(%ebp) contains -Fminl
        movl    $0,-8(%ebp)
        movl    $0x00008000,%ecx        /// final needs this
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
        je      .Lfinal
        jb      .Laddulp
.Lsubulp:
        movl    12(%ebp),%edx   / high word of significand of x
        movl    16(%ebp),%ecx   / x's exponent
        andl    $0x0000ffff,%ecx
        movl    %edx,%eax
        not     %eax
        andl    $0x80000000,%eax        / look at explicit leading bit
        orl     %ecx,%eax
        andl    $0x80007fff,%eax
        jnz     .Lnot_pseudonormal      / zero value implies pseudonormal
        addl    $1,%ecx         / if pseudonormal, turn into equivalent normal
.Lnot_pseudonormal:
        movl    8(%ebp),%eax    / low x
        subl    $1,%eax         / low x - ulp
        movl    %eax,-12(%ebp)
        cmpl    $0xffffffff,%eax        / this means low x was 0
        jz      .Lborrow
        movl    %edx,-8(%ebp)
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lborrow:
        cmpl    $0x80000000,%edx        / look at high x
        je      .Lsecond_borrow
        subl    $1,%edx
        movl    %edx,-8(%ebp)
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lsecond_borrow:
        movl    %ecx,%eax
        andl    $0x7fff,%eax    / look at exp x without sign bit
        cmpl    $1,%eax
        jbe     .Lsubnormal_result      / exp > 1 ==> result will be normal
        movl    $0xffffffff,-8(%ebp)
        subl    $1,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lsubnormal_result:
        movl    $0x7fffffff,-8(%ebp)
        movl    %ecx,%eax
        andl    $0x8000,%eax    / look at sign bit
        jz      .Lpositive
        movl    $0x8000,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lpositive:
        movl    $0,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Laddulp:
        movl    12(%ebp),%edx   / high x
        movl    16(%ebp),%ecx   / x's exponent
        andl    $0x0000ffff,%ecx
```

```
 127         movl    %edx,%eax
 128         not     %eax
 129         andl    $0x80000000,%eax        / look at explicit leading bit
 130         orl     %ecx,%eax
 131         andl    $0x80007fff,%eax
 132         jnz     .Lnot_pseudonormal_2    / zero value implies pseudonormal
 133         addl    $1,%ecx
 134 .Lnot_pseudonormal_2:
 135         movl    8(%ebp),%eax    / low x
 136         addl    $1,%eax         / low x + ulp
 137         movl    %eax,-12(%ebp)
 138         jz      .Lcarry         / jump if the content of %eax is 0
 139         movl    %edx,-8(%ebp)
 140         movl    %ecx,-4(%ebp)
 141         jmp .Lfinal
 142 .Lcarry:
 143         movl    %edx,%eax
 144         andl    $0x7fffffff,%eax
 145         cmpl    $0x7fffffff,%eax        / look at high x
 146         je      .Lsecond_carry
 147         addl    $1,%edx
 148         movl    %edx,-8(%ebp)
 149         movl    %ecx,-4(%ebp)
 150         jmp     .Lfinal
 151 .Lsecond_carry:
 152         movl    $0x80000000,-8(%ebp)
 153         addl    $1,%ecx
 154         movl    %ecx,-4(%ebp)
 155 .Lfinal:
 156         fstp    %st(0)
 157         fldt    -12(%ebp)
 158         andl    $0x00007fff,%ecx
 159         jz      .Lunderflow
 160         cmpw    $0x7fff,%cx
 161         je      .Loverflow
 162         jmp     .Lreturn
 163 .Loverflow:
 164         PIC_SETUP(1)
 165         fldt    PIC_L(.LFmaxl)
 166         PIC_WRAPUP
 167         fmulp   %st,%st(0)      / create overflow signal
 168         jmp     .Lreturn
 169 .Lunderflow:
 170         PIC_SETUP(2)
 171         fldt    PIC_L(.LFminl)
 172         PIC_WRAPUP
 173         fmulp   %st,%st(0)      / create underflow signal
 174         jmp     .Lreturn
 175 .Lequal:
 176         fstp    %st(0)          / C99 says to return y when x == y
 177         jmp     .Lreturn
 178 .LNaN:
 179         faddp   %st,%st(1)      / x+y,x
 180 .Lreturn:
 181         fwait
 182         leave
 183         ret
 184         .align  4
 185         SET_SIZE(nextafterl)
```

*unchanged_portion_omitted_*

```
**********************************************************
    4205 Tue Nov  4 19:02:19 2014
new/usr/src/lib/libm/i386/src/nexttowardl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "nexttowardl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(nexttowardl,function)
#include "libm_synonyms.h"

        .section .rodata
        .align  4
.LFmaxl:        .long   0xffffffff,0xffffffff,0x00007ffe
.LFminl:        .long   0x1,0x0,0x0


        ENTRY(nexttowardl)
        pushl   %ebp
        movl    %esp,%ebp
        fldt    20(%ebp)        / y
        subl    $12,%esp
        fldt    8(%ebp)         / load x
        fucom                   / x : y
        fstsw   %ax
        sahf
        jp      .LNaN
        je      .Lequal
        fstp    %st(1)          / x
        ja      .Lbigger
        / x < y
        ftst
        movl    $1,-12(%ebp)    /// -12(%ebp) contains Fminl
        movl    $0,-8(%ebp)
        movl    $0,%ecx                 /// final needs this
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
        je      .Lfinal
        ja      .Laddulp
        jb      .Lsubulp
.Lbigger:
        / x > y
        ftst
        movl    $1,-12(%ebp)    /// -12(%ebp) contains -Fminl
        movl    $0,-8(%ebp)
        movl    $0x00008000,%ecx        /// final needs this
        movl    %ecx,-4(%ebp)
        fnstsw  %ax
        sahf
        je      .Lfinal
        jb      .Laddulp
.Lsubulp:
        movl    12(%ebp),%edx   / high word of significand of x
        movl    16(%ebp),%ecx   / x's exponent
        andl    $0x0000ffff,%ecx
        movl    %edx,%eax
        not     %eax
        andl    $0x80000000,%eax        / look at explicit leading bit
        orl     %ecx,%eax
        andl    $0x80007fff,%eax
        jnz     .Lnot_pseudonormal      / zero value implies pseudonormal
        addl    $1,%ecx         / if pseudonormal, turn into equivalent normal
.Lnot_pseudonormal:
        movl    8(%ebp),%eax    / low x
        subl    $1,%eax         / low x - ulp
        movl    %eax,-12(%ebp)
        cmpl    $0xffffffff,%eax        / this means low x was 0
        jz      .Lborrow
        movl    %edx,-8(%ebp)
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lborrow:
        cmpl    $0x80000000,%edx        / look at high x
        je      .Lsecond_borrow
        subl    $1,%edx
        movl    %edx,-8(%ebp)
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lsecond_borrow:
        movl    %ecx,%eax
        andl    $0x7fff,%eax    / look at exp x without sign bit
        cmpl    $1,%eax
        jbe     .Lsubnormal_result      / exp > 1 ==> result will be normal
        movl    $0xffffffff,-8(%ebp)
        subl    $1,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lsubnormal_result:
        movl    $0x7fffffff,-8(%ebp)
        movl    %ecx,%eax
        andl    $0x8000,%eax    / look at sign bit
        jz      .Lpositive
        movl    $0x8000,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Lpositive:
        movl    $0,%ecx
        movl    %ecx,-4(%ebp)
        jmp     .Lfinal
.Laddulp:
        movl    12(%ebp),%edx   / high x
        movl    16(%ebp),%ecx   / x's exponent
        andl    $0x0000ffff,%ecx
```

```
 127         movl    %edx,%eax
 128         not     %eax
 129         andl    $0x80000000,%eax        / look at explicit leading bit
 130         orl     %ecx,%eax
 131         andl    $0x80007fff,%eax
 132         jnz     .Lnot_pseudonormal_2    / zero value implies pseudonormal
 133         addl    $1,%ecx
 134 .Lnot_pseudonormal_2:
 135         movl    8(%ebp),%eax    / low x
 136         addl    $1,%eax         / low x + ulp
 137         movl    %eax,-12(%ebp)
 138         jz      .Lcarry         / jump if the content of %eax is 0
 139         movl    %edx,-8(%ebp)
 140         movl    %ecx,-4(%ebp)
 141         jmp .Lfinal
 142 .Lcarry:
 143         movl    %edx,%eax
 144         andl    $0x7fffffff,%eax
 145         cmpl    $0x7fffffff,%eax        / look at high x
 146         je      .Lsecond_carry
 147         addl    $1,%edx
 148         movl    %edx,-8(%ebp)
 149         movl    %ecx,-4(%ebp)
 150         jmp     .Lfinal
 151 .Lsecond_carry:
 152         movl    $0x80000000,-8(%ebp)
 153         addl    $1,%ecx
 154         movl    %ecx,-4(%ebp)
 155 .Lfinal:
 156         fstp    %st(0)
 157         fldt    -12(%ebp)
 158         andl    $0x00007fff,%ecx
 159         jz      .Lunderflow
 160         cmpw    $0x7fff,%cx
 161         je      .Loverflow
 162         jmp     .Lreturn
 163 .Loverflow:
 164         PIC_SETUP(1)
 165         fldt    PIC_L(.LFmaxl)
 166         PIC_WRAPUP
 167         fmulp   %st,%st(0)      / create overflow signal
 168         jmp     .Lreturn
 169 .Lunderflow:
 170         PIC_SETUP(2)
 171         fldt    PIC_L(.LFminl)
 172         PIC_WRAPUP
 173         fmulp   %st,%st(0)      / create underflow signal
 174         jmp     .Lreturn
 175 .Lequal:
 176         fstp    %st(0)          / C99 says to return y when x == y
 177         jmp     .Lreturn
 178 .LNaN:
 179         faddp   %st,%st(1)      / x+y,x
 180 .Lreturn:
 181         fwait
 182         leave
 183         ret
 184         .align  4
 185         SET_SIZE(nexttowardl)
```

*unchanged_portion_omitted_*

```
**********************************************************
   11164 Tue Nov  4 19:02:20 2014
new/usr/src/lib/libm/i386/src/pow.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "pow.s"

/ Note: 0^NaN should not signal "invalid" but this implementation
/ does because y is placed on the NPX stack.

/ Special cases:
/
/ x ** 0 is 1                           _SVID_libm_err if x is 0 or NaN
/ 1 ** y is 1                           (C99)
/ x ** NaN is NaN
/ NaN ** y (except 0) is NaN
/ x ** 1 is x
/ +-(|x| > 1) **  +inf is +inf
/ +-(|x| > 1) **  -inf is +0
/ +-(|x| < 1) **  +inf is +0
/ +-(|x| < 1) **  -inf is +inf
/ (-1) ** +-inf is +1                   (C99)
/ +0 ** +y (except 0, NaN)              is +0
/ -0 ** +y (except 0, NaN, odd int)     is +0
/ -0 ** +y (odd int)                    is -0
/ +-0 ** -y (except 0, NaN)             _SVID_libm_err
/ +inf ** +y (except 0, NaN)            is +inf
/ +inf ** -y (except 0, NaN)            is +0
/ -inf ** +-y (except 0, NaN)           is -0 ** -+y (NO z flag)
/ x ** -1 is 1/x
/ x ** 2 is x*x
/ -x ** y (an integer) is (-1)**(y) * (+x)**(y)
/ x ** y (x negative & y not integer)   _SVID_libm_err
/ if x and y are finite and x**y = 0    _SVID_libm_err (underflow)
/ if x and y are finite and x**y = inf  _SVID_libm_err (overflow)

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(pow,function)
```



```
#include "libm_synonyms.h"
#include "libm_protos.h"
#include "xpg6.h"

#undef fabs

        .data
        .align  4
negzero:
        .float  -0.0
one:
        .float  1.0
negone:
        .float  -1.0
two:
        .float  2.0
Snan:
        .long   0x7f800001
pinfinity:
        .long   0x7f800000
ninfinity:
        .long   0xff800000


        ENTRY(pow)
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)

        fldl    8(%ebp)                 / x
        fxam                            / determine class of x
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dh                 / %dh <- condition code of x

        fldl    16(%ebp)                / y , x
        fxam                            / determine class of y
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dl                 / %dl <- condition code of y

        call    .pow_main               /// LOCAL
        PIC_WRAPUP
        leave
        ret

.pow_main:
        / x ** 0 is 1 unless x is 0 or a NaN
        movb    %dl,%cl
        andb    $0x45,%cl
        cmpb    $0x40,%cl               / C3=1 C2=0 C1=? C0=0 when +-0
        jne     1f
        movb    %dh,%cl
        andb    $0x45,%cl
        cmpb    $0x40,%cl               / C3=1 C2=0 C1=? C0=0 when +-0
        jne     2f
        / 0^0
        pushl   $20
        jmp     .SVIDerr                / SVID error handler
2:
        cmpb    $0x01,%cl               /// C3=0 C2=0 C1=? C0=1 when +-NaN
        jne     2f
        / NaN^0
        pushl   $42
        jmp     .SVIDerr
2:
        / (not 0 or NaN)^0
        fstp    %st(0)                  / x
```

```
127         fstp    %st(0)                  / stack empty
128         fld1                            / 1
129         ret

131 1:      / y is not zero
132         PIC_G_LOAD(movzwl,__xpg6,eax)
133         andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
134         cmpl    $0,%eax
135         je      1f

137         / C99: 1 ** anything is 1
138         fld1                            / 1, y, x
139         fucomp  %st(2)                  / y, x
140         fnstsw  %ax                     / store status in %ax
141         sahf                            / 80387 flags in %ax to 80386 flags
142         jp      1f                      / so that pow(NaN1,NaN2) returns NaN2
143         jne     1f
144         fstp    %st(0)                  / x
145         ret

147 1:
148         / x ** NaN is NaN
149         movb    %dl,%cl
150         andb    $0x45,%cl
151         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
152         jne     1f
153         fstp    %st(1)                  / y
154         ret

156 1:      / y is not NaN
157         / NaN ** y (except 0) is NaN
158         movb    %dh,%cl
159         andb    $0x45,%cl
160         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
161         jne     1f
162         fstp    %st(0)                  / x
163         ret

165 1:      / x is not NaN
166         / x ** 1 is x
167         fcoms   PIC_L(one)              / y , x
168         fnstsw  %ax                     / store status in %ax
169         sahf                            / 80387 flags in %ax to 80386 flags
170         jne     1f
171         fstp    %st(0)                  / x
172         ret

174 1:      / y is not 1
175         / +-(|x| > 1) **  +inf is +inf
176         / +-(|x| > 1) **  -inf is +0
177         / +-(|x| < 1) **  +inf is +0
178         / +-(|x| < 1) **  -inf is +inf
179         / +-(|x| = 1) ** +-inf is NaN
180         movb    %dl,%cl
181         andb    $0x47,%cl
182         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
183         je      .yispinf
184         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
185         je      .yisninf

187         / +0 ** +y (except 0, NaN)              is +0
188         / -0 ** +y (except 0, NaN, odd int)     is +0
189         / +0 ** -y (except 0, NaN)              is +inf (z flag)
190         / -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
191         / -0 ** y (odd int)                     is - (+0 ** x)
192         movb    %dh,%cl
```



```
193         andb    $0x47,%cl
194         cmpb    $0x40,%cl               / C3=1 C2=0 C1=0 C0=0 when +0
195         je      .xispzero
196         cmpb    $0x42,%cl               / C3=1 C2=0 C1=1 C0=0 when -0
197         je      .xisnzero

199         / +inf ** +y (except 0, NaN)    is +inf
200         / +inf ** -y (except 0, NaN)    is +0
201         / -inf ** +-y (except 0, NaN)   is -0 ** -+y (NO z flag)
202         movb    %dh,%cl
203         andb    $0x47,%cl
204         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
205         je      .xispinf
206         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
207         je      .xisninf

209         / x ** -1 is 1/x
210         fcoms   PIC_L(negone)           / y , x
211         fnstsw  %ax                     / store status in %ax
212         sahf                            / 80387 flags in %ax to 80386 flags
213         jne     1f
214         fld     %st(1)                  / x , y , x
215         fdivrs  PIC_L(one)              / 1/x , y , x
216         jmp     .signok                 / check for over/underflow

218 1:      / y is not -1
219         / x ** 2 is x*x
220         fcoms   PIC_L(two)              / y , x
221         fnstsw  %ax                     / store status in %ax
222         sahf                            / 80387 flags in %ax to 80386 flags
223         jne     1f
224         fld     %st(1)                  / x , y , x
225         fld     %st(0)                  / x , x , y , x
226         fmulp                           / x^2 , y , x
227         jmp     .signok                 / check for over/underflow

229 1:      / y is not 2
230         / make copies of x & y
231         fld     %st(1)                  / x , y , x
232         fld     %st(1)                  / y , x , y , x

234         / -x ** y (an integer) is (-1)**(y) * (+x)**(y)
235         / x ** y (x negative & y not integer) is  NaN
236         movl    $0,%ecx                 / track whether to flip sign of result
237         fld     %st(1)                  / x , y , x , y , x
238         ftst                            / compare %st(0) with 0
239         fnstsw  %ax                     / store status in %ax
240         sahf                            / 80387 flags in %ax to 80386 flags
241         fstp    %st(0)                  / y , x , y , x
242         ja      .merge                  / x > 0
243         / x < 0
244         call    .y_is_int
245         cmpl    $0,%ecx
246         jne     1f
247         / x < 0, y is non-integral
248         fstp    %st(0)                  / x , y , x
249         fstp    %st(0)                  / y , x
250         pushl   $24
251         jmp     .SVIDerr                / SVID error handler

253 1:      / x < 0 & y = int
254         fxch                            / x , y , y , x
255         fchs                            / px = -x , y , y , x
256         fxch                            / y , px , y , x
257 .merge:
258         / px > 0
```

```
259         fxch                            / px , y , y , x

261         / x**y   =   exp(y*ln(x))
262         fyl2x                           / t=y*log2(px) , y , x
263         fld     %st(0)                  / t , t , y , x
264         frndint                         / [t] , t , y , x
265         fxch                            / t , [t] , y , x
266         fucom
267         fnstsw  %ax                     / store status in %ax
268         sahf                            / 80387 flags in %ax to 80386 flags
269         je      1f                      / t is integral
270         fsub    %st(1),%st              / t-[t] , [t] , y , x
271         f2xm1                           / 2**(t-[t])-1 , [t] , y , x
272         fadds   PIC_L(one)              / 2**(t-[t]) , [t] , y , x
273         fscale                          / 2**t = px**y , [t] , y , x
274         jmp     2f
275 1:
276         fstp    %st(0)                  / t=[t] , y , x
277         fld1                            / 1 , t , y , x
278         fscale                          / 1*2**t = x**y , t , y , x
279 2:
280         fstp    %st(1)                  / x**y , y , x
281         cmpl    $1,%ecx
282         jne     .signok
283         fchs                            / change sign since x<0 & y=-int
284 .signok:
285         subl    $8,%esp
286         fstpl   (%esp)                  / round to double precision
287         fldl    (%esp)                  / place result on NPX stack
288         addl    $8,%esp

290         fxam                            / determine class of x**y
291         fnstsw  %ax                     / store status in %ax
292         andw    $0x4500,%ax
293         / check for overflow
294         cmpw    $0x0500,%ax             / C0=0 C1=1 C2=? C3=1 then +-inf
295         jne     1f
296         / x^y overflows
297         fstp    %st(0)                  / y , x
298         pushl   $21
299         jmp     .SVIDerr
300 1:
301         / check for underflow
302         cmpw    $0x4000,%ax             / C0=1 C1=0 C2=? C3=0 then +-0
303         jne     1f
304         / x^y underflows
305         fstp    %st(0)                  / y , x
306         pushl   $22
307         jmp     .SVIDerr
308 1:
309         fstp    %st(2)                  / y , x**y
310         fstp    %st(0)                  / x**y
311         ret

313 / ------------------------------------------------------------------------

315 .xispinf:
316         ftst                            / compare %st(0) with 0
317         fnstsw  %ax                     / store status in %ax
318         sahf                            / 80387 flags in %ax to 80386 flags
319         ja      .retpinf                / y > 0
320         jmp     .retpzero               / y < 0

322 .xisninf:
323         / -inf ** +-y is -0 ** -+y
324         fchs                            / -y , x
```



```
325         flds    PIC_L(negzero)          / -0 , -y , x
326         fstp    %st(2)                  / -y , -0
327         jmp     .xisnzero

329 .yispinf:
330         fld     %st(1)                  / x , y , x
331         fabs                            / |x| , y , x
332         fcomps  PIC_L(one)              / y , x
333         fnstsw  %ax                     / store status in %ax
334         sahf                            / 80387 flags in %ax to 80386 flags
335         je      .retponeorinvalid       / x == -1       C99
336         ja      .retpinf                / |x| > 1
337         jmp     .retpzero               / |x| < 1

339 .yisninf:
340         fld     %st(1)                  / x , y , x
341         fabs                            / |x| , y , x
342         fcomps  PIC_L(one)              / y , x
343         fnstsw  %ax                     / store status in %ax
344         sahf                            / 80387 flags in %ax to 80386 flags
345         je      .retponeorinvalid       / x == -1       C99
346         ja      .retpzero               / |x| > 1
347         jmp     .retpinf                / |x| < 1

349 .xispzero:
350         / y cannot be 0 or NaN ; stack has      y , x
351         ftst                            / compare %st(0) with 0
352         fnstsw  %ax                     / store status in %ax
353         sahf                            / 80387 flags in %ax to 80386 flags
354         ja      .retpzero               / y > 0
355         / x = +0 & y < 0
356         jmp     .SVIDzerotoneg

358 .xisnzero:
359         / y cannot be 0 or NaN ; stack has      y , x
360         call    .y_is_int
361         cmpl    $1,%ecx
362         jne     1f                      / y is not an odd integer
363         / y is an odd integer
364         ftst                            / compare %st(0) with 0
365         fnstsw  %ax                     / store status in %ax
366         sahf                            / 80387 flags in %ax to 80386 flags
367         ja      .retnzero               / y > 0
368         / x = -0 & y < 0 (odd int)      return -inf (z flag)
369         / x = -inf & y != 0 or NaN      return -inf (NO z flag)
370         movb    %dh,%cl
371         andb    $0x45,%cl
372         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
373         jne     .SVIDzerotoneg
374         fstp    %st(0)                  / x
375         fstp    %st(0)                  / stack empty
376         flds    PIC_L(ninfinity)        / -inf
377         ret

379 1:      / y is not an odd integer
380         ftst                            / compare %st(0) with 0
381         fnstsw  %ax                     / store status in %ax
382         sahf                            / 80387 flags in %ax to 80386 flags
383         ja      .retpzero               / y > 0
384         / x = -0 & y < 0 (not odd int)  return +inf (z flag)
385         / x = -inf & y not 0 or NaN     return +inf (NO z flag)
386         movb    %dh,%cl
387         andb    $0x45,%cl
388         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
389         jne     .SVIDzerotoneg
390         jmp     .retpinf                / return +inf (NO z flag)
```

```
392 .retpzero:
393         fstp    %st(0)                  / x
394         fstp    %st(0)                  / stack empty
395         fldz                            / +0
396         ret

398 .retnzero:
399         fstp    %st(0)                  / x
400         fstp    %st(0)                  / stack empty
401         flds    PIC_L(negzero)          / -0
402         ret

404 .retponeorinvalid:
405         PIC_G_LOAD(movzwl,__xpg6,eax)
406         andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
407         cmpl    $0,%eax
408         je      1f
409         fstp    %st(0)                  / x
410         fstp    %st(0)                  / stack empty
411         fld1                            / 1
412         ret

414 1:
415         fstp    %st(0)                  / x
416         fstp    %st(0)                  / stack empty
417         flds    PIC_L(Snan)             / Q NaN (i flag)
418         fwait
419         ret

421 .retpinf:
422         fstp    %st(0)                  / x
423         fstp    %st(0)                  / stack empty
424         flds    PIC_L(pinfinity)        / +inf
425         ret

427 .SVIDzerotoneg:
428         pushl   $23
429 .SVIDerr:
430         / At this point the fp stack contains  y , x  and the number
431         / of the error case has been pushed on the memory stack.
432         subl    $16,%esp
433         fstpl   8(%esp)                 / push y
434         fstpl   (%esp)                  / push x; NPX stack empty
435         call    PIC_F(_SVID_libm_err)   / report result/error according to SVID
436         addl    $20,%esp
437         ret

439 / Set %ecx to 2 if y is an even integer, 1 if y is an odd integer,
440 / 0 otherwise.  Assume y is not zero.  Do not raise inexact or modify
441 / %edx.
442 .y_is_int:
443         movl    20(%ebp),%eax
444         andl    $0x7fffffff,%eax        / |y|
445         cmpl    $0x43400000,%eax
446         jae     1f                      / |y| >= 2^53, an even int
447         cmpl    $0x3ff00000,%eax
448         jb      2f                      / |y| < 1, can't be an int
449         movl    %eax,%ecx
450         sarl    $20,%ecx
451         subl    $0x433,%ecx
452         negl    %ecx                    / 52 - unbiased exponent of y
453         movl    16(%ebp),%eax
454         bsfl    %eax,%eax               / index of least sig. 1 bit
455         jne     3f                      / jump if 1 bit found
456         movl    20(%ebp),%eax
```



```
457         bsfl    %eax,%eax
458         addl    $32,%eax                / 32 + index of least sig. 1 bit
459 3:
460         cmpl    %ecx,%eax
461         jb      2f
462         ja      1f
463         movl    $1,%ecx
464         ret
465 1:
466         movl    $2,%ecx
467         ret
468 2:
469         xorl    %ecx,%ecx
470         ret
471         .align  4
472         SET_SIZE(pow)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   10485 Tue Nov  4 19:02:20 2014
new/usr/src/lib/libm/i386/src/powf.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "powf.s"

/ Note: 0^SNaN should not signal "invalid" but this implementation
/ does because y is placed on the NPX stack.

/ Special cases:
/
/ x ** 0 is 1
/ 1 ** y is 1                           (C99)
/ x ** NaN is NaN
/ NaN ** y (except 0) is NaN
/ x ** 1 is x
/ +-(|x| > 1) **  +inf is +inf
/ +-(|x| > 1) **  -inf is +0
/ +-(|x| < 1) **  +inf is +0
/ +-(|x| < 1) **  -inf is +inf
/ (-1) ** +-inf is +1                   (C99)
/ +0 ** +y (except 0, NaN)              is +0
/ -0 ** +y (except 0, NaN, odd int)     is +0
/ +0 ** -y (except 0, NaN)              is +inf (z flag)
/ -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
/ -0 ** y (odd int)                     is - (+0 ** x)
/ +inf ** +y (except 0, NaN)            is +inf
/ +inf ** -y (except 0, NaN)            is +0
/ -inf ** +-y (except 0, NaN)           is -0 ** -+y (NO z flag)
/ x ** -1 is 1/x
/ x ** 2 is x*x
/ -x ** y (an integer) is (-1)**(y) * (+x)**(y)
/ x ** y (x negative & y not integer) is NaN (i flag)

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(powf,function)
#include "libm_synonyms.h"
```



```
#include "libm_protos.h"
#include "xpg6.h"

#undef fabs

        .data
        .align  4
negzero:
        .float  -0.0
half:
        .float  0.5
one:
        .float  1.0
negone:
        .float  -1.0
two:
        .float  2.0
Snan:
        .long   0x7f800001
pinfinity:
        .long   0x7f800000
ninfinity:
        .long   0xff800000


        ENTRY(powf)
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)

        flds    8(%ebp)                 / x
        fxam                            / determine class of x
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dh                 / %dh <- condition code of x

        flds    12(%ebp)                / y , x
        fxam                            / determine class of y
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dl                 / %dl <- condition code of y

        call    .pow_main               /// LOCAL
        PIC_WRAPUP
        leave
        ret

.pow_main:
        / x ** 0 is 1
        movb    %dl,%cl
        andb    $0x45,%cl
        cmpb    $0x40,%cl               / C3=1 C2=0 C1=? C0=0 when +-0
        jne     1f
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fld1                            / 1
        ret

1:      / y is not zero
        PIC_G_LOAD(movzwl,__xpg6,eax)
        andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
        cmpl    $0,%eax
        je      1f

        / C99: 1 ** anything is 1
        fld1                            / 1, y, x
        fucomp  %st(2)                  / y, x
        fnstsw  %ax                     / store status in %ax
```

```
127         sahf                            / 80387 flags in %ax to 80386 flags
128         jp      1f                      / so that pow(NaN1,NaN2) returns NaN2
129         jne     1f
130         fstp    %st(0)                  / x
131         ret

133 1:
134         / x ** NaN is NaN
135         movb    %dl,%cl
136         andb    $0x45,%cl
137         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
138         jne     1f
139         fstp    %st(1)                  / y
140         ret

142 1:      / y is not NaN
143         / NaN ** y (except 0) is NaN
144         movb    %dh,%cl
145         andb    $0x45,%cl
146         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
147         jne     1f
148         fstp    %st(0)                  / x
149         ret

151 1:      / x is not NaN
152         / x ** 1 is x
153         fcoms   PIC_L(one)              / y , x
154         fnstsw  %ax                     / store status in %ax
155         sahf                            / 80387 flags in %ax to 80386 flags
156         jne     1f
157         fstp    %st(0)                  / x
158         ret

160 1:      / y is not 1
161         / +-(|x| > 1) **  +inf is +inf
162         / +-(|x| > 1) **  -inf is +0
163         / +-(|x| < 1) **  +inf is +0
164         / +-(|x| < 1) **  -inf is +inf
165         / +-(|x| = 1) ** +-inf is NaN
166         movb    %dl,%cl
167         andb    $0x47,%cl
168         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
169         je      .yispinf
170         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
171         je      .yisninf

173         / +0 ** +y (except 0, NaN)              is +0
174         / -0 ** +y (except 0, NaN, odd int)     is +0
175         / +0 ** -y (except 0, NaN)              is +inf (z flag)
176         / -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
177         / -0 ** y (odd int)                     is - (+0 ** x)
178         movb    %dh,%cl
179         andb    $0x47,%cl
180         cmpb    $0x40,%cl               / C3=1 C2=0 C1=0 C0=0 when +0
181         je      .xispzero
182         cmpb    $0x42,%cl               / C3=1 C2=0 C1=1 C0=0 when -0
183         je      .xisnzero

185         / +inf ** +y (except 0, NaN)    is +inf
186         / +inf ** -y (except 0, NaN)    is +0
187         / -inf ** +-y (except 0, NaN)   is -0 ** -+y (NO z flag)
188         movb    %dh,%cl
189         andb    $0x47,%cl
190         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
191         je      .xispinf
192         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
```



```
193         je      .xisninf

195         / x ** -1 is 1/x
196         fcoms   PIC_L(negone)           / y , x
197         fnstsw  %ax                     / store status in %ax
198         sahf                            / 80387 flags in %ax to 80386 flags
199         jne     1f
200         fld     %st(1)                  / x , y , x
201         fdivrs  PIC_L(one)              / 1/x , y , x
202         jmp     .signok                 / check for over/underflow

204 1:      / y is not -1
205         / x ** 2 is square(x)
206         fcoms   PIC_L(two)              / y , x
207         fnstsw  %ax                     / store status in %ax
208         sahf                            / 80387 flags in %ax to 80386 flags
209         jne     1f
210         fld     %st(1)                  / x , y , x
211         fld     %st(0)                  / x , x , y , x
212         fmulp                           / x^2 , y , x
213         jmp     .signok                 / check for over/underflow

215 1:      / y is not 2
216         / x ** 1/2 is sqrt(x)
217         fcoms   PIC_L(half)             / y , x
218         fnstsw  %ax                     / store status in %ax
219         sahf                            / 80387 flags in %ax to 80386 flags
220         jne     1f
221         fld     %st(1)                  / x , y , x
222         fsqrt                           / sqrt(x) , y , x
223         jmp     .signok                 / check for over/underflow

225 1:      / y is not 2
226         / make copies of x & y
227         fld     %st(1)                  / x , y , x
228         fld     %st(1)                  / y , x , y , x

230         / -x ** y (an integer) is (-1)**(y) * (+x)**(y)
231         / x ** y (x negative & y not integer) is  NaN
232         movl    $0,%ecx                 / track whether to flip sign of result
233         fld     %st(1)                  / x , y , x , y , x
234         ftst                            / compare %st(0) with 0
235         fnstsw  %ax                     / store status in %ax
236         sahf                            / 80387 flags in %ax to 80386 flags
237         fstp    %st(0)                  / y , x , y , x
238         ja      .merge                  / x > 0
239         / x < 0
240         call    .y_is_int
241         cmpl    $0,%ecx
242         jne     1f
243         / x < 0 & y != int so x**y = NaN (i flag)
244         fstp    %st(0)                  / x , y , x
245         fstp    %st(0)                  / y , x
246         fstp    %st(0)                  / y , x
247         fstp    %st(0)                  / y , x
248         fldz
249         fdiv    %st,%st(0)              / 0/0
250         ret

252 1:      / x < 0 & y = int
253         fxch                            / x , y , y , x
254         fchs                            / px = -x , y , y , x
255         fxch                            / y , px , y , x
256 .merge:
257         / px > 0
258         fxch                            / px , y , y , x
```

```
260         / x**y   =   exp(y*ln(x))
261         fyl2x                           / t=y*log2(px) , y , x
262         fld     %st(0)                  / t , t , y , x
263         frndint                         / [t] , t , y , x
264         fxch                            / t , [t] , y , x
265         fucom
266         fnstsw  %ax                     / store status in %ax
267         sahf                            / 80387 flags in %ax to 80386 flags
268         je      1f                      / px = int
269         fsub    %st(1),%st              / t-[t] , [t] , y , x
270         f2xm1                           / 2**(t-[t])-1 , [t] , y , x
271         fadds   PIC_L(one)              / 2**(t-[t]) , [t] , y , x
272         fscale                          / 2**t = px**y , [t] , y , x
273         jmp     2f
274 1:
275         fstp    %st(0)                  / t=[t] , y , x
276         fld1                            / 1 , t , y , x
277         fscale                          / 1*2**t = x**y , t , y , x
278 2:
279         fstp    %st(1)                  / x**y , y , x
280         cmpl    $1,%ecx
281         jne     .signok
282         fchs                            / change sign since x<0 & y=-int
283 .signok:
284         subl    $4,%esp
285         fstps   (%esp)                  / round to single precision
286         flds    (%esp)                  / place result on NPX stack
287         addl    $4,%esp
288         fstp    %st(2)                  / y , x**y
289         fstp    %st(0)                  / x**y
290         ret

292 / ------------------------------------------------------------------------

294 .xispinf:
295         ftst                            / compare %st(0) with 0
296         fnstsw  %ax                     / store status in %ax
297         sahf                            / 80387 flags in %ax to 80386 flags
298         ja      .retpinf                / y > 0
299         jmp     .retpzero               / y < 0

301 .xisninf:
302         / -inf ** +-y is -0 ** -+y
303         fchs                            / -y , x
304         flds    PIC_L(negzero)          / -0 , -y , x
305         fstp    %st(2)                  / -y , -0
306         jmp     .xisnzero

308 .yispinf:
309         fld     %st(1)                  / x , y , x
310         fabs                            / |x| , y , x
311         fcomps  PIC_L(one)              / y , x
312         fnstsw  %ax                     / store status in %ax
313         sahf                            / 80387 flags in %ax to 80386 flags
314         je      .retponeorinvalid       / x == -1       C99
315         ja      .retpinf                / |x| > 1
316         jmp     .retpzero               / |x| < 1

318 .yisninf:
319         fld     %st(1)                  / x , y , x
320         fabs                            / |x| , y , x
321         fcomps  PIC_L(one)              / y , x
322         fnstsw  %ax                     / store status in %ax
323         sahf                            / 80387 flags in %ax to 80386 flags
324         je      .retponeorinvalid       / x == -1       C99
```



```
325         ja      .retpzero               / |x| > 1
326         jmp     .retpinf                / |x| < 1

328 .xispzero:
329         / y cannot be 0 or NaN ; stack has      y , x
330         ftst                            / compare %st(0) with 0
331         fnstsw  %ax                     / store status in %ax
332         sahf                            / 80387 flags in %ax to 80386 flags
333         ja      .retpzero               / y > 0
334         / x = +0 & y < 0 so x**y = +inf
335         jmp     .retpinfzflag           / ret +inf & z flag

337 .xisnzero:
338         / y cannot be 0 or NaN ; stack has      y , x
339         call    .y_is_int
340         cmpl    $1,%ecx
341         jne     1f                      / y is not an odd integer
342         / y is an odd integer
343         ftst                            / compare %st(0) with 0
344         fnstsw  %ax                     / store status in %ax
345         sahf                            / 80387 flags in %ax to 80386 flags
346         ja      .retnzero               / y > 0
347         / x = -0 & y < 0 (odd int)      return -inf (z flag)
348         / x = -inf & y != 0 or NaN      return -inf (NO z flag)
349         movb    %dh,%cl
350         andb    $0x45,%cl
351         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
352         je      2f
353         fdiv    %st,%st(1)              / y / x, x (raise z flag)
354 2:
355         fstp    %st(0)                  / x
356         fstp    %st(0)                  / stack empty
357         flds    PIC_L(ninfinity)        / -inf
358         ret

360 1:      / y is not an odd integer
361         ftst                            / compare %st(0) with 0
362         fnstsw  %ax                     / store status in %ax
363         sahf                            / 80387 flags in %ax to 80386 flags
364         ja      .retpzero               / y > 0
365         / x = -0 & y < 0 (not odd int)  return +inf (z flag)
366         / x = -inf & y not 0 or NaN     return +inf (NO z flag)
367         movb    %dh,%cl
368         andb    $0x45,%cl
369         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
370         jne     .retpinfzflag           / ret +inf & divide-by-0 flag
371         jmp     .retpinf                / return +inf (NO z flag)

373 .retpzero:
374         fstp    %st(0)                  / x
375         fstp    %st(0)                  / stack empty
376         fldz                            / +0
377         ret

379 .retnzero:
380         fstp    %st(0)                  / x
381         fstp    %st(0)                  / stack empty
382         flds    PIC_L(negzero)          / -0
383         ret

385 .retponeorinvalid:
386         PIC_G_LOAD(movzwl,__xpg6,eax)
387         andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
388         cmpl    $0,%eax
389         je      1f
390         fstp    %st(0)                  / x
```

```
391         fstp    %st(0)                  / stack empty
392         fld1                            / 1
393         ret

395 1:
396         fstp    %st(0)                  / x
397         fstp    %st(0)                  / stack empty
398         flds    PIC_L(Snan)             / Q NaN (i flag)
399         fwait
400         ret

402 .retpinf:
403         fstp    %st(0)                  / x
404         fstp    %st(0)                  / stack empty
405         flds    PIC_L(pinfinity)        / +inf
406         ret

408 .retpinfzflag:
409         fstp    %st(0)                  / x
410         fstp    %st(0)                  / stack empty
411         fldz
412         fdivrs  PIC_L(one)              / 1/0
413         ret

415 / Set %ecx to 2 if y is an even integer, 1 if y is an odd integer,
416 / 0 otherwise.  Assume y is not zero.  Do not raise inexact or modify
417 / %edx.
418 .y_is_int:
419         movl    12(%ebp),%eax
420         andl    $0x7fffffff,%eax        / |y|
421         cmpl    $0x4b800000,%eax
422         jae     1f                      / |y| >= 2^24, an even int
423         cmpl    $0x3f800000,%eax
424         jb      2f                      / |y| < 1, can't be an int
425         movl    %eax,%ecx
426         sarl    $23,%ecx
427         subl    $150,%ecx
428         negl    %ecx                    / 23 - unbiased exponent of y
429         bsfl    %eax,%eax               / index of least sig. 1 bit
430         cmpl    %ecx,%eax
431         jb      2f
432         ja      1f
433         movl    $1,%ecx
434         ret
435 1:
436         movl    $2,%ecx
437         ret
438 2:
439         xorl    %ecx,%ecx
440         ret
441         .align  4
442         SET_SIZE(powf)
```

*unchanged_portion_omitted_*

```
**********************************************************
   10362 Tue Nov  4 19:02:21 2014
new/usr/src/lib/libm/i386/src/powl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "powl.s"

/ Special cases:
/
/ x ** 0 is 1
/ 1 ** y is 1                           (C99)
/ x ** NaN is NaN
/ NaN ** y (except 0) is NaN
/ x ** 1 is x
/ +-(|x| > 1) **  +inf is +inf
/ +-(|x| > 1) **  -inf is +0
/ +-(|x| < 1) **  +inf is +0
/ +-(|x| < 1) **  -inf is +inf
/ (-1) ** +-inf is +1                   (C99)
/ +0 ** +y (except 0, NaN)              is +0
/ -0 ** +y (except 0, NaN, odd int)     is +0
/ +0 ** -y (except 0, NaN)              is +inf (z flag)
/ -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
/ -0 ** y (odd int)                     is - (+0 ** x)
/ +inf ** +y (except 0, NaN)            is +inf
/ +inf ** -y (except 0, NaN)            is +0
/ -inf ** +-y (except 0, NaN)           is -0 ** -+y (NO z flag)
/ x ** -1 is 1/x
/ x ** 2 is x*x
/ -x ** y (an integer) is (-1)**(y) * (+x)**(y)
/ x ** y (x negative & y not integer) is NaN (i flag)

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(powl,function)
#include "libm_synonyms.h"
#include "xpg6.h"

#undef fabs
```



```
        .data
        .align  4
negzero:
        .float  -0.0
half:
        .float  0.5
one:
        .float  1.0
negone:
        .float  -1.0
two:
        .float  2.0
Snan:
        .long   0x7f800001
pinfinity:
        .long   0x7f800000
ninfinity:
        .long   0xff800000


        ENTRY(powl)
        pushl   %ebp
        movl    %esp,%ebp
        PIC_SETUP(1)

        fldt    8(%ebp)                 / x
        fxam                            / determine class of x
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dh                 / %dh <- condition code of x

        fldt    20(%ebp)                / y , x
        fxam                            / determine class of y
        fnstsw  %ax                     / store status in %ax
        movb    %ah,%dl                 / %dl <- condition code of y

        call    .pow_main               /// LOCAL
        PIC_WRAPUP
        leave
        ret

.pow_main:
        / x ** 0 is 1
        movb    %dl,%cl
        andb    $0x45,%cl
        cmpb    $0x40,%cl               / C3=1 C2=0 C1=? C0=0 when +-0
        jne     1f
        fstp    %st(0)                  / x
        fstp    %st(0)                  / stack empty
        fld1                            / 1
        ret

1:      / y is not zero
        PIC_G_LOAD(movzwl,__xpg6,eax)
        andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
        cmpl    $0,%eax
        je      1f

        / C99: 1 ** anything is 1
        fld1                            / 1, y, x
        fucomp  %st(2)                  / y, x
        fnstsw  %ax                     / store status in %ax
        sahf                            / 80387 flags in %ax to 80386 flags
        jp      1f                      / so that pow(NaN1,NaN2) returns NaN2
        jne     1f
        fstp    %st(0)                  / x
```

```
127         ret

129 1:
130         / x ** NaN is NaN
131         movb    %dl,%cl
132         andb    $0x45,%cl
133         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
134         jne     1f
135         fstp    %st(1)                  / y
136         ret

138 1:      / y is not NaN
139         / NaN ** y (except 0) is NaN
140         movb    %dh,%cl
141         andb    $0x45,%cl
142         cmpb    $0x01,%cl               / C3=0 C2=0 C1=? C0=1 when +-NaN
143         jne     1f
144         fstp    %st(0)                  / x
145         ret

147 1:      / x is not NaN
148         / x ** 1 is x
149         fcoms   PIC_L(one)              / y , x
150         fnstsw  %ax                     / store status in %ax
151         sahf                            / 80387 flags in %ax to 80386 flags
152         jne     1f
153         fstp    %st(0)                  / x
154         ret

156 1:      / y is not 1
157         / +-(|x| > 1) **  +inf is +inf
158         / +-(|x| > 1) **  -inf is +0
159         / +-(|x| < 1) **  +inf is +0
160         / +-(|x| < 1) **  -inf is +inf
161         / +-(|x| = 1) ** +-inf is NaN
162         movb    %dl,%cl
163         andb    $0x47,%cl
164         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
165         je      .yispinf
166         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
167         je      .yisninf

169         / +0 ** +y (except 0, NaN)              is +0
170         / -0 ** +y (except 0, NaN, odd int)     is +0
171         / +0 ** -y (except 0, NaN)              is +inf (z flag)
172         / -0 ** -y (except 0, NaN, odd int)     is +inf (z flag)
173         / -0 ** y (odd int)                     is - (+0 ** x)
174         movb    %dh,%cl
175         andb    $0x47,%cl
176         cmpb    $0x40,%cl               / C3=1 C2=0 C1=0 C0=0 when +0
177         je      .xispzero
178         cmpb    $0x42,%cl               / C3=1 C2=0 C1=1 C0=0 when -0
179         je      .xisnzero

181         / +inf ** +y (except 0, NaN)    is +inf
182         / +inf ** -y (except 0, NaN)    is +0
183         / -inf ** +-y (except 0, NaN)   is -0 ** -+y (NO z flag)
184         movb    %dh,%cl
185         andb    $0x47,%cl
186         cmpb    $0x05,%cl               / C3=0 C2=1 C1=0 C0=1 when +inf
187         je      .xispinf
188         cmpb    $0x07,%cl               / C3=0 C2=1 C1=1 C0=1 when -inf
189         je      .xisninf

191         / x ** -1 is 1/x
192         fcoms   PIC_L(negone)           / y , x
```



```
193         fnstsw  %ax                     / store status in %ax
194         sahf                            / 80387 flags in %ax to 80386 flags
195         jne     1f
196         fld     %st(1)                  / x , y , x
197         fdivrs  PIC_L(one)              / 1/x , y , x
198         jmp     .signok                 / check for over/underflow

200 1:      / y is not -1
201         / x ** 2 is x*x
202         fcoms   PIC_L(two)              / y , x
203         fnstsw  %ax                     / store status in %ax
204         sahf                            / 80387 flags in %ax to 80386 flags
205         jne     1f
206         fld     %st(1)                  / x , y , x
207         fld     %st(0)                  / x , x , y , x
208         fmulp                           / x^2 , y , x
209         jmp     .signok                 / check for over/underflow

211 1:      / y is not 2
212         / x ** 1/2 is sqrt(x)
213         fcoms   PIC_L(half)             / y , x
214         fnstsw  %ax                     / store status in %ax
215         sahf                            / 80387 flags in %ax to 80386 flags
216         jne     1f
217         fld     %st(1)                  / x , y , x
218         fsqrt                           / sqrt(x) , y , x
219         jmp     .signok                 / check for over/underflow

221 1:      / y is not 1/2
222         / make copies of x & y
223         fld     %st(1)                  / x , y , x
224         fld     %st(1)                  / y , x , y , x

226         / -x ** y (an integer) is (-1)**(y) * (+x)**(y)
227         / x ** y (x negative & y not integer) is  NaN
228         movl    $0,%ecx                 / track whether to flip sign of result
229         fld     %st(1)                  / x , y , x , y , x
230         ftst                            / compare %st(0) with 0
231         fnstsw  %ax                     / store status in %ax
232         sahf                            / 80387 flags in %ax to 80386 flags
233         fstp    %st(0)                  / y , x , y , x
234         ja      .merge                  / x > 0
235         / x < 0
236         call    .y_is_int
237         cmpl    $0,%ecx
238         jne     1f
239         / x < 0 & y != int so x**y = NaN (i flag)
240         fstp    %st(0)                  / x , y , x
241         fstp    %st(0)                  / y , x
242         fstp    %st(0)                  / x
243         fstp    %st(0)                  / stack empty
244         fldz
245         fdiv    %st,%st(0)              / 0/0
246         ret

248 1:      / x < 0 & y = int
249         fxch                            / x , y , y , x
250         fchs                            / px = -x , y , y , x
251         fxch                            / y , px , y , x
252 .merge:
253         / px > 0
254         fxch                            / px , y , y , x

256         / x**y   =   exp(y*ln(x))
257         fyl2x                           / t=y*log2(px) , y , x
258         fld     %st(0)                  / t , t , y , x
```

```
259         frndint                         / [t] , t , y , x
260         fxch                            / t , [t] , y , x
261         fucom
262         fnstsw  %ax                     / store status in %ax
263         sahf                            / 80387 flags in %ax to 80386 flags
264         je      1f                      / t is integral
265         fsub    %st(1),%st              / t-[t] , [t] , y , x
266         f2xm1                           / 2**(t-[t])-1 , [t] , y , x
267         fadds   PIC_L(one)              / 2**(t-[t]) , [t] , y , x
268         fscale                          / 2**t = px**y , [t] , y , x
269         jmp     2f
270 1:
271         fstp    %st(0)                  / t=[t] , y , x
272         fld1                            / 1 , t , y , x
273         fscale                          / 1*2**t = x**y , t , y , x
274 2:
275         fstp    %st(1)                  / x**y , y , x
276         cmpl    $1,%ecx
277         jne     .signok
278         fchs                            / change sign since x<0 & y=-int
279 .signok:
280         fstp    %st(2)                  / y , x**y
281         fstp    %st(0)                  / x**y
282         ret

284 / ---------------------------------------------------------------------

286 .xispinf:
287         ftst                            / compare %st(0) with 0
288         fnstsw  %ax                     / store status in %ax
289         sahf                            / 80387 flags in %ax to 80386 flags
290         ja      .retpinf                / y > 0
291         jmp     .retpzero               / y < 0

293 .xisninf:
294         / -inf ** +-y is -0 ** -+y
295         fchs                            / -y , x
296         flds    PIC_L(negzero)          / -0 , -y , x
297         fstp    %st(2)                  / -y , -0
298         jmp     .xisnzero

300 .yispinf:
301         fld     %st(1)                  / x , y , x
302         fabs                            / |x| , y , x
303         fcomps  PIC_L(one)              / y , x
304         fnstsw  %ax                     / store status in %ax
305         sahf                            / 80387 flags in %ax to 80386 flags
306         je      .retponeorinvalid       / x == -1       C99
307         ja      .retpinf                / |x| > 1
308         jmp     .retpzero               / |x| < 1

310 .yisninf:
311         fld     %st(1)                  / x , y , x
312         fabs                            / |x| , y , x
313         fcomps  PIC_L(one)              / y , x
314         fnstsw  %ax                     / store status in %ax
315         sahf                            / 80387 flags in %ax to 80386 flags
316         je      .retponeorinvalid       / x == -1       C99
317         ja      .retpzero               / |x| > 1
318         jmp     .retpinf                / |x| < 1

320 .xispzero:
321         / y cannot be 0 or NaN ; stack has      y , x
322         ftst                            / compare %st(0) with 0
323         fnstsw  %ax                     / store status in %ax
324         sahf                            / 80387 flags in %ax to 80386 flags
```



```
325         ja      .retpzero               / y > 0
326         / x = +0 & y < 0 so x**y = +inf
327         jmp     .retpinfzflag           / ret +inf & z flag

329 .xisnzero:
330         / y cannot be 0 or NaN ; stack has      y , x
331         call    .y_is_int
332         cmpl    $1,%ecx
333         jne     1f                      / y is not an odd integer
334         / y is an odd integer
335         ftst                            / compare %st(0) with 0
336         fnstsw  %ax                     / store status in %ax
337         sahf                            / 80387 flags in %ax to 80386 flags
338         ja      .retnzero               / y > 0
339         / x = -0 & y < 0 (odd int)      return -inf (z flag)
340         / x = -inf & y != 0 or NaN      return -inf (NO z flag)
341         movb    %dh,%cl
342         andb    $0x45,%cl
343         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
344         je      2f
345         fdiv    %st,%st(1)              / y / x, x (raise z flag)
346 2:
347         fstp    %st(0)                  / x
348         fstp    %st(0)                  / stack empty
349         flds    PIC_L(ninfinity)        / -inf
350         ret

352 1:      / y is not an odd integer
353         ftst                            / compare %st(0) with 0
354         fnstsw  %ax                     / store status in %ax
355         sahf                            / 80387 flags in %ax to 80386 flags
356         ja      .retpzero               / y > 0
357         / x = -0 & y < 0 (not odd int)  return +inf (z flag)
358         / x = -inf & y not 0 or NaN     return +inf (NO z flag)
359         movb    %dh,%cl
360         andb    $0x45,%cl
361         cmpb    $0x05,%cl               / C3=0 C2=1 C1=? C0=1 when +-inf
362         jne     .retpinfzflag           / ret +inf & divide-by-0 flag
363         jmp     .retpinf                / return +inf (NO z flag)

365 .retpzero:
366         fstp    %st(0)                  / x
367         fstp    %st(0)                  / stack empty
368         fldz                            / +0
369         ret

371 .retnzero:
372         fstp    %st(0)                  / x
373         fstp    %st(0)                  / stack empty
374         flds    PIC_L(negzero)          / -0
375         ret

377 .retponeorinvalid:
378         PIC_G_LOAD(movzwl,__xpg6,eax)
379         andl    $_C99SUSv3_pow_treats_Inf_as_an_even_int,%eax
380         cmpl    $0,%eax
381         je      1f
382         fstp    %st(0)                  / x
383         fstp    %st(0)                  / stack empty
384         fld1                            / 1
385         ret

387 1:
388         fstp    %st(0)                  / x
389         fstp    %st(0)                  / stack empty
390         flds    PIC_L(Snan)             / Q NaN (i flag)
```

```
 391         fwait
 392         ret

 394 .retpinf:
 395         fstp    %st(0)                  / x
 396         fstp    %st(0)                  / stack empty
 397         flds    PIC_L(pinfinity)        / +inf
 398         ret

 400 .retpinfzflag:
 401         fstp    %st(0)                  / x
 402         fstp    %st(0)                  / stack empty
 403         fldz
 404         fdivrs  PIC_L(one)              / 1/0
 405         ret

 407 / Set %ecx to 2 if y is an even integer, 1 if y is an odd integer,
 408 / 0 otherwise.  Assume y is not zero.  Do not raise inexact or modify
 409 / %edx.
 410 .y_is_int:
 411         movl    28(%ebp),%eax
 412         andl    $0x7fff,%eax            / exponent of y
 413         cmpl    $0x403f,%eax
 414         jae     1f                      / |y| >= 2^64, an even int
 415         cmpl    $0x3fff,%eax
 416         jb      2f                      / |y| < 1, can't be an int
 417         movl    %eax,%ecx
 418         subl    $0x403e,%ecx
 419         negl    %ecx                    / 63 - unbiased exponent of y
 420         movl    20(%ebp),%eax
 421         bsfl    %eax,%eax               / index of least sig. 1 bit
 422         jne     3f                      / jump if 1 bit found
 423         movl    24(%ebp),%eax
 424         bsfl    %eax,%eax
 425         addl    $32,%eax                / 32 + index of least sig. 1 bit
 426 3:
 427         cmpl    %ecx,%eax
 428         jb      2f
 429         ja      1f
 430         movl    $1,%ecx
 431         ret
 432 1:
 433         movl    $2,%ecx
 434         ret
 435 2:
 436         xorl    %ecx,%ecx
 437         ret
 438         .align  4
 439         SET_SIZE(powl)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2101 Tue Nov  4 19:02:21 2014
new/usr/src/lib/libm/i386/src/remainder.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainder.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainder,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(remainder)
        pushl   %ebp
        movl    %esp,%ebp
        fldl    16(%esp)                / load arg y
        fldl    8(%esp)                 / load arg x
        fucom
        fnstsw  %ax
        sahf
        jp      .rem_loop               / if x or y is NaN, use fprem1

        movl    20(%esp),%eax           / eax <-- hi_32(y)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|y|)
        orl     16(%esp),%eax           / eax <-- lo_32(y)|hi_32(|y|)
        je      .yzero_or_xinf

        movl    12(%esp),%eax           / eax <-- hi_32(x)
        andl    $0x7fffffff,%eax        / eax <-- hi_32(|x|)
        cmpl    $0x7ff00000,%eax
        jne     .rem_loop
        cmpl    $0,8(%esp)
        je      .yzero_or_xinf

.rem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check for incomplete reduction
```



```
        jne     .rem_loop               / while incomplete, do fprem1 again
        fstp    %st(1)
        leave
        ret

.yzero_or_xinf:
        PIC_SETUP(1)
        fstp    %st(0)                  / x
        fstp    %st(0)                  / empty NPX stack
        pushl   $28                     / case 28 in _SVID_libm_err
        pushl   20(%ebp)                / pass y
        pushl   16(%ebp)
        pushl   12(%ebp)                / pass x
        pushl   8(%ebp)
        call    PIC_F(_SVID_libm_err)
        addl    $20,%esp
        PIC_WRAPUP
        leave
        ret
        .align  4
        SET_SIZE(remainder)
______unchanged_portion_omitted_
```

****
1399 Tue Nov 4 19:02:22 2014
new/usr/src/lib/libm/i386/src/remainderf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainderf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainderf,function)
#include "libm_synonyms.h"

        ENTRY(remainderf)
        flds    8(%esp)                 / load arg y
        flds    4(%esp)                 / load arg x
.rem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .rem_loop               / while reduction incomplete, do fprem1
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(remainderf)
```

_____***unchanged_portion_omitted_***

****
1399 Tue Nov 4 19:02:22 2014
new/usr/src/lib/libm/i386/src/remainderl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remainderl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remainderl,function)
#include "libm_synonyms.h"

        ENTRY(remainderl)
        fldt    16(%esp)                / load arg y
        fldt    4(%esp)                 / load arg x
.rem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .rem_loop               / while reduction incomplete, do fprem1
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(remainderl)
```

_____unchanged_portion_omitted_

```
**********************************************************
    1904 Tue Nov  4 19:02:22 2014
new/usr/src/lib/libm/i386/src/remquo.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remquo.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remquo,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(remquo)
        fldl    12(%esp)                / load arg y
        fldl    4(%esp)                 / load arg x
.Lrem_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .Lrem_loop              / while reduction incomplete, do fprem1
        fstsw   %ax
        fwait
        fstp    %st(1)
        movw    %ax,%dx
        andw    $0x4000,%dx             / get C3
        sarw    $13,%dx
        movw    %ax,%cx
        andw    $0x100,%cx              / get C0
        sarw    $6,%cx
        addw    %cx,%dx
        andw    $0x200,%ax              / get C1
        sarw    $9,%ax
        addw    %dx,%ax
        cwtl
        movl    8(%esp),%edx            / sign and bexp of x
        movl    16(%esp),%ecx           / sign and bexp of y
        andl    $0x80000000,%edx        / edx <- sign(x)
        andl    $0x80000000,%ecx        / ecx <- sign(y)
```



```
        cmpl    %edx,%ecx
        je      .pos
        negl    %eax                    / negative n
.pos:
        movl    20(%esp),%ecx
        movl    %eax,0(%ecx)            / last 3 significant bits of quotient
        ret
        .align  4
        SET_SIZE(remquo)
______unchanged_portion_omitted_
```

```
**********************************************************
    1909 Tue Nov  4 19:02:23 2014
new/usr/src/lib/libm/i386/src/remquof.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remquof.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remquof,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(remquof)
        flds    8(%esp)                 / load arg y
        flds    4(%esp)                 / load arg x
.Lremf_loop:
        fprem1                          / partial remainder
        fstsw   %ax                     / store status word
        andw    $0x400,%ax              / check whether reduction complete
        jne     .Lremf_loop             / while reduction incomplete, do fprem1
        fstsw   %ax
        fwait
        fstp    %st(1)
        movw    %ax,%dx
        andw    $0x4000,%dx             / get C3
        sarw    $13,%dx
        movw    %ax,%cx
        andw    $0x100,%cx              / get C0
        sarw    $6,%cx
        addw    %cx,%dx
        andw    $0x200,%ax              / get C1
        sarw    $9,%ax
        addw    %dx,%ax
        cwtl
        movl    4(%esp),%edx            / sign and bexp of x
        movl    8(%esp),%ecx            / sign and bexp of y
        andl    $0x80000000,%edx        / edx <- sign(x)
        andl    $0x80000000,%ecx        / ecx <- sign(y)
```



```
        cmpl    %edx,%ecx
        je      .pos
        negl    %eax                    / negative n
.pos:
        movl    12(%esp),%ecx
        movl    %eax,0(%ecx)            / last 3 significant bits of quotient
        ret
        .align  4
        SET_SIZE(remquof)
______unchanged_portion_omitted_
```

```
**********************************************************
    1911 Tue Nov  4 19:02:23 2014
new/usr/src/lib/libm/i386/src/remquol.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "remquol.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(remquol,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(remquol)
        fldt    16(%esp)               / load arg y
        fldt    4(%esp)                / load arg x
.Lreml_loop:
        fprem1                         / partial remainder
        fstsw   %ax                    / store status word
        andw    $0x400,%ax             / check whether reduction complete
        jne     .Lreml_loop            / while reduction incomplete, do fprem1
        fstsw   %ax
        fwait
        fstp    %st(1)
        movw    %ax,%dx
        andw    $0x4000,%dx            / get C3
        sarw    $13,%dx
        movw    %ax,%cx
        andw    $0x100,%cx             / get C0
        sarw    $6,%cx
        addw    %cx,%dx
        andw    $0x200,%ax             / get C1
        sarw    $9,%ax
        addw    %dx,%ax
        cwtl
        movl    12(%esp),%edx          / sign and bexp of x
        movl    24(%esp),%ecx          / sign and bexp of y
        andl    $0x00008000,%edx       / edx <- sign(x)
        andl    $0x00008000,%ecx       / ecx <- sign(y)
```



```
        cmpl    %edx,%ecx
        je      .pos
        negl    %eax                   / negative n
.pos:
        movl    28(%esp),%ecx
        movl    %eax,0(%ecx)           / last 3 significant bits of quotient
        ret
        .align  4
        SET_SIZE(remquol)
______unchanged_portion_omitted_
```

****
1369 Tue Nov 4 19:02:24 2014
new/usr/src/lib/libm/i386/src/rint.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file "rint.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(rint,function)
33 #include "libm_synonyms.h"

34         ENTRY(rint)
35         fldl    4(%esp)                 / load x
36         movl    8(%esp),%eax            / eax <-- hi_32(x)
37         andl    $0x7fffffff,%eax        / eax <-- hi_32(|x|)
38         cmpl    $0x43300000,%eax        / is |x| >= 2**52?
39         jae     .done                   / if so, branch (already integral)
40         frndint                         / [x], per rounding mode
41 .done:
42         fwait
43         ret
44         .align  4
45         SET_SIZE(rint)
```

_____***unchanged_portion_omitted_***

****
1359 Tue Nov 4 19:02:25 2014
new/usr/src/lib/libm/i386/src/rintf.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file "rintf.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(rintf,function)
33 #include "libm_synonyms.h"

34         ENTRY(rintf)
35         flds    4(%esp)                 / load x
36         movl    4(%esp),%eax            / eax <-- x
37         andl    $0x7fffffff,%eax        / eax <-- |x|
38         cmpl    $0x4b000000,%eax        / is |x| >= 2**23?
39         jae     .done                   / if so, branch (already integral)
40         frndint                         / [x], per rounding mode
41 .done:
42         fwait
43         ret
44         .align  4
45         SET_SIZE(rintf)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1186 Tue Nov  4 19:02:25 2014
new/usr/src/lib/libm/i386/src/rintl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file   "rintl.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(rintl,function)
  33 #include "libm_synonyms.h"

  34         ENTRY(rintl)
  35         fldt    4(%esp)                 / load x
  36         frndint                         / [x], per rounding mode
  37         fwait
  38         ret
  39         .align  4
  40         SET_SIZE(rintl)
______unchanged_portion_omitted_
```

```
**********************************************************
    2905 Tue Nov  4 19:02:25 2014
new/usr/src/lib/libm/i386/src/rndintl.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "rndintl.s"

#include "libm.h"

LIBM_ANSI_PRAGMA_WEAK(aintl,function)
LIBM_ANSI_PRAGMA_WEAK(irintl,function)
LIBM_ANSI_PRAGMA_WEAK(anintl,function)
LIBM_ANSI_PRAGMA_WEAK(nintl,function)
#include "libm_synonyms.h"
#undef fabs

        ENTRY(aintl)
        movl    %esp,%eax
        subl    $8,%esp
        fstcw   -8(%eax)
        fldt    4(%eax)
        movw    -8(%eax),%cx
        orw     $0x0c00,%cx
        movw    %cx,-4(%eax)
        fldcw   -4(%eax)                / set RD = to_zero
        frndint
        fstcw   -4(%eax)
        movw    -4(%eax),%dx
        andw    $0xf3ff,%dx
        movw    -8(%eax),%cx
        andw    $0x0c00,%cx
        orw     %dx,%cx
        movw    %cx,-8(%eax)
        fldcw   -8(%eax)                / restore RD
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(aintl)
______unchanged_portion_omitted_
```

```
**********************************************************
    2140 Tue Nov  4 19:02:26 2014
new/usr/src/lib/libm/i386/src/round.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file   "round.s"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(round,function)
  33 #include "libm_synonyms.h"
  33 #undef fabs

  35         .section .rodata
  36         .align  4
  37 .Lhalf: .float  0.5

  39         ENTRY(round)
  40         movl    %esp,%ecx
  41         subl    $8,%esp
  42         fstcw   -8(%ecx)
  43         fldl    4(%ecx)
  44         movw    -8(%ecx),%dx
  45         andw    $0xf3ff,%dx
  46         movw    %dx,-4(%ecx)
  47         fldcw   -4(%ecx)                / set RD = to_nearest
  48         fld     %st(0)
  49         frndint                         / [x],x
  50         fstcw   -4(%ecx)
  51         movw    -4(%ecx),%dx
  52         andw    $0xf3ff,%dx
  53         movw    -8(%ecx),%ax
  54         andw    $0x0c00,%ax
  55         orw     %dx,%ax
  56         movw    %ax,-8(%ecx)
  57         fldcw   -8(%ecx)                / restore RD
  58         fucom                           / check if x is already an integer
  59         fstsw   %ax
  60         sahf
```



```
  61         jp      0f
  62         je      0f
  63         fxch                            / x,[x]
  64         fsub    %st(1),%st              / x-[x],[x]
  65         fabs                            / |x-[x]|,[x]
  66         PIC_SETUP(1)
  67         fcoms   PIC_L(.Lhalf)
  68         PIC_WRAPUP
  69         fnstsw  %ax
  70         sahf
  71         jae     2f                      / if |x-[x]| = 0.5 goto halfway,
  72                                         / most cases will not take branch.
  73 0:
  74         addl    $8,%esp
  75         fstp    %st(0)
  76         ret
  77 2:
  78     / x = n+0.5, recompute round(x) as x+sign(x)*0.5
  79         fldl    4(%ecx)                 / x, 0.5, [x]
  80         movl    8(%ecx),%eax            / high part of x
  81         andl    $0x80000000,%eax
  82         jnz     3f
  83         faddp
  84         addl    $8,%esp
  85         fstp    %st(1)
  86         ret
  87 3:
  88         / here, x is negative, so return x-0.5
  89         fsubp   %st,%st(1)              / x-0.5,[x]
  90         addl    $8,%esp
  91         fstp    %st(1)
  92         ret
  93         .align  4
  94         SET_SIZE(round)
______unchanged_portion_omitted_
```

```
**********************************************************
    2159 Tue Nov  4 19:02:26 2014
new/usr/src/lib/libm/i386/src/roundl.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "roundl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(roundl,function)
#include "libm_synonyms.h"
#undef fabs

        .section .rodata
        .align  4
.Lhalf: .float  0.5

        ENTRY(roundl)
        movl    %esp,%ecx
        subl    $8,%esp
        fstcw   -8(%ecx)
        fldt    4(%ecx)
        movw    -8(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    %dx,-4(%ecx)
        fldcw   -4(%ecx)                / set RD = to_nearest
        fld     %st(0)
        frndint                         / [x],x
        fstcw   -4(%ecx)
        movw    -4(%ecx),%dx
        andw    $0xf3ff,%dx
        movw    -8(%ecx),%ax
        andw    $0x0c00,%ax
        orw     %dx,%ax
        movw    %ax,-8(%ecx)
        fldcw   -8(%ecx)                / restore RD
        fucom                           / check if x is already an integer
        fstsw   %ax
        sahf
```



```
        jp      0f
        je      0f
        fxch                            / x,[x]
        fsub    %st(1),%st              / x-[x],[x]
        fabs                            / |x-[x]|,[x]
        PIC_SETUP(1)
        fcoms   PIC_L(.Lhalf)
        PIC_WRAPUP
        fnstsw  %ax
        sahf
        jae     2f                      / if |x-[x]| = 0.5 goto halfway,
                                        / most cases will not take branch.
0:
        addl    $8,%esp
        fstp    %st(0)
        ret
2:
    / x = n+0.5, recompute roundl(x) as x+sign(x)*0.5
        fldt    4(%ecx)                 / x, 0.5, [x]
        movw    12(%ecx),%ax            / sign+exp of x
        andw    $0x8000,%ax             / look at sign bit
        jnz     3f
        faddp
        addl    $8,%esp
        fstp    %st(1)
        ret
3:
        / here, x is negative, so return x-0.5
        fsubp   %st,%st(1)              / x-0.5,[x]
        addl    $8,%esp
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(roundl)
______unchanged_portion_omitted_
```

```
**********************************************************
    1212 Tue Nov  4 19:02:27 2014
new/usr/src/lib/libm/i386/src/scalbln.s
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "scalbln.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(scalbln,function)
#include "libm_synonyms.h"

        ENTRY(scalbln)
        fildl   12(%esp)                / convert N to extended
        fldl    4(%esp)                 / push x
        fscale
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(scalbln)
______unchanged_portion_omitted_
```

****
1216 Tue Nov 4 19:02:27 2014
new/usr/src/lib/libm/i386/src/scalblnf.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "scalblnf.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(scalblnf,function)
#include "libm_synonyms.h"

        ENTRY(scalblnf)
        fildl   8(%esp)                 / convert N to extended
        flds    4(%esp)                 / push x
        fscale
        fstp    %st(1)
        ret
        .align  4
        SET_SIZE(scalblnf)
```

_____unchanged_portion_omitted_

****
1245 Tue Nov 4 19:02:27 2014
new/usr/src/lib/libm/i386/src/scalblnl.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "scalblnl.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(scalblnl,function)
33 #include "libm_synonyms.h"

34         ENTRY(scalblnl)
35         fildl   16(%esp)                / convert 32-bit integer N
36                                         / to extended-double
37         fldt    4(%esp)                 / push x
38         fscale
39         fstp    %st(1)
40         ret
41         .align  4
42         SET_SIZE(scalblnl)
```

*_____unchanged_portion_omitted_*

****
1208 Tue Nov 4 19:02:28 2014
new/usr/src/lib/libm/i386/src/scalbn.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "scalbn.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(scalbn,function)
33 #include "libm_synonyms.h"

34         ENTRY(scalbn)
35         fildl   12(%esp)                / convert N to extended
36         fldl    4(%esp)                 / push x
37         fscale
38         fstp    %st(1)
39         ret
40         .align  4
41         SET_SIZE(scalbn)
```

_____***unchanged_portion_omitted_***

****
1212 Tue Nov 4 19:02:28 2014
new/usr/src/lib/libm/i386/src/scalbnf.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "scalbnf.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(scalbnf,function)
33 #include "libm_synonyms.h"

34         ENTRY(scalbnf)
35         fildl   8(%esp)                 / convert N to extended
36         flds    4(%esp)                 / push x
37         fscale
38         fstp    %st(1)
39         ret
40         .align  4
41         SET_SIZE(scalbnf)
```

_____unchanged_portion_omitted_

****
1241 Tue Nov 4 19:02:29 2014
new/usr/src/lib/libm/i386/src/scalbnl.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "scalbnl.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(scalbnl,function)
33 #include "libm_synonyms.h"

34         ENTRY(scalbnl)
35         fildl   16(%esp)                / convert 32-bit integer N
36                                         / to extended-double
37         fldt    4(%esp)                 / push x
38         fscale
39         fstp    %st(1)
40         ret
41         .align  4
42         SET_SIZE(scalbnl)
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1322 Tue Nov  4 19:02:29 2014
new/usr/src/lib/libm/i386/src/sin.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "sin.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(sin,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(sin)
        PIC_SETUP(1)
        call    PIC_F(__reduction)
        PIC_WRAPUP
        cmpl    $1,%eax
        jl      .sin0
        je      .sin1
        cmpl    $2,%eax
        je      .sin2
        fcos
        fchs
        ret
.sin2:
        fsin
        fchs
        ret
.sin1:
        fcos
        ret
.sin0:
        fsin
        ret
        .align  4
        SET_SIZE(sin)
```

_____unchanged_portion_omitted_

```
**********************************************************
    1685 Tue Nov  4 19:02:30 2014
new/usr/src/lib/libm/i386/src/sincos.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "sincos.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(sincos,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(sincos)
        PIC_SETUP(1)
        call    PIC_F(__reduction)
        PIC_WRAPUP
        fsincos
        cmpl    $1,%eax
        jl      .sincos0
        je      .sincos1
        cmpl    $2,%eax
        je      .sincos2
        / n=3
        fchs
        movl    12(%esp),%eax
        fstpl   0(%eax)
        movl    16(%esp),%eax
        fstpl   0(%eax)
        fwait
        ret
.sincos2:
        / n=2
        fchs
        movl    16(%esp),%eax
        fstpl   0(%eax)
        fchs
        movl    12(%esp),%eax
        fstpl   0(%eax)
```



```
        fwait
        ret
.sincos1:
        / n=1
        movl    12(%esp),%eax
        fstpl   0(%eax)
        fchs
        movl    16(%esp),%eax
        fstpl   0(%eax)
        fwait
        ret
.sincos0:
        / n=0
        movl    16(%esp),%eax
        fstpl   0(%eax)
        movl    12(%esp),%eax
        fstpl   0(%eax)
        fwait
        ret
        .align  4
        SET_SIZE(sincos)
______unchanged_portion_omitted_
```

****
1145 Tue Nov 4 19:02:30 2014
new/usr/src/lib/libm/i386/src/sqrtl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "sqrtl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(sqrtl,function)
#include "libm_synonyms.h"

        ENTRY(sqrtl)
        fldt    4(%esp)
        fsqrt
        ret
        .align  4
        SET_SIZE(sqrtl)
```

_____unchanged_portion_omitted_

```
**********************************************************
    1293 Tue Nov  4 19:02:31 2014
new/usr/src/lib/libm/i386/src/tan.s
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file "tan.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(tan,function)
#include "libm_synonyms.h"
#include "libm_protos.h"

        ENTRY(tan)
        PIC_SETUP(1)
        call    PIC_F(__reduction)
        PIC_WRAPUP
        andl    $1,%eax
        cmpl    $0,%eax
        je      .tan1
        fptan
        fdivp   %st,%st(1)
        fchs
        ret
.tan1:
        fptan
        fstp    %st(0)
        ret
        .align  4
        SET_SIZE(tan)
```

_____unchanged_portion_omitted_

****
1443 Tue Nov 4 19:02:31 2014
new/usr/src/lib/libm/i386/src/trunc.s
5261 libm should stop using synonyms.h
****

```
 1 /*
 2  * CDDL HEADER START
 3  *
 4  * The contents of this file are subject to the terms of the
 5  * Common Development and Distribution License (the "License").
 6  * You may not use this file except in compliance with the License.
 7  *
 8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 9  * or http://www.opensolaris.org/os/licensing.
10  * See the License for the specific language governing permissions
11  * and limitations under the License.
12  *
13  * When distributing Covered Code, include this CDDL HEADER in each
14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15  * If applicable, add the following below this CDDL HEADER, with the
16  * fields enclosed by brackets "[]" replaced with your own identifying
17  * information: Portions Copyright [yyyy] [name of copyright owner]
18  *
19  * CDDL HEADER END
20  */
21 /*
22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
23  */
24 /*
25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29         .file   "trunc.s"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(trunc,function)
33 #include "libm_synonyms.h"

34         ENTRY(trunc)
35         movl    %esp,%eax
36         subl    $8,%esp
37         fstcw   -8(%eax)
38         fldl    4(%eax)
39         movw    -8(%eax),%cx
40         orw     $0x0c00,%cx
41         movw    %cx,-4(%eax)
42         fldcw   -4(%eax)                / set RD = to_zero
43         frndint
44         fstcw   -4(%eax)
45         movw    -4(%eax),%dx
46         andw    $0xf3ff,%dx
47         movw    -8(%eax),%cx
48         andw    $0x0c00,%cx
49         orw     %dx,%cx
50         movw    %cx,-8(%eax)
51         fldcw   -8(%eax)                / restore RD
52         addl    $8,%esp
53         ret
54         .align  4
55         SET_SIZE(trunc)
```

*_____unchanged_portion_omitted_*

****
1447 Tue Nov 4 19:02:31 2014
new/usr/src/lib/libm/i386/src/truncl.s
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "truncl.s"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(truncl,function)
#include "libm_synonyms.h"

        ENTRY(truncl)
        movl    %esp,%eax
        subl    $8,%esp
        fstcw   -8(%eax)
        fldt    4(%eax)
        movw    -8(%eax),%cx
        orw     $0x0c00,%cx
        movw    %cx,-4(%eax)
        fldcw   -4(%eax)                / set RD = to_zero
        frndint
        fstcw   -4(%eax)
        movw    -4(%eax),%dx
        andw    $0xf3ff,%dx
        movw    -8(%eax),%cx
        andw    $0x0c00,%cx
        orw     %dx,%cx
        movw    %cx,-8(%eax)
        fldcw   -8(%eax)                / restore RD
        addl    $8,%esp
        ret
        .align  4
        SET_SIZE(truncl)
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    1239 Tue Nov  4 19:02:32 2014
new/usr/src/lib/libm/sparc/src/copysign.S
5261 libm should stop using synonyms.h
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "copysign.S"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(copysign,function)
#include "libm_synonyms.h"

        ENTRY(copysign)
        sethi   %hi(0x80000000),%o3
        andn    %o0,%o3,%o0
        and     %o2,%o3,%o2
        or      %o2,%o0,%o0
        std     %o0,[%sp+0x48]
        retl
        ldd     [%sp+0x48],%f0

        SET_SIZE(copysign)
______unchanged_portion_omitted_
```

```
**********************************************************
    1220 Tue Nov  4 19:02:32 2014
new/usr/src/lib/libm/sparc/src/fabs.S
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  23  */
  24 /*
  25  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

  29         .file   "fabs.S"

  31 #include "libm.h"
  32 LIBM_ANSI_PRAGMA_WEAK(fabs,function)
  33 #include "libm_synonyms.h"

  34         ENTRY(fabs)
  35         sethi   %hi(0x80000000),%o2
  36         andn    %o0,%o2,%o0
  37         std     %o0,[%sp+0x48]
  38         nop
  39         nop
  40         nop
  41         nop
  42         nop
  43         nop
  44         retl
  45         ldd     [%sp+0x48],%f0

  47         SET_SIZE(fabs)
______unchanged_portion_omitted_
```

```
**********************************************************
    2545 Tue Nov  4 19:02:33 2014
new/usr/src/lib/libm/sparc/src/nextafter.S
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

        .file   "nextafter.S"

#include "libm.h"
LIBM_ANSI_PRAGMA_WEAK(nextafter,function)
        .weak _nextafter
        .type _nextafter,#function
_nextafter      = __nextafter
#include "libm_synonyms.h"
#include "libm_protos.h"

        RO_DATA
        .align  8
.Lconstant:
two54   = 0x00
        .word   0x43500000,0x0          !  2**54
twom54  = 0x08
        .word   0x3c900000,0x0          !  2**-54
tiny    = 0x10
        .word   0x00100000,0x0          !  tiny

! variable using fp
x       = -0x8
y       = -0x10

        ENTRY(nextafter)
        save    %sp,-128,%sp
        PIC_SETUP(l7)
        std     %i0,[%fp+x]
        or      %g0,%i0,%o0             ! save original arguments
        or      %g0,%i1,%o1
        std     %i2,[%fp+y]
        or      %g0,%i2,%o2
        or      %g0,%i3,%o3
```



```
        ldd     [%fp+x],%f2             ! x
        ldd     [%fp+y],%f0             ! y
        fcmpd   %f2,%f0                 ! x:y
        PIC_SET(l7,.Lconstant,l0)
        sethi   %hi(0x80000000),%l1
        andn    %i0,%l1,%l4
        fbe     9f                      ! next_return
        nop
        fbu,a   9f                      ! next_return
        fmuld   %f2,%f0,%f0             ! + -> * for Cheetah
        orcc    %i1,%l4,%g0             ! see if x is zero
        bne     1f
        tst     %i0
    ! x is zero, return sign(y)*min
        and     %i2,%l1,%i0
        ba      4f                      ! next_final
        mov     1,%i1
1:      bge     2f
        nop
    ! x is negative
        fbl     1f                      ! next_subulp
        nop
        fbg     3f                      ! next_addulp
        nop
2:
        fbl     3f                      ! next_addulp
        nop
1:                                      ! next_subulp
        subcc   %i1,1,%i1
        ba      4f                      ! next_final
        subx    %i0,0,%i0
3:                                      ! next_addulp
        addcc   %i1,1,%i1
        addx    %i0,0,%i0
4:                                      ! next_final
        sethi   %hi(0x7ff00000),%l3
        std     %i0,[%fp+x]
        andcc   %i0,%l3,%i2
        be,a    1f                      ! xflow
        ldd     [%l0+tiny],%f2
        cmp     %i2,%l3
        bne,a   9f                      ! next_return
        ldd     [%fp+x],%f0
        call    NAME(_SVID_libm_err)    ! overflow
        or      %g0,46,%o4
        ba      9f
        nop
1:                                      ! xflow
        fmuld   %f2,%f2,%f2
        ldd     [%fp+x],%f0
9:                                      ! next_return
        ret
        restore

        SET_SIZE(nextafter)
______unchanged_portion_omitted_
```

```
**********************************************************
    5616 Tue Nov  4 19:02:33 2014
new/usr/src/lib/libmvec/Makefile.com
5261 libm should stop using synonyms.h
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#

LIBMDIR         = $(SRC)/lib/libm

mvecOBJS        = \
                  __vTBL_atan1.o \
                  __vTBL_atan2.o \
                  __vTBL_rsqrt.o \
                  __vTBL_sincos.o \
                  __vTBL_sincos2.o \
                  __vTBL_sqrtf.o \
                  __vatan.o \
                  __vatan2.o \
                  __vatan2f.o \
                  __vatanf.o \
                  __vc_abs.o \
                  __vc_exp.o \
                  __vc_log.o \
                  __vc_pow.o \
                  __vcos.o \
                  __vcosbig.o \
                  __vcosbigf.o \
                  __vcosf.o \
                  __vexp.o \
                  __vexpf.o \
                  __vhypot.o \
                  __vhypotf.o \
                  __vlog.o \
                  __vlogf.o \
                  __vpow.o \
                  __vpowf.o \
                  __vrem_pio2m.o \
                  __vrhypot.o \
                  __vrhypotf.o \
                  __vrsqrt.o \
                  __vrsqrtf.o \
                  __vsin.o \
                  __vsinbig.o \
                  __vsinbigf.o \
                  __vsincos.o \
                  __vsincosbig.o \
                  __vsincosbigf.o \
                  __vsincosf.o \
                  __vsinf.o \
                  __vsqrt.o \
                  __vsqrtf.o \
                  __vz_abs.o \
                  __vz_exp.o \
```



```
                  __vz_log.o \
                  __vz_pow.o \
                  vatan2_.o \
                  vatan2f_.o \
                  vatan_.o \
                  vatanf_.o \
                  vc_abs_.o \
                  vc_exp_.o \
                  vc_log_.o \
                  vc_pow_.o \
                  vcos_.o \
                  vcosf_.o \
                  vexp_.o \
                  vexpf_.o \
                  vhypot_.o \
                  vhypotf_.o \
                  vlog_.o \
                  vlogf_.o \
                  vpow_.o \
                  vpowf_.o \
                  vrhypot_.o \
                  vrhypotf_.o \
                  vrsqrt_.o \
                  vrsqrtf_.o \
                  vsin_.o \
                  vsincos_.o \
                  vsincosf_.o \
                  vsinf_.o \
                  vsqrt_.o \
                  vsqrtf_.o \
                  vz_abs_.o \
                  vz_exp_.o \
                  vz_log_.o \
                  vz_pow_.o \
                  #end

mvecvisCOBJS    = \
                  __vTBL_atan1.o \
                  __vTBL_atan2.o \
                  __vTBL_rsqrt.o \
                  __vTBL_sincos.o \
                  __vTBL_sincos2.o \
                  __vTBL_sqrtf.o \
                  __vcosbig.o \
                  __vcosbigf.o \
                  __vrem_pio2m.o \
                  __vsinbig.o \
                  __vsinbigf.o \
                  __vsincosbig.o \
                  __vsincosbigf.o \
                  #end

mvecvisSOBJS    = \
                  __vatan.o \
                  __vatan2.o \
                  __vatan2f.o \
                  __vatanf.o \
                  __vcos.o \
                  __vcosf.o \
                  __vexp.o \
                  __vexpf.o \
                  __vhypot.o \
                  __vhypotf.o \
                  __vlog.o \
                  __vlogf.o \
                  __vpow.o \
```

```
126                 __vpowf.o \
127                 __vrhypot.o \
128                 __vrhypotf.o \
129                 __vrsqrt.o \
130                 __vrsqrtf.o \
131                 __vsin.o \
132                 __vsincos.o \
133                 __vsincosf.o \
134                 __vsinf.o \
135                 __vsqrt.o \
136                 __vsqrtf.o \
139                 #end

138 mvecvis2COBJS   = \
139                 __vTBL_sincos.o \
140                 __vTBL_sincos2.o \
141                 __vTBL_sqrtf.o \
142                 __vcosbig.o \
143                 __vcosbig_ultra3.o \
144                 __vrem_pio2m.o \
145                 __vsinbig.o \
146                 __vsinbig_ultra3.o \
150                 #end

148 mvecvis2SOBJS   = \
149                 __vcos_ultra3.o \
150                 __vlog_ultra3.o \
151                 __vsin_ultra3.o \
152                 __vsqrtf_ultra3.o \
157                 #end

154 include         $(SRC)/lib/Makefile.lib
155 include         $(SRC)/lib/Makefile.rootfs
156 include         $(LIBMDIR)/Makefile.libm.com

158 LIBS            = $(DYNLIB)
159 SRCDIR          = ../common/
160 DYNFLAGS        += $(ZIGNORE)
165 DYNFLAGS        += -zignore

162 LINTERROFF      = -erroff=E_FP_DIVISION_BY_ZERO
163 LINTERROFF      += -erroff=E_FP_INVALID
164 LINTERROFF      += -erroff=E_BAD_PTR_CAST_ALIGN
165 LINTERROFF      += -erroff=E_ASSIGMENT_CAUSE_LOSS_PREC
166 LINTERROFF      += -erroff=E_FUNC_SET_NOT_USED

168 LINTFLAGS       += $(LINTERROFF)
169 LINTFLAGS64     += $(LINTERROFF)
170 LINTFLAGS64     += -errchk=longptr64

172 CLAGS           += $(LINTERROFF)
173 CFLAGS64        += $(LINTERROFF)

175 FLTRPATH_sparc          = $$ORIGIN/cpu/$$ISALIST/libmvec_isa.so.1
176 FLTRPATH_sparcv9        = $$ORIGIN/../cpu/$$ISALIST/sparcv9/libmvec_isa.so.1
177 FLTRPATH_i386           = $$ORIGIN/libmvec/$$HWCAP
178 FLTRPATH                = $(FLTRPATH_$(TARGET_ARCH))

180 sparc_CFLAGS += -_cc=-W0,-xintrinsic
181 sparcv9_CFLAGS += -_cc=-W0,-xintrinsic
182 CPPFLAGS_i386   += -Dfabs=__fabs

184 SRCS_mvec_i386 = \
185         ../common/__vsqrtf.c \
191         #end
```



```
187 SRCS_mvec_sparc = \
188         $(SRCS_mvec_i386) \

195         #end
190 SRCS_mvec_sparcv9 = \
191         $(SRCS_mvec_i386) \

198         #end

194 SRCS_mvec = \
195         $(SRCS_mvec_$(TARGETMACH)) \
196         ../common/__vTBL_atan1.c \
197         ../common/__vTBL_atan2.c \
198         ../common/__vTBL_rsqrt.c \
199         ../common/__vTBL_sincos.c \
200         ../common/__vTBL_sincos2.c \
201         ../common/__vTBL_sqrtf.c \
202         ../common/__vatan.c \
203         ../common/__vatan2.c \
204         ../common/__vatan2f.c \
205         ../common/__vatanf.c \
206         ../common/__vc_abs.c \
207         ../common/__vc_exp.c \
208         ../common/__vc_log.c \
209         ../common/__vc_pow.c \
210         ../common/__vcos.c \
211         ../common/__vcosbig.c \
212         ../common/__vcosbigf.c \
213         ../common/__vcosf.c \
214         ../common/__vexp.c \
215         ../common/__vexpf.c \
216         ../common/__vhypot.c \
217         ../common/__vhypotf.c \
218         ../common/__vlog.c \
219         ../common/__vlogf.c \
220         ../common/__vpow.c \
221         ../common/__vpowf.c \
222         ../common/__vrem_pio2m.c \
223         ../common/__vrhypot.c \
224         ../common/__vrhypotf.c \
225         ../common/__vrsqrt.c \
226         ../common/__vrsqrtf.c \
227         ../common/__vsin.c \
228         ../common/__vsinbig.c \
229         ../common/__vsinbigf.c \
230         ../common/__vsincos.c \
231         ../common/__vsincosbig.c \
232         ../common/__vsincosbigf.c \
233         ../common/__vsincosf.c \
234         ../common/__vsinf.c \
235         ../common/__vsqrt.c \
236         ../common/__vz_abs.c \
237         ../common/__vz_exp.c \
238         ../common/__vz_log.c \
239         ../common/__vz_pow.c \
240         ../common/vatan2_.c \
241         ../common/vatan2f_.c \
242         ../common/vatan_.c \
243         ../common/vatanf_.c \
244         ../common/vc_abs_.c \
245         ../common/vc_exp_.c \
246         ../common/vc_log_.c \
247         ../common/vc_pow_.c \
248         ../common/vcos_.c \
249         ../common/vcosf_.c \
250         ../common/vexp_.c \
```

```
251         ../common/vexpf_.c \
252         ../common/vhypot_.c \
253         ../common/vhypotf_.c \
254         ../common/vlog_.c \
255         ../common/vlogf_.c \
256         ../common/vpow_.c \
257         ../common/vpowf_.c \
258         ../common/vrhypot_.c \
259         ../common/vrhypotf_.c \
260         ../common/vrsqrt_.c \
261         ../common/vrsqrtf_.c \
262         ../common/vsin_.c \
263         ../common/vsincos_.c \
264         ../common/vsincosf_.c \
265         ../common/vsinf_.c \
266         ../common/vsqrt_.c \
267         ../common/vsqrtf_.c \
268         ../common/vz_abs_.c \
269         ../common/vz_exp_.c \
270         ../common/vz_log_.c \
271         ../common/vz_pow_.c \
278         #end

273 .KEEP_STATE:

275 all:    $(LIBS)

277 lint:   lintcheck

279 pics/%.o: ../$(TARGET_ARCH)/src/%.S
280         $(COMPILE.s) -o $@ $<
281         $(POST_PROCESS_O)

283 pics/%.o: ../common/$$(CHIP)/%.S
284         $(COMPILE.s) -o $@ $<
285         $(POST_PROCESS_O)
```

```
**********************************************************
    8456 Tue Nov  4 19:02:34 2014
new/usr/src/lib/libmvec/common/__vhypot.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include <sys/isa_defs.h>
#include "libm_synonyms.h"
#include "libm_inlines.h"

#ifdef _LITTLE_ENDIAN
#define HI(x)   *(1+(int*)x)
#define LO(x)   *(unsigned*)x
#else
#define HI(x)   *(int*)x
#define LO(x)   *(1+(unsigned*)x)
#endif

#ifdef __RESTRICT
#define restrict _Restrict
#else
#define restrict
#endif

/* double hypot(double x, double y)
 *
 * Method :
 *      1. Special cases:
 *              x or y is +Inf or -Inf                          => +Inf
 *              x or y is NaN                                   => QNaN
 *      2. Computes hypot(x,y):
 *              hypot(x,y) = m * sqrt(xnm * xnm + ynm * ynm)
 *      Where:
 *              m = max(|x|,|y|)
 *              xnm = x * (1/m)
 *              ynm = y * (1/m)
 *
 *      Compute xnm * xnm + ynm * ynm by simulating
```



```
 *      muti-precision arithmetic.
 *
 * Accuracy:
 *      Maximum error observed: less than 0.872 ulp after 16.777.216.000
 *      results.
 */

#define sqrt __sqrt

extern double sqrt(double);
extern double fabs(double);

static const unsigned long long LCONST[] = {
0x41b0000000000000ULL,  /* D2ON28 = 2 ** 28             */
0x0010000000000000ULL,  /* D2ONM1022 = 2 ** -1022       */
0x7fd0000000000000ULL   /* D2ONP1022 = 2 **  1022       */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    4145 Tue Nov  4 19:02:35 2014
new/usr/src/lib/libmvec/common/__vhypotf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "libm_synonyms.h"
#include "libm_inlines.h"

#ifdef __RESTRICT
#define restrict _Restrict
#else
#define restrict
#endif

#define sqrt __sqrt

extern double sqrt(double);

void
__vhypotf(int n, float * restrict x, int stridex, float * restrict y,
        int stridey, float * restrict z, int stridez)
{
        float           x0, x1, x2, y0, y1, y2, z0, z1, z2, *pz0, *pz1, *pz2;
        unsigned        hx0, hx1, hx2, hy0, hy1, hy2;
        int                     i, j0, j1, j2;

        do
        {
LOOP0:
                hx0 = *(unsigned*)x & ~0x80000000;
                hy0 = *(unsigned*)y & ~0x80000000;
                *(unsigned*)&x0 = hx0;
                *(unsigned*)&y0 = hy0;
                if (hy0 > hx0)
                {
                        i = hy0 - hx0;
                        j0 = hy0 & 0x7f800000;
```



```
                        if (hx0 == 0)
                                i = 0x7f800000;
                }
                else
                {
                        i = hx0 - hy0;
                        j0 = hx0 & 0x7f800000;
                        if (hy0 == 0)
                                i = 0x7f800000;
                        else if (hx0 == 0)
                                i = 0x7f800000;
                }
                if (i >= 0x0c800000 || j0 >= 0x7f800000)
                {
                        z0 = x0 + y0;
                        if (hx0 == 0x7f800000)
                                z0 = x0;
                        else if (hy0  == 0x7f800000)
                                z0 = y0;
                        else if (hx0 > 0x7f800000 || hy0 > 0x7f800000)
                                z0 = *x + *y;
                        *z = z0;
                        x += stridex;
                        y += stridey;
                        z += stridez;
                        i = 0;
                        if (--n <= 0)
                                break;
                        goto LOOP0;
                }
                pz0 = z;
                x += stridex;
                y += stridey;
                z += stridez;
                i = 1;
                if (--n <= 0)
                        break;

LOOP1:
                hx1 = *(unsigned*)x & ~0x80000000;
                hy1 = *(unsigned*)y & ~0x80000000;
                *(unsigned*)&x1 = hx1;
                *(unsigned*)&y1 = hy1;
                if (hy1 > hx1)
                {
                        i = hy1 - hx1;
                        j1 = hy1 & 0x7f800000;
                        if (hx1 == 0)
                                i = 0x7f800000;
                }
                else
                {
                        i = hx1 - hy1;
                        j1 = hx1 & 0x7f800000;
                        if (hy1 == 0)
                                i = 0x7f800000;
                        else if (hx1 == 0)
                                i = 0x7f800000;
                }
                if (i >= 0x0c800000 || j1 >= 0x7f800000)
                {
                        z1 = x1 + y1;
                        if (hx1 == 0x7f800000)
                                z1 = x1;
                        else if (hy1 == 0x7f800000)
                                z1 = y1;
```

```
125                         else if (hx1 > 0x7f800000 || hy1 > 0x7f800000)
126                                 z1 = *x + *y;
127                         *z = z1;
128                         x += stridex;
129                         y += stridey;
130                         z += stridez;
131                         i = 1;
132                         if (--n <= 0)
133                                 break;
134                         goto LOOP1;
135                 }
136                 pz1 = z;
137                 x += stridex;
138                 y += stridey;
139                 z += stridez;
140                 i = 2;
141                 if (--n <= 0)
142                         break;

144 LOOP2:
145                 hx2 = *(unsigned*)x & ~0x80000000;
146                 hy2 = *(unsigned*)y & ~0x80000000;
147                 *(unsigned*)&x2 = hx2;
148                 *(unsigned*)&y2 = hy2;
149                 if (hy2 > hx2)
150                 {
151                         i = hy2 - hx2;
152                         j2 = hy2 & 0x7f800000;
153                         if (hx2 == 0)
154                                 i = 0x7f800000;
155                 }
156                 else
157                 {
158                         i = hx2 - hy2;
159                         j2 = hx2 & 0x7f800000;
160                         if (hy2 == 0)
161                                 i = 0x7f800000;
162                         else if (hx2 == 0)
163                                 i = 0x7f800000;
164                 }
165                 if (i >= 0x0c800000 || j2 >= 0x7f800000)
166                 {
167                         z2 = x2 + y2;
168                         if (hx2 == 0x7f800000)
169                                 z2 = x2;
170                         else if (hy2 == 0x7f800000)
171                                 z2 = y2;
172                         else if (hx2 > 0x7f800000 || hy2 > 0x7f800000)
173                                 z2 = *x + *y;
174                         *z = z2;
175                         x += stridex;
176                         y += stridey;
177                         z += stridez;
178                         i = 2;
179                         if (--n <= 0)
180                                 break;
181                         goto LOOP2;
182                 }
183                 pz2 = z;

185                 z0 = sqrt(x0 * (double)x0 + y0 * (double)y0);
186                 z1 = sqrt(x1 * (double)x1 + y1 * (double)y1);
187                 z2 = sqrt(x2 * (double)x2 + y2 * (double)y2);
188                 *pz0 = z0;
189                 *pz1 = z1;
190                 *pz2 = z2;
```



```
192                 x += stridex;
193                 y += stridey;
194                 z += stridez;
195                 i = 0;
196         } while (--n > 0);

198         if (i > 0)
199         {
200                 if (i > 1)
201                 {
202                         z1 = sqrt(x1 * (double)x1 + y1 * (double)y1);
203                         *pz1 = z1;
204                 }
205                 z0 = sqrt(x0 * (double)x0 + y0 * (double)y0);
206                 *pz0 = z0;
207         }
208 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   11558 Tue Nov  4 19:02:35 2014
new/usr/src/lib/libmvec/common/__vrhypot.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #include <sys/isa_defs.h>
  31 #include "libm_synonyms.h"
  31 #include "libm_inlines.h"

  33 #ifdef _LITTLE_ENDIAN
  34 #define HI(x)   *(1+(int*)x)
  35 #define LO(x)   *(unsigned*)x
  36 #else
  37 #define HI(x)   *(int*)x
  38 #define LO(x)   *(1+(unsigned*)x)
  39 #endif

  41 #ifdef __RESTRICT
  42 #define restrict _Restrict
  43 #else
  44 #define restrict
  45 #endif

  47 /* double rhypot(double x, double y)
  48  *
  49  * Method :
  50  *      1. Special cases:
  51  *              x or  y = Inf                                   => 0
  52  *              x or  y = NaN                                   => QNaN
  53  *              x and y = 0                                     => Inf + divide-
  54  *      2. Computes rhypot(x,y):
  55  *              rhypot(x,y) = m * sqrt(1/(xnm * xnm + ynm * ynm))
  56  *      Where:
  57  *              m = 1/max(|x|,|y|)
  58  *              xnm = x * m
  59  *              ynm = y * m
  60  *
```



```
  61  *      Compute 1/(xnm * xnm + ynm * ynm) by simulating
  62  *      muti-precision arithmetic.
  63  *
  64  * Accuracy:
  65  *      Maximum error observed: less than 0.869 ulp after 1.000.000.000
  66  *      results.
  67  */

  70 #define sqrt __sqrt

  69 extern double sqrt(double);

  70 extern double fabs(double);

  72 static const int __vlibm_TBL_rhypot[] = {
  73 /* i = [0,127]
  74  * TBL[i] = 0x3ff00000 + *(int*)&(1.0 / *(double*)&(0x3ff0000000000000ULL + (i <
  75  0x7fe00000,  0x7fdfc07f, 0x7fdf81f8,  0x7fdf4465,
  76  0x7fdf07c1,  0x7fdecc07, 0x7fde9131,  0x7fde573a,
  77  0x7fde1e1e,  0x7fdde5d6, 0x7fddae60,  0x7fdd77b6,
  78  0x7fdd41d4,  0x7fdd0cb5, 0x7fdcd856,  0x7fdca4b3,
  79  0x7fdc71c7,  0x7fdc3f8f, 0x7fdc0e07,  0x7fdbdd2b,
  80  0x7fdbacf9,  0x7fdb7d6c, 0x7fdb4e81,  0x7fdb2036,
  81  0x7fdaf286,  0x7fdac570, 0x7fda98ef,  0x7fda6d01,
  82  0x7fda41a4,  0x7fda16d3, 0x7fd9ec8e,  0x7fd9c2d1,
  83  0x7fd99999,  0x7fd970e4, 0x7fd948b0,  0x7fd920fb,
  84  0x7fd8f9c1,  0x7fd8d301, 0x7fd8acb9,  0x7fd886e5,
  85  0x7fd86186,  0x7fd83c97, 0x7fd81818,  0x7fd7f405,
  86  0x7fd7d05f,  0x7fd7ad22, 0x7fd78a4c,  0x7fd767dc,
  87  0x7fd745d1,  0x7fd72428, 0x7fd702e0,  0x7fd6e1f7,
  88  0x7fd6c16c,  0x7fd6a13c, 0x7fd68168,  0x7fd661ec,
  89  0x7fd642c8,  0x7fd623fa, 0x7fd60581,  0x7fd5e75b,
  90  0x7fd5c988,  0x7fd5ac05, 0x7fd58ed2,  0x7fd571ed,
  91  0x7fd55555,  0x7fd53909, 0x7fd51d07,  0x7fd50150,
  92  0x7fd4e5e0,  0x7fd4cab8, 0x7fd4afd6,  0x7fd49539,
  93  0x7fd47ae1,  0x7fd460cb, 0x7fd446f8,  0x7fd42d66,
  94  0x7fd41414,  0x7fd3fb01, 0x7fd3e22c,  0x7fd3c995,
  95  0x7fd3b13b,  0x7fd3991c, 0x7fd38138,  0x7fd3698d,
  96  0x7fd3521c,  0x7fd33ae4, 0x7fd323e3,  0x7fd30d19,
  97  0x7fd2f684,  0x7fd2e025, 0x7fd2c9fb,  0x7fd2b404,
  98  0x7fd29e41,  0x7fd288b0, 0x7fd27350,  0x7fd25e22,
  99  0x7fd24924,  0x7fd23456, 0x7fd21fb7,  0x7fd20b47,
 100  0x7fd1f704,  0x7fd1e2ef, 0x7fd1cf06,  0x7fd1bb4a,
 101  0x7fd1a7b9,  0x7fd19453, 0x7fd18118,  0x7fd16e06,
 102  0x7fd15b1e,  0x7fd1485f, 0x7fd135c8,  0x7fd12358,
 103  0x7fd11111,  0x7fd0fef0, 0x7fd0ecf5,  0x7fd0db20,
 104  0x7fd0c971,  0x7fd0b7e6, 0x7fd0a681,  0x7fd0953f,
 105  0x7fd08421,  0x7fd07326, 0x7fd0624d,  0x7fd05197,
 106  0x7fd04104,  0x7fd03091, 0x7fd02040,  0x7fd01010,
 107 };
______unchanged_portion_omitted_
```

```
**********************************************************
   15231 Tue Nov  4 19:02:36 2014
new/usr/src/lib/libmvec/common/__vrhypotf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #include <sys/isa_defs.h>
  31 #include "libm_synonyms.h"
  31 #include "libm_inlines.h"

  33 #ifdef _LITTLE_ENDIAN
  34 #define HI(x)   *(1+(int*)x)
  35 #define LO(x)   *(unsigned*)x
  36 #else
  37 #define HI(x)   *(int*)x
  38 #define LO(x)   *(1+(unsigned*)x)
  39 #endif

  41 #ifdef __RESTRICT
  42 #define restrict _Restrict
  43 #else
  44 #define restrict
  45 #endif

  47 /* float rhypotf(float x, float y)
  48  *
  49  * Method :
  50  *      1. Special cases:
  51  *              for x or y = Inf                        => 0;
  52  *              for x or y = NaN                        => QNaN;
  53  *              for x and y = 0                         => +Inf + divide-by-zero
  54  *      2. Computes d = x * x + y * y;
  55  *      3. Computes reciprocal square root from:
  56  *              d = m * 2**n
  57  *      Where:
  58  *              m = [0.5, 2),
  59  *              n = ((exponent + 1) & ~1).
  60  *      Then:
```



```
  61  *              rsqrtf(d) = 1/sqrt( m * 2**n ) = (2 ** (-n/2)) * (1/sqrt(m))
  62  *      4. Computes 1/sqrt(m) from:
  63  *              1/sqrt(m) = (1/sqrt(m0)) * (1/sqrt(1 + (1/m0)*dm))
  64  *      Where:
  65  *              m = m0 + dm,
  66  *              m0 = 0.5 * (1 + k/64) for m = [0.5,         0.5+127/256), k = [0
  67  *              m0 = 1.0 * (0 + k/64) for m = [0.5+127/256, 1.0+127/128), k = [6
  68  *      Then:
  69  *              1/sqrt(m0), 1/m0 are looked up in a table,
  70  *              1/sqrt(1 + (1/m0)*dm) is computed using approximation:
  71  *                      1/sqrt(1 + z) = ((a3 * z + a2) * z + a1) * z + a0
  72  *                      where z = [-1/64, 1/64].
  73  *
  74  * Accuracy:
  75  *      The maximum relative error for the approximating
  76  *      polynomial is 2**(-27.87).
  77  *      Maximum error observed: less than 0.535 ulp after 3.000.000.000
  78  *      results.
  79  */

  81 #pragma align 32 (__vlibm_TBL_rhypotf)

  83 static const double __vlibm_TBL_rhypotf[] = {
  84 /*
  85  i = [0,63]
  86  TBL[2*i+0] = 1.0 / (*(double*)&(0x3ff0000000000000LL + (i << 46)));
  87  TBL[2*i+1] = (double)(0.5/sqrtl(2) / sqrtl(*(double*)&(0x3ff0000000000000LL + (
  88  TBL[128+2*i+0] = 1.0 / (*(double*)&(0x3ff0000000000000LL + (i << 46)));
  89  TBL[128+2*i+1] = (double)(0.25 / sqrtl(*(double*)&(0x3ff0000000000000LL + (i <<
  90 */
  91  1.0000000000000000000e+00, 3.5355339059327378637e-01,
  92  9.8461538461538467004e-01, 3.5082320772281166965e-01,
  93  9.6969696969696972388e-01, 3.4815531191139570399e-01,
  94  9.5522388059701490715e-01, 3.4554737023254405992e-01,
  95  9.4117647058823528106e-01, 3.4299717028501769400e-01,
  96  9.2753623188405798228e-01, 3.4050261230349943009e-01,
  97  9.1428571428571425717e-01, 3.3806170189140660742e-01,
  98  9.0140845070422537244e-01, 3.3567254331867563133e-01,
  99  8.8888888888888883955e-01, 3.3333333333333331483e-01,
 100  8.7671232876712323900e-01, 3.3104235544094717802e-01,
 101  8.6486486486486491287e-01, 3.2879797461071458287e-01,
 102  8.5333333333333338810e-01, 3.2659863237109043599e-01,
 103  8.4210526315789469010e-01, 3.2444284226152508843e-01,
 104  8.3116883116883122362e-01, 3.2232918561015211356e-01,
 105  8.2051282051282048435e-01, 3.2025630761017426229e-01,
 106  8.1012658227848100001e-01, 3.1822291367029204023e-01,
 107  8.0000000000000004441e-01, 3.1622776601683794118e-01,
 108  7.9012345679012341293e-01, 3.1426968052735443360e-01,
 109  7.8048780487804880757e-01, 3.1234752377721214378e-01,
 110  7.7108433734939763049e-01, 3.1046021028253312224e-01,
 111  7.6190476190476186247e-01, 3.0860669992418382490e-01,
 112  7.5294117647058822484e-01, 3.0678599553894819740e-01,
 113  7.4418604651162789665e-01, 3.0499714066520933198e-01,
 114  7.3563218390804596680e-01, 3.0323921743156134756e-01,
 115  7.2727272727272729291e-01, 3.0151134457776362918e-01,
 116  7.1910112359550559802e-01, 2.9981267559834456904e-01,
 117  7.1111111111111113825e-01, 2.9814239699997197031e-01,
 118  7.0329670329670335160e-01, 2.9649972666444046610e-01,
 119  6.9565217391304345895e-01, 2.9488391230979427160e-01,
 120  6.8817204301075274309e-01, 2.9329423004270660513e-01,
 121  6.8085106382978721751e-01, 2.9172998299578911663e-01,
 122  6.7368421052631577428e-01, 2.9019050004400465115e-01,
 123  6.6666666666666662966e-01, 2.8867513459481286553e-01,
 124  6.5979381443298967813e-01, 2.8718326344709527165e-01,
 125  6.5306122448979586625e-01, 2.8571428571428569843e-01,
 126  6.4646464646464651960e-01, 2.8426762180748055275e-01,
```

```
  6.4000000000000001332e-01, 2.8284271247461900689e-01,
  6.3366336633663367106e-01, 2.8143901789211672737e-01,
  6.2745098039215685404e-01, 2.8005601680560193723e-01,
  6.2135922330097081989e-01, 2.7869320571664707442e-01,
  6.1538461538461541878e-01, 2.7735009811261457369e-01,
  6.0952380952380957879e-01, 2.7602622373694168934e-01,
  6.0377358490566035432e-01, 2.7472112789737807015e-01,
  5.9813084112149528249e-01, 2.7343437080986532361e-01,
  5.9259259259259255970e-01, 2.7216552697590867815e-01,
  5.8715596330275232617e-01, 2.7091418459143856712e-01,
  5.8181818181818178992e-01, 2.6967994498529684888e-01,
  5.7657657657657657158e-01, 2.6846242208560971987e-01,
  5.7142857142857139685e-01, 2.6726124191242439654e-01,
  5.6637168141592919568e-01, 2.6607604209509572168e-01,
  5.6140350877192979340e-01, 2.6490647141300877054e-01,
  5.5652173913043478937e-01, 2.6375218935831479250e-01,
  5.5172413793103447510e-01, 2.6261286571944508772e-01,
  5.4700854700854706358e-01, 2.6148818018424535570e-01,
  5.4237288135593220151e-01, 2.6037782196164771520e-01,
  5.3781512605042014474e-01, 2.5928148942086576278e-01,
  5.3333333333333332593e-01, 2.5819888974716115326e-01,
  5.2892561983471075848e-01, 2.5712973861329002645e-01,
  5.2459016393442625681e-01, 2.5607375986579195004e-01,
  5.2032520325203257539e-01, 2.5503068522533534068e-01,
  5.1612903225806450180e-01, 2.5400025400038100942e-01,
  5.1200000000000001066e-01, 2.5298221281347033074e-01,
  5.0793650793650790831e-01, 2.5197631533948483540e-01,
  5.0393700787401574104e-01, 2.5098232205526344041e-01,
  1.0000000000000000000e+00, 2.5000000000000000000e-01,
  9.8461538461538467004e-01, 2.4806946917841690703e-01,
  9.6969696969696972388e-01, 2.4618298195866547551e-01,
  9.5522388059701490715e-01, 2.4433888871261044695e-01,
  9.4117647058823528106e-01, 2.4253562503633296910e-01,
  9.2753623188405798228e-01, 2.4077170617153839660e-01,
  9.1428571428571425717e-01, 2.3904572186687872426e-01,
  9.0140845070422537244e-01, 2.3735633163877067897e-01,
  8.8888888888888883955e-01, 2.3570226039551583908e-01,
  8.7671232876712323900e-01, 2.3408229439226113655e-01,
  8.6486486486486491287e-01, 2.3249527748763856860e-01,
  8.5333333333333338810e-01, 2.3094010767585029797e-01,
  8.4210526315789469010e-01, 2.2941573387056177213e-01,
  8.3116883116883122362e-01, 2.2792115291927589338e-01,
  8.2051282051282048435e-01, 2.2645540682891915352e-01,
  8.1012658227848100001e-01, 2.2501758018520479077e-01,
  8.0000000000000004441e-01, 2.2360679774997896385e-01,
  7.9012345679012341293e-01, 2.2222222222222220989e-01,
  7.8048780487804880757e-01, 2.2086305214969309541e-01,
  7.7108433734939763049e-01, 2.1952851997938069295e-01,
  7.6190476190476186247e-01, 2.1821789023599238999e-01,
  7.5294117647058822484e-01, 2.1693045781865616384e-01,
  7.4418604651162789665e-01, 2.1566554640687682354e-01,
  7.3563218390804596680e-01, 2.1442250696755896233e-01,
  7.2727272727272729291e-01, 2.1320071635561044232e-01,
  7.1910112359550559802e-01, 2.1199957600127200541e-01,
  7.1111111111111113825e-01, 2.1081851067789195153e-01,
  7.0329670329670335160e-01, 2.0965696734438366011e-01,
  6.9565217391304345895e-01, 2.0851441405707477061e-01,
  6.8817204301075274309e-01, 2.0739033894608505104e-01,
  6.8085106382978721751e-01, 2.0628424925175867233e-01,
  6.7368421052631577428e-01, 2.0519567041703082322e-01,
  6.6666666666666662966e-01, 2.0412414523193150862e-01,
  6.5979381443298967813e-01, 2.0306923302672380549e-01,
  6.5306122448979586625e-01, 2.0203050891044216364e-01,
  6.4646464646464651960e-01, 2.0100756305184241945e-01,
  6.4000000000000001332e-01, 2.0000000000000001110e-01,
  6.3366336633663367106e-01, 1.9900743804199783060e-01,
```



```
  6.2745098039215685404e-01, 1.9802950859533485772e-01,
  6.2135922330097081989e-01, 1.9706585563285863860e-01,
  6.1538461538461541878e-01, 1.9611613513818404453e-01,
  6.0952380952380957879e-01, 1.9518001458970662965e-01,
  6.0377358490566035432e-01, 1.9425717247145282696e-01,
  5.9813084112149528249e-01, 1.9334729780913270658e-01,
  5.9259259259259255970e-01, 1.9245008972987526219e-01,
  5.8715596330275232617e-01, 1.9156525704423027490e-01,
  5.8181818181818178992e-01, 1.9069251784911847580e-01,
  5.7657657657657657158e-01, 1.8983159915049979682e-01,
  5.7142857142857139685e-01, 1.8898223650461362655e-01,
  5.6637168141592919568e-01, 1.8814417367671945613e-01,
  5.6140350877192979340e-01, 1.8731716231633879777e-01,
  5.5652173913043478937e-01, 1.8650096164806276300e-01,
  5.5172413793103447510e-01, 1.8569533817705186074e-01,
  5.4700854700854706358e-01, 1.8490006540840969729e-01,
  5.4237288135593220151e-01, 1.8411492357966466327e-01,
  5.3781512605042014474e-01, 1.8333969940564226464e-01,
  5.3333333333333332593e-01, 1.8257418583505535814e-01,
  5.2892561983471075848e-01, 1.8181818181818182323e-01,
  5.2459016393442625681e-01, 1.8107149208503706128e-01,
  5.2032520325203257539e-01, 1.8033392693348646030e-01,
  5.1612903225806450180e-01, 1.7960530202677491007e-01,
  5.1200000000000001066e-01, 1.7888543819998317663e-01,
  5.0793650793650790831e-01, 1.7817416127494958844e-01,
  5.0393700787401574104e-01, 1.7747130188322274291e-01,
};

#define fabsf   __fabsf

extern float fabsf(float);

static const double
        A0 = 9.99999997962321453275e-01,
        A1 =-4.99999981660775806600e-01,
        A2 = 3.75066768969515586277e-01,
        A3 =-3.12560092408808548438e-01;

static void
__vrhypotf_n(int n, float * restrict px, int stridex, float * restrict py,
        int stridey, float * restrict pz, int stridez);

#pragma no_inline(__vrhypotf_n)

#define RETURN(ret)                                                     \
{                                                                       \
        *pz = (ret);                                                    \
        pz += stridez;                                                  \
        if (n_n == 0)                                                   \
        {                                                               \
                spx = px; spy = py; spz = pz;                           \
                ay0 = *(int*)py;                                        \
                continue;                                               \
        }                                                               \
        n--;                                                            \
        break;                                                          \
}
______unchanged_portion_omitted_
```

```
**********************************************************
   10451 Tue Nov  4 19:02:36 2014
new/usr/src/lib/libmvec/common/__vrsqrt.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
  24  */
  25 /*
  26  * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
  27  * Use is subject to license terms.
  28  */

  30 #include <sys/isa_defs.h>
  31 #include "libm_synonyms.h"
  31 #include "libm_inlines.h"

  33 #ifdef _LITTLE_ENDIAN
  34 #define HI(x)   *(1+(int*)x)
  35 #define LO(x)   *(unsigned*)x
  36 #else
  37 #define HI(x)   *(int*)x
  38 #define LO(x)   *(1+(unsigned*)x)
  39 #endif

  41 #ifdef __RESTRICT
  42 #define restrict _Restrict
  43 #else
  44 #define restrict
  45 #endif

  47 /* double rsqrt(double x)
  48  *
  49  * Method :
  50  *      1. Special cases:
  51  *              for x = NaN                             => QNaN;
  52  *              for x = +Inf                            => 0;
  53  *              for x is negative, -Inf                 => QNaN + invalid;
  54  *              for x = +0                              => +Inf + divide-by-zero
  55  *              for x = -0                              => -Inf + divide-by-zero
  56  *      2. Computes reciprocal square root from:
  57  *              x = m * 2**n
  58  *      Where:
  59  *              m = [0.5, 2),
  60  *              n = ((exponent + 1) & ~1).
```



```
  61  *      Then:
  62  *              rsqrt(x) = 1/sqrt( m * 2**n ) = (2 ** (-n/2)) * (1/sqrt(m))
  63  *      2. Computes 1/sqrt(m) from:
  64  *              1/sqrt(m) = (1/sqrt(m0)) * (1/sqrt(1 + (1/m0)*dm))
  65  *      Where:
  66  *              m = m0 + dm,
  67  *              m0 = 0.5 * (1 + k/64) for m = [0.5,         0.5+127/256), k = [0
  68  *              m0 = 1.0 * (0 + k/64) for m = [0.5+127/256, 1.0+127/128), k = [6
  69  *              m0 = 2.0              for m = [1.0+127/128, 2.0),         k = 12
  70  *      Then:
  71  *              1/sqrt(m0) is looked up in a table,
  72  *              1/m0 is computed as (1/sqrt(m0)) * (1/sqrt(m0)).
  73  *              1/sqrt(1 + (1/m0)*dm) is computed using approximation:
  74  *                      1/sqrt(1 + z) = (((((a6 * z + a5) * z + a4) * z + a3)
  75  *                                                      * z + a2) * z + a1) * z + a0
  76  *                      where z = [-1/128, 1/128].
  77  *
  78  * Accuracy:
  79  *      The maximum relative error for the approximating
  80  *      polynomial is 2**(-56.26).
  81  *      Maximum error observed: less than 0.563 ulp after 1.500.000.000
  82  *      results.
  83  */

  86 #define sqrt __sqrt

  85 extern double sqrt (double);
  86 extern const double __vlibm_TBL_rsqrt[];

  88 static void
  89 __vrsqrt_n(int n, double * restrict px, int stridex, double * restrict py, int s

  91 #pragma no_inline(__vrsqrt_n)

  93 #define RETURN(ret)                                                     \
  94 {                                                                       \
  95         *py = (ret);                                                  \
  96         py += stridey;                                                \
  97         if (n_n == 0)                                                 \
  98         {                                                             \
  99                 spx = px; spy = py;                                   \
 100                 hx = HI(px);                                          \
 101                 continue;                                             \
 102         }                                                             \
 103         n--;                                                          \
 104         break;                                                        \
 105 }
______unchanged_portion_omitted_
```

```
**********************************************************
   16770 Tue Nov  4 19:02:37 2014
new/usr/src/lib/libmvec/common/__vrsqrtf.c
5261 libm should stop using synonyms.h
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "libm_synonyms.h"
#include "libm_inlines.h"

#ifdef __RESTRICT
#define restrict _Restrict
#else
#define restrict
#endif

/* float rsqrtf(float x)
 *
 * Method :
 *      1. Special cases:
 *              for x = NaN                             => QNaN;
 *              for x = +Inf                            => 0;
 *              for x is negative, -Inf                 => QNaN + invalid;
 *              for x = +0                              => +Inf + divide-by-zero
 *              for x = -0                              => -Inf + divide-by-zero
 *      2. Computes reciprocal square root from:
 *              x = m * 2**n
 *      Where:
 *              m = [0.5, 2),
 *              n = ((exponent + 1) & ~1).
 *      Then:
 *              rsqrtf(x) = 1/sqrt( m * 2**n ) = (2 ** (-n/2)) * (1/sqrt(m))
 *      2. Computes 1/sqrt(m) from:
 *              1/sqrt(m) = (1/sqrt(m0)) * (1/sqrt(1 + (1/m0)*dm))
 *      Where:
 *              m = m0 + dm,
 *              m0 = 0.5 * (1 + k/64) for m = [0.5,         0.5+127/256), k = [0
 *              m0 = 1.0 * (0 + k/64) for m = [0.5+127/256, 1.0+127/128), k = [6
 *      Then:
```



```
 *              1/sqrt(m0), 1/m0 are looked up in a table,
 *              1/sqrt(1 + (1/m0)*dm) is computed using approximation:
 *                      1/sqrt(1 + z) = ((a3 * z + a2) * z + a1) * z + a0
 *                      where z = [-1/64, 1/64].
 *
 * Accuracy:
 *      The maximum relative error for the approximating
 *      polynomial is 2**(-27.87).
 *      Maximum error observed: less than 0.534 ulp for the
 *      whole float type range.
 */

#define sqrtf __sqrtf

extern float sqrtf(float);

static const double __TBL_rsqrtf[] = {
/*
i = [0,63]
 TBL[2*i  ] = 1 / (*(double*)&(0x3fe0000000000000ULL + (i << 46))) * 2**-24;
 TBL[2*i+1] = 1 / sqrtl(*(double*)&(0x3fe0000000000000ULL + (i << 46)));
i = [64,127]
 TBL[2*i  ] = 1 / (*(double*)&(0x3fe0000000000000ULL + (i << 46))) * 2**-23;
 TBL[2*i+1] = 1 / sqrtl(*(double*)&(0x3fe0000000000000ULL + (i << 46)));
*/
 1.192092895507812500e-07, 1.414213562373095145e+00,
 1.173753004807692372e-07, 1.403292830891246678e+00,
 1.155968868371212153e-07, 1.392621247645582816e+00,
 1.138715601679104455e-07, 1.382189480930176239e+00,
 1.121969784007352925e-07, 1.371988681140070776e+00,
 1.105709352355072477e-07, 1.362010449213997720e+00,
 1.089913504464285680e-07, 1.352246807565626429e+00,
 1.074562610035211291e-07, 1.342690173274702525e+00,
 1.059638129340277719e-07, 1.333333333333332593e+00,
 1.045122538527397202e-07, 1.324169421763788712e+00,
 1.030999260979729787e-07, 1.315191898442858331e+00,
 1.017252604166666732e-07, 1.306394529484361744e+00,
 1.003867701480263102e-07, 1.297771369046100353e+00,
 9.908304586038961692e-08, 1.289316742440608454e+00,
 9.781275040064102224e-08, 1.281025230440697049e+00,
 9.657461431962025165e-08, 1.272891654681168160e+00,
 9.536743164062500529e-08, 1.264911064067351764e+00,
 9.419005594135801946e-08, 1.257078722109417734e+00,
 9.304139672256097883e-08, 1.249390095108848575e+00,
 9.192041603915663129e-08, 1.241840841130132489e+00,
 9.082612537202380448e-08, 1.234426799696735299e+00,
 8.975758272058823404e-08, 1.227143982155792789e+00,
 8.871388989825581272e-08, 1.219988562660837327e+00,
 8.769419001436781487e-08, 1.212956869726235902e+00,
 8.669766512784091149e-08, 1.206045378311054516e+00,
 8.572353405898876166e-08, 1.199250702393378276e+00,
 8.477105034722225457e-08, 1.192569587999887881e+00,
 8.383950034306599951e-08, 1.185998906657761864e+00,
 8.292820142663043248e-08, 1.179535649239177086e+00,
 8.203650033602151192e-08, 1.173176920170826420e+00,
 8.116377160904255122e-08, 1.166919931983156466e+00,
 8.030941611842105082e-08, 1.160762000176018604e+00,
 7.947285970052082892e-08, 1.154700538379251462e+00,
 7.865355186855669953e-08, 1.148733053788381086e+00,
 7.785096460459183052e-08, 1.142857142857142793e+00,
 7.706459122474748129e-08, 1.137070487229922211e+00,
 7.629394531250000158e-08, 1.131370849898476027e+00,
 7.553855971534653557e-08, 1.125756071568466909e+00,
 7.479798560049019504e-08, 1.120224067224077489e+00,
 7.407179156553397415e-08, 1.114772822866588297e+00,
 7.335956280048077330e-08, 1.109400392450458294e+00,
```

```
 125  7.26609002976190541 73e-08, 1.1041048949477667573e+00,
 126  7.1975420106132072725e-08, 1.0988845115895122806e+00,
 127  7.1302752628504667579e-08, 1.0937374832394612945e+00,
 128  7.0642541956018514597e-08, 1.0886621079036347126e+00,
 129  6.9994445240825691959e-08, 1.0836567383657542685e+00,
 130  6.9358132102272723904e-08, 1.0787197799411873955e+00,
 131  6.8733284065315314719e-08, 1.0738496883243887795e+00,
 132  6.8119594029017853361e-08, 1.0690449676496975862e+00,
 133  6.7516765763274335346e-08, 1.0643041683803828867e+00,
 134  6.6924513432017540145e-08, 1.0596258856520350822e+00,
 135  6.6342561141304348632e-08, 1.0550087574332591700e+00,
 136  6.5770642510775861156e-08, 1.0504514628777803509e+00,
 137  6.5208500267094023655e-08, 1.0459527207369814228e+00,
 138  6.4655885858050847233e-08, 1.0415112878465908608e+00,
 139  6.4112559086134451001e-08, 1.0371259576834630511e+00,
 140  6.3578287760416665784e-08, 1.0327955589886446131e+00,
 141  6.3052847365702481089e-08, 1.0285189544531601058e+00,
 142  6.2536020747950822927e-08, 1.0242950394631678002e+00,
 143  6.2027597815040656970e-08, 1.0201227409013413627e+00,
 144  6.1527375252016127325e-08, 1.0160010160015240377e+00,
 145  6.1035156250000001271e-08, 1.0119288512538813229e+00,
 146  6.0550750248015869655e-08, 1.0079052613579393416e+00,
 147  6.0073972687007873182e-08, 1.0039292882210537616e+00,
 148  1.1920928955078125000e-07, 1.0000000000000000000e+00,
 149  1.1737530048076923728e-07, 9.9227787671366762812e-01,
 150  1.1559688683712121533e-07, 9.8473192783466190203e-01,
 151  1.1387156016791044559e-07, 9.7735555485044178781e-01,
 152  1.1219697840073529256e-07, 9.7014250014533187638e-01,
 153  1.1057093523550724772e-07, 9.6308682468615358641e-01,
 154  1.0899135044642856803e-07, 9.5618288746751489704e-01,
 155  1.0745626100352112918e-07, 9.4942532655508271588e-01,
 156  1.0596381293402777190e-07, 9.4280904158206335630e-01,
 157  1.0451225385273972023e-07, 9.3632917756904454620e-01,
 158  1.0309992609797297870e-07, 9.2998110995055427441e-01,
 159  1.0172526041666667320e-07, 9.2376043070340119190e-01,
 160  1.0038677014802631022e-07, 9.1766293548224708854e-01,
 161  9.9083045860389616921e-08, 9.1168461167710357351e-01,
 162  9.7812750400641022247e-08, 9.0582162731567661407e-01,
 163  9.6574614319620251657e-08, 9.0007032074081916306e-01,
 164  9.5367431640625005294e-08, 8.9442719099991585541e-01,
 165  9.4190055941358019463e-08, 8.8888888888888883955e-01,
 166  9.3041396722560978838e-08, 8.8345220859877238162e-01,
 167  9.1920416039156631290e-08, 8.7811407991752277180e-01,
 168  9.0826125372023804482e-08, 8.7287156094396955996e-01,
 169  8.9757582720588234048e-08, 8.6772183127462465535e-01,
 170  8.8713889898255812722e-08, 8.6266218562750729415e-01,
 171  8.7694190014367814875e-08, 8.5769002787023584933e-01,
 172  8.6697665127840911497e-08, 8.5280286542244176928e-01,
 173  8.5723534058988761666e-08, 8.4799830400508802164e-01,
 174  8.4771050347222225457e-08, 8.4327404271156780613e-01,
 175  8.3839500343406599951e-08, 8.3862786937753464045e-01,
 176  8.2928201426630432481e-08, 8.3405765622829908246e-01,
 177  8.2036500336021511923e-08, 8.2956135578434020417e-01,
 178  8.1163771609042551220e-08, 8.2513699700703468931e-01,
 179  8.0309416118421050820e-08, 8.2078268166812329287e-01,
 180  7.9472859700520828922e-08, 8.1649658092772603446e-01,
 181  7.8653551868556699530e-08, 8.1227693210689522196e-01,
 182  7.7850964604591830522e-08, 8.0812203564176865456e-01,
 183  7.7064591224747481298e-08, 8.0403025220736967782e-01,
 184  7.6293945312500001588e-08, 8.0000000000000004441e-01,
 185  7.5538559715346535571e-08, 7.9602975216799132241e-01,
 186  7.4797985600490195040e-08, 7.9211803438133943089e-01,
 187  7.4071791565533974158e-08, 7.8826342253143455441e-01,
 188  7.3359562800480773303e-08, 7.8446454055273617811e-01,
 189  7.2660900297619054173e-08, 7.8072005835882651859e-01,
 190  7.1975420106132072725e-08, 7.7702868988581130782e-01,
```



```
 191  7.1302752628504667579e-08, 7.7338919123653082632e-01,
 192  7.0642541956018514597e-08, 7.6980035891950104876e-01,
 193  6.9994445240825691959e-08, 7.6626102817692109959e-01,
 194  6.9358132102272723904e-08, 7.6277007139647390321e-01,
 195  6.8733284065315314719e-08, 7.5932639660199918730e-01,
 196  6.8119594029017853361e-08, 7.5592894601845450619e-01,
 197  6.7516765763274335346e-08, 7.5257669470687782454e-01,
 198  6.6924513432017540145e-08, 7.4926864926535519107e-01,
 199  6.6342561141304348632e-08, 7.4600384659225105199e-01,
 200  6.5770642510775861156e-08, 7.4278135270820744296e-01,
 201  6.5208500267094023655e-08, 7.3960026163363878915e-01,
 202  6.4655885858050847233e-08, 7.3645969431865865307e-01,
 203  6.4112559086134451001e-08, 7.3335879762256905856e-01,
 204  6.3578287760416665784e-08, 7.3029674334022143256e-01,
 205  6.3052847365702481089e-08, 7.2727272727272729291e-01,
 206  6.2536020747950822927e-08, 7.2428596834014824513e-01,
 207  6.2027597815040656970e-08, 7.2133570773394584119e-01,
 208  6.1527375252016127325e-08, 7.1842120810709964029e-01,
 209  6.1035156250000001271e-08, 7.1554175279993270653e-01,
 210  6.0550750248015869655e-08, 7.1269664509979835376e-01,
 211  6.0073972687007873182e-08, 7.0988520753289097165e-01,
 212 };
_____unchanged_portion_omitted_
```

****
1296 Tue Nov 4 19:02:37 2014
new/usr/src/lib/libmvec/common/__vsqrt.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#include "libm_synonyms.h"
#include "libm_inlines.h"

#ifdef __RESTRICT
#define restrict _Restrict
#else
#define restrict
#endif

#define sqrt __sqrt

extern double sqrt(double);

void
__vsqrt(int n, double * restrict x, int stridex, double * restrict y, int stride
{
        for(; n > 0 ; n--)
        {
                *y = sqrt(*x);
                x += stridex;
                y += stridey;
        }
}
```

****
1295 Tue Nov 4 19:02:37 2014
new/usr/src/lib/libmvec/common/__vsqrtf.c
5261 libm should stop using synonyms.h
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifdef __RESTRICT
#define restrict _Restrict
#else
#define restrict
#endif

#include "libm_synonyms.h"
#include "libm_inlines.h"

#define sqrtf __sqrtf

extern float sqrtf(float);

void
__vsqrtf(int n, float * restrict x, int stridex, float * restrict y, int stridey
{
        for(; n > 0 ; n--)
        {
                *y = sqrtf(*x);
                x += stridex;
                y += stridey;
        }
}
```