

```

*****
19602 Mon Oct 27 12:36:38 2014
new/usr/src/lib/libm/Makefile.com
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
14 #
15 #
16 LIBRARY      = libm.a
17 VERS         = .2
18 #
19 LIBMDIR      = $(SRC)/lib/libm
20 #
21 m9xsseOBJS_i386 = \
22     __fex_hdlr.o \
23     __fex_i386.o \
24     __fex_sse.o \
25     __fex_sym.o \
26     fex_log.o
27 #
28 m9xsseOBJS   = $(m9xsseOBJS_$(TARGET_ARCH))
29 #
30 m9xOBJS_amd64 = \
31     __fex_sse.o \
32     feprec.o
33 #
34 m9xOBJS_sparc = \
35     lrint.o \
36     lrintf.o \
37     lrintl.o \
38     lround.o \
39     lroundf.o \
40     lroundl.o
41 #
42 m9xOBJS_i386 = \
43     __fex_sse.o \
44     feprec.o \
45     lrint.o \
46     lrintf.o \
47     lrintl.o \
48     lround.o \
49     lroundf.o \
50     lroundl.o
51 #
52 #
53 # lrint.o, lrintf.o, lrintl.o, lround.o, lroundf.o & lroundl.o are 32-bit only
54 #
55 m9xOBJS      = \
56     $(m9xOBJS_$(TARGET_ARCH)) \
57     __fex_$(MACH).o \
58     __fex_hdlr.o \
59     __fex_sym.o \
60     fdim.o \

```

```

61     fdimf.o \
62     fdiml.o \
63     feexcept.o \
64     fenv.o \
65     feround.o \
66     fex_handler.o \
67     fex_log.o \
68     fma.o \
69     maf.o \
70     fmal.o \
71     fmax.o \
72     fmaxf.o \
73     fmaxl.o \
74     fmin.o \
75     fminf.o \
76     fminl.o \
77     frexp.o \
78     frexpf.o \
79     frexpl.o \
80     ldexp.o \
81     ldexpf.o \
82     ldexpl.o \
83     llrint.o \
84     llrintf.o \
85     llrintl.o \
86     llround.o \
87     llroundf.o \
88     llroundl.o \
89     modf.o \
90     modff.o \
91     modfl.o \
92     nan.o \
93     nanf.o \
94     nanl.o \
95     nearbyint.o \
96     nearbyintf.o \
97     nearbyintl.o \
98     nexttoward.o \
99     nexttowardf.o \
100    nexttowardl.o \
101    remquo.o \
102    remquof.o \
103    remquol.o \
104    round.o \
105    roundf.o \
106    roundl.o \
107    scalbln.o \
108    scalblnf.o \
109    scalblnl.o \
110    tgamma.o \
111    tgammaf.o \
112    tgammal.o \
113    trunc.o \
114    truncf.o \
115    trunc.o \
116
117 OBJM9XSSE   = $(m9xsseOBJS:=%pics/%)
118 #
119 COBJS_i386  = \
120     __libx_errno.o
121 #
122 COBJS_sparc = \
123     $(COBJS_i386) \
124     _TBL_atan.o \
125     _TBL_exp2.o \
126     _TBL_log.o \

```

```

127         _TBL_log2.o \
128         _TBL_tan.o \
129         _tan.o \
130         __tanf.o

132 #
133 # atan2pi.o and sincospi.o is for internal use only
134 #

136 COBJS_amd64 = \
137         _TBL_atan.o \
138         _TBL_exp2.o \
139         _TBL_log.o \
140         _TBL_log2.o \
141         _tan.o \
142         __tanf.o \
143         _TBL_tan.o \
144         copysign.o \
145         exp.o \
146         fabs.o \
147         fmod.o \
148         ilogb.o \
149         isnan.o \
150         nextafter.o \
151         remainder.o \
152         rint.o \
153         scalbn.o

155 COBJS_sparcv9 = $(COBJS_amd64)

157 COBJS      = \
158         $(COBJS_$(TARGET_ARCH)) \
159         _cos.o \
160         _lgamma.o \
161         _rem_pio2.o \
162         _rem_pio2m.o \
163         _sin.o \
164         _sincos.o \
165         _xpg6.o \
166         _lib_version.o \
167         _SVID_error.o \
168         _TBL_ipio2.o \
169         _TBL_sin.o \
170         acos.o \
171         acosh.o \
172         asin.o \
173         asinh.o \
174         atan.o \
175         atan2.o \
176         atan2pi.o \
177         atanh.o \
178         cbrt.o \
179         ceil.o \
180         cos.o \
181         cosh.o \
182         erf.o \
183         expl0.o \
184         exp2.o \
185         expm1.o \
186         floor.o \
187         gamma.o \
188         gamma_r.o \
189         hypot.o \
190         j0.o \
191         j1.o \
192         jn.o \

```

```

193         lgamma.o \
194         lgamma_r.o \
195         log.o \
196         log10.o \
197         loglp.o \
198         log2.o \
199         logb.o \
200         matherr.o \
201         pow.o \
202         scalb.o \
203         signgam.o \
204         significand.o \
205         sin.o \
206         sincos.o \
207         sincospi.o \
208         sinh.o \
209         sqrt.o \
210         tan.o \
211         tanh.o

213 #
214 # LSARC/2003/658 adds isnanl
215 #
216 QOBSJS_sparc      = \
217         _TBL_atanl.o \
218         _TBL_expl.o \
219         _TBL_expm1l.o \
220         _TBL_logl.o \
221         finitel.o \
222         isnanl.o

224 QOBSJS_sparcv9   = $(QOBSJS_sparc)

226 QOBSJS_amd64     = \
227         finitel.o \
228         isnanl.o

230 #
231 # atan2pil.o, ieee_func1.o, rndintl.o, sinpil.o, sincospil.o
232 # are for internal use only
233 #
234 # LSARC/2003/279 adds the following:
235 #          gammal.o          1
236 #          gammal_r.o       1
237 #          j0l.o            2
238 #          j1l.o            2
239 #          jnl.o            2
240 #          lgammal_r.o      1
241 #          scalbl.o         1
242 #          significandl.o   1
243 #
244 QOBSJS           = \
245         $(QOBSJS_$(TARGET_ARCH)) \
246         _cosl.o \
247         _lgammal.o \
248         _poly_libmq.o \
249         _rem_pio2l.o \
250         _sincosl.o \
251         _sinl.o \
252         _tanl.o \
253         _TBL_cosl.o \
254         _TBL_ipio2l.o \
255         _TBL_sinl.o \
256         _TBL_tanl.o \
257         acoshl.o \
258         acosl.o \

```

```

259         asinhl.o \
260         asinl.o \
261         atan2l.o \
262         atan2pil.o \
263         atanh.o \
264         atan.o \
265         cbrt.o \
266         copysignl.o \
267         cosh.o \
268         cosl.o \
269         erfl.o \
270         expl0l.o \
271         exp2l.o \
272         expl.o \
273         expml.o \
274         fabs.o \
275         floor.o \
276         fmod.o \
277         gamma.o \
278         gamma_r.o \
279         hypot.o \
280         ieee_func.o \
281         ilogb.o \
282         j0.o \
283         j1.o \
284         jnl.o \
285         lgamma.o \
286         lgamma_r.o \
287         log10.o \
288         logl.o \
289         log2.o \
290         logb.o \
291         log.o \
292         nextafter.o \
293         pow.o \
294         remainder.o \
295         rint.o \
296         rndint.o \
297         scalb.o \
298         scalbn.o \
299         signgam.o \
300         significand.o \
301         sincos.o \
302         sincospil.o \
303         sin.o \
304         sinl.o \
305         sinpil.o \
306         sqrt.o \
307         tanh.o \
308         tan.o \
310 #
311 # LSARC/2003/658 adds isnanf
312 #
313 ROBJs_sparc = \
314     __cosf.o \
315     __sincosf.o \
316     __sinf.o \
317     isnanf.o
319 ROBJs_sparcv9 = $(ROBJs_sparc)
321 ROBJs_amd64 = \
322     isnanf.o \
323     __cosf.o \
324     __sincosf.o \

```

```

325         __sinf.o
327 #
328 # atan2pif.o, sincosf.o, sincospif.o are for internal use only
329 #
330 # LSARC/2003/279 adds the following:
331 #         besself.o         6
332 #         scalbf.o         1
333 #         gammaf.o         1
334 #         gammaf_r.o       1
335 #         lgammaf_r.o      1
336 #         significandf.o   1
337 #
338 ROBJs = \
339     $(ROBJs_$(TARGET_ARCH)) \
340     _TBL_r_atan.o \
341     acosf.o \
342     acoshf.o \
343     asinf.o \
344     asinhf.o \
345     atan2f.o \
346     atan2pif.o \
347     atanf.o \
348     atanhf.o \
349     besself.o \
350     cbrtf.o \
351     copysignf.o \
352     cosf.o \
353     coshf.o \
354     erff.o \
355     expl0f.o \
356     exp2f.o \
357     expf.o \
358     expmlf.o \
359     fabsf.o \
360     floorf.o \
361     fmodf.o \
362     gammaf.o \
363     gammaf_r.o \
364     hypotf.o \
365     ilogbf.o \
366     lgammaf.o \
367     lgammaf_r.o \
368     log10f.o \
369     loglpf.o \
370     log2f.o \
371     logbf.o \
372     logf.o \
373     nextafterf.o \
374     powf.o \
375     remainderf.o \
376     rintf.o \
377     scalbf.o \
378     scalbnf.o \
379     signgamf.o \
380     significandf.o \
381     sinf.o \
382     sinhf.o \
383     sincosf.o \
384     sincospif.o \
385     sqrtf.o \
386     tanf.o \
387     tanhf.o
389 #
390 # LSARC/2003/658 adds isnanf/isnanl

```

```

391 #
393 SOBJS_sparc = \
394     copysign.o \
395     exp.o \
396     fabs.o \
397     fmod.o \
398     ilogb.o \
399     isnan.o \
400     nextafter.o \
401     remainder.o \
402     rint.o \
403     scalbn.o
405 SOBJS_i386 = \
406     __reduction.o \
407     finitef.o \
408     finitel.o \
409     isnanf.o \
410     isnanl.o \
411     $(SOBJS_sparc)
413 SOBJS_amd64 = \
414     __swapFLAGS.o
415 #     _xtoll.o \
416 #     _xtoull.o
419 SOBJS = \
420     $(SOBJS_$(TARGET_ARCH))
422 complexOBJJS = \
423     cabs.o \
424     cabsf.o \
425     cabsl.o \
426     cacos.o \
427     cacosf.o \
428     cacosh.o \
429     cacoshf.o \
430     cacoshl.o \
431     cacosl.o \
432     carg.o \
433     cargf.o \
434     cargl.o \
435     casin.o \
436     casinf.o \
437     casin.h.o \
438     casinhf.o \
439     casinhl.o \
440     casinl.o \
441     catan.o \
442     catanf.o \
443     catanh.o \
444     catanhf.o \
445     catanhl.o \
446     catanl.o \
447     ccos.o \
448     ccosf.o \
449     ccosh.o \
450     ccoshf.o \
451     ccoshl.o \
452     ccosl.o \
453     cexp.o \
454     cexpf.o \
455     cexpl.o \
456     cimag.o \

```

```

457     cimagf.o \
458     cimagl.o \
459     clog.o \
460     clogf.o \
461     clogl.o \
462     conj.o \
463     conjf.o \
464     conjl.o \
465     cpow.o \
466     cpowf.o \
467     cpowl.o \
468     cproj.o \
469     cprojf.o \
470     cprojl.o \
471     creal.o \
472     crealf.o \
473     creall.o \
474     csin.o \
475     csinf.o \
476     csinh.o \
477     csinhf.o \
478     csinhl.o \
479     csinl.o \
480     csqrt.o \
481     csqrtf.o \
482     csqrtl.o \
483     ctan.o \
484     ctanf.o \
485     ctanh.o \
486     ctanhf.o \
487     ctanhl.o \
488     ctanl.o \
489     k_atan2.o \
490     k_atan2l.o \
491     k_cexp.o \
492     k_cexpl.o \
493     k_clog_r.o \
494     k_clog_rl.o
496 OBJECTS = $(COBJS) $(ROBJS) $(QOBJS) $(SOBJS) $(m9xOBJJS) $(complexOBJJS)
498 include $(SRC)/lib/Makefile.lib
499 include $(LIBMDIR)/Makefile.libm.com
500 include $(SRC)/lib/Makefile.rootfs
502 SRCDIR = ../common/
503 LIBS = $(DYNLIB) $(LINTLIB)
505 LINTERROFF = -erroff=E_FUNC_SET_NOT_USED
506 LINTERROFF += -erroff=E_FUNC_RET_ALWAYS_IGNORED
507 LINTERROFF += -erroff=E_FUNC_RET_MAYBE_IGNORED2
508 LINTERROFF += -erroff=E_IMPL_CONV_RETURN
509 LINTERROFF += -erroff=E_NAME_MULTIPLY_DEF2
510 LINTFLAGS += $(LINTERROFF)
511 LINTFLAGS64 += $(LINTERROFF)
512 LINTFLAGS64 += -errchk=longptr64
514 CPPFLAGS += -DLIBM_BUILD
514 CFLAGS += $(C_BIGPICFLAGS)
515 CFLAGS64 += $(C_BIGPICFLAGS)
517 m9x_IL = $(LIBMDIR)/common/m9x/___fenv_$(TARGET_ARCH).il
519 SRCS_LD_i386_amd64 = \
520     ../common/LD/finitel.c \

```

```

521 ../common/LD/isnanl.c \
522 ../common/LD/nextafterl.c

524 SRCS_LD = \
525 $(SRCS_LD_i386_${TARGET_ARCH}) \
526 ../common/LD/__cosl.c \
527 ../common/LD/__lgammal.c \
528 ../common/LD/__poly_libmq.c \
529 ../common/LD/__rem_pio2l.c \
530 ../common/LD/__sincosl.c \
531 ../common/LD/__sinl.c \
532 ../common/LD/__tanl.c \
533 ../common/LD/_TBL_cosl.c \
534 ../common/LD/_TBL_ipio2l.c \
535 ../common/LD/_TBL_sinl.c \
536 ../common/LD/_TBL_tanl.c \
537 ../common/LD/acoshl.c \
538 ../common/LD/asinhl.c \
539 ../common/LD/atan2pil.c \
540 ../common/LD/atanhl.c \
541 ../common/LD/cbrtl.c \
542 ../common/LD/coshl.c \
543 ../common/LD/cosl.c \
544 ../common/LD/erfl.c \
545 ../common/LD/gammal.c \
546 ../common/LD/gammal_r.c \
547 ../common/LD/hypotl.c \
548 ../common/LD/j0l.c \
549 ../common/LD/j1l.c \
550 ../common/LD/jnl.c \
551 ../common/LD/lgammal.c \
552 ../common/LD/lgammal_r.c \
553 ../common/LD/loglpl.c \
554 ../common/LD/logdbl.c \
555 ../common/LD/scalbl.c \
556 ../common/LD/signgaml.c \
557 ../common/LD/significandl.c \
558 ../common/LD/sincosl.c \
559 ../common/LD/sincospil.c \
560 ../common/LD/sinhl.c \
561 ../common/LD/sinl.c \
562 ../common/LD/sinpil.c \
563 ../common/LD/tanhl.c \
564 ../common/LD/tanl.c

566 SRCS_LD_i386 = \
567 $(SRCS_LD)

569 SRCS_R_amd64 = \
570 ../common/R/__tanf.c \
571 ../common/R/isnanf.c \
572 ../common/R/__cosf.c \
573 ../common/R/__sincosf.c \
574 ../common/R/__sinf.c \
575 ../common/R/acosf.c \
576 ../common/R/asinf.c \
577 ../common/R/atan2f.c \
578 ../common/R/copysignf.c \
579 ../common/R/exp10f.c \
580 ../common/R/exp2f.c \
581 ../common/R/expmlf.c \
582 ../common/R/fabsf.c \
583 ../common/R/hypotf.c \
584 ../common/R/ilogbf.c \
585 ../common/R/log10f.c \
586 ../common/R/log2f.c \

```

```

587 ../common/R/nextafterf.c \
588 ../common/R/powf.c \
589 ../common/R/rintf.c \
590 ../common/R/scalbnf.c

592 # sparc + sparcv9
593 SRCS_R_sparc = \
594 ../common/R/__tanf.c \
595 ../common/R/__cosf.c \
596 ../common/R/__sincosf.c \
597 ../common/R/__sinf.c \
598 ../common/R/isnanf.c \
599 ../common/R/acosf.c \
600 ../common/R/asinf.c \
601 ../common/R/atan2f.c \
602 ../common/R/copysignf.c \
603 ../common/R/exp10f.c \
604 ../common/R/exp2f.c \
605 ../common/R/expmlf.c \
606 ../common/R/fabsf.c \
607 ../common/R/fmodf.c \
608 ../common/R/hypotf.c \
609 ../common/R/ilogbf.c \
610 ../common/R/log10f.c \
611 ../common/R/log2f.c \
612 ../common/R/nextafterf.c \
613 ../common/R/powf.c \
614 ../common/R/remainderf.c \
615 ../common/R/rintf.c \
616 ../common/R/scalbnf.c

618 SRCS_R = \
619 $(SRCS_R_${MACH}) \
620 $(SRCS_R_${TARGET_ARCH}) \
621 ../common/R/_TBL_r_atan.c \
622 ../common/R/acoshf.c \
623 ../common/R/asinhf.c \
624 ../common/R/atan2pif.c \
625 ../common/R/atanf.c \
626 ../common/R/atanhf.c \
627 ../common/R/besself.c \
628 ../common/R/cbrtf.c \
629 ../common/R/cosf.c \
630 ../common/R/coshf.c \
631 ../common/R/erff.c \
632 ../common/R/expf.c \
633 ../common/R/floorf.c \
634 ../common/R/gammaf.c \
635 ../common/R/gammaf_r.c \
636 ../common/R/lgammaf.c \
637 ../common/R/lgammaf_r.c \
638 ../common/R/loglpf.c \
639 ../common/R/logbf.c \
640 ../common/R/logf.c \
641 ../common/R/scalbf.c \
642 ../common/R/signgamf.c \
643 ../common/R/significandf.c \
644 ../common/R/sinf.c \
645 ../common/R/sinhf.c \
646 ../common/R/sincosf.c \
647 ../common/R/sincospif.c \
648 ../common/R/sqrtf.c \
649 ../common/R/tanf.c \
650 ../common/R/tanhf.c

652 SRCS_Q = \

```

```

653 ../common/Q/_TBL_atanl.c \
654 ../common/Q/_TBL_expl.c \
655 ../common/Q/_TBL_expm1l.c \
656 ../common/Q/_TBL_logl.c \
657 ../common/Q/finitel.c \
658 ../common/Q/isnanl.c \
659 ../common/Q/_cosl.c \
660 ../common/Q/_lgammal.c \
661 ../common/Q/_poly_libmq.c \
662 ../common/Q/_rem_pio2l.c \
663 ../common/Q/_sincosl.c \
664 ../common/Q/_sinl.c \
665 ../common/Q/_tanl.c \
666 ../common/Q/_TBL_cosl.c \
667 ../common/Q/_TBL_ipio2l.c \
668 ../common/Q/_TBL_sinl.c \
669 ../common/Q/_TBL_tanl.c \
670 ../common/Q/acoshl.c \
671 ../common/Q/acosl.c \
672 ../common/Q/asinh1.c \
673 ../common/Q/asinl.c \
674 ../common/Q/atan2l.c \
675 ../common/Q/atan2pil.c \
676 ../common/Q/atanhl.c \
677 ../common/Q/atanl.c \
678 ../common/Q/cbrtl.c \
679 ../common/Q/copysignl.c \
680 ../common/Q/coshl.c \
681 ../common/Q/cosl.c \
682 ../common/Q/erfl.c \
683 ../common/Q/exp10l.c \
684 ../common/Q/exp2l.c \
685 ../common/Q/expl.c \
686 ../common/Q/expm1l.c \
687 ../common/Q/fabs1.c \
688 ../common/Q/floorl.c \
689 ../common/Q/fmodl.c \
690 ../common/Q/gammal.c \
691 ../common/Q/gammal_r.c \
692 ../common/Q/hypotl.c \
693 ../common/Q/ieee_funcl.c \
694 ../common/Q/ilogbl.c \
695 ../common/Q/j0l.c \
696 ../common/Q/j1l.c \
697 ../common/Q/jnl.c \
698 ../common/Q/lgammal.c \
699 ../common/Q/lgammal_r.c \
700 ../common/Q/log10l.c \
701 ../common/Q/log1pl.c \
702 ../common/Q/log2l.c \
703 ../common/Q/logbl.c \
704 ../common/Q/logl.c \
705 ../common/Q/nextafterl.c \
706 ../common/Q/powl.c \
707 ../common/Q/remainderl.c \
708 ../common/Q/rintl.c \
709 ../common/Q/rndintl.c \
710 ../common/Q/scalbl.c \
711 ../common/Q/scalbnl.c \
712 ../common/Q/signgaml.c \
713 ../common/Q/significandl.c \
714 ../common/Q/sincosl.c \
715 ../common/Q/sincospil.c \
716 ../common/Q/sinh1.c \
717 ../common/Q/sinl.c \
718 ../common/Q/sinpil.c \

```

```

719 ../common/Q/sqrtl.c \
720 ../common/Q/tanhl.c \
721 ../common/Q/tanl.c

723 SRCS_Q_sparc = \
724     $(SRCS_Q)

726 SRCS_complex = \
727     ../common/complex/cabs.c \
728     ../common/complex/cabsf.c \
729     ../common/complex/cabs1.c \
730     ../common/complex/cacos.c \
731     ../common/complex/cacosf.c \
732     ../common/complex/cacosh.c \
733     ../common/complex/cacoshf.c \
734     ../common/complex/cacosh1.c \
735     ../common/complex/cacos1.c \
736     ../common/complex/carg.c \
737     ../common/complex/cargf.c \
738     ../common/complex/carg1.c \
739     ../common/complex/casin.c \
740     ../common/complex/casinf.c \
741     ../common/complex/casinh.c \
742     ../common/complex/casinhf.c \
743     ../common/complex/casinh1.c \
744     ../common/complex/casin1.c \
745     ../common/complex/catan.c \
746     ../common/complex/catanf.c \
747     ../common/complex/catanh.c \
748     ../common/complex/catanhf.c \
749     ../common/complex/catanh1.c \
750     ../common/complex/catan1.c \
751     ../common/complex/ccos.c \
752     ../common/complex/ccosf.c \
753     ../common/complex/ccosh.c \
754     ../common/complex/ccoshf.c \
755     ../common/complex/ccosh1.c \
756     ../common/complex/ccosl.c \
757     ../common/complex/cexp.c \
758     ../common/complex/cexpf.c \
759     ../common/complex/cexpl.c \
760     ../common/complex/cimag.c \
761     ../common/complex/cimagf.c \
762     ../common/complex/cimag1.c \
763     ../common/complex/clog.c \
764     ../common/complex/clogf.c \
765     ../common/complex/clog1.c \
766     ../common/complex/conj.c \
767     ../common/complex/conjf.c \
768     ../common/complex/conj1.c \
769     ../common/complex/cpow.c \
770     ../common/complex/cpowf.c \
771     ../common/complex/cpowl.c \
772     ../common/complex/cproj.c \
773     ../common/complex/cprojf.c \
774     ../common/complex/cproj1.c \
775     ../common/complex/creal.c \
776     ../common/complex/crealf.c \
777     ../common/complex/creall.c \
778     ../common/complex/csin.c \
779     ../common/complex/csinf.c \
780     ../common/complex/csinh.c \
781     ../common/complex/csinhf.c \
782     ../common/complex/csinh1.c \
783     ../common/complex/csin1.c \
784     ../common/complex/csqr1.c \

```

```

785 ../common/complex/csqrtf.c \
786 ../common/complex/csqrtl.c \
787 ../common/complex/ctan.c \
788 ../common/complex/ctanf.c \
789 ../common/complex/ctanh.c \
790 ../common/complex/ctanhf.c \
791 ../common/complex/ctanh1.c \
792 ../common/complex/ctanl.c \
793 ../common/complex/k_atan2.c \
794 ../common/complex/k_atan2l.c \
795 ../common/complex/k_cexp.c \
796 ../common/complex/k_cexpl.c \
797 ../common/complex/k_clog_r.c \
798 ../common/complex/k_clog_rl.c \

800 SRCS_m9x_i386 = \
801 ../common/m9x/___fex_sse.c \
802 ../common/m9x/feprec.c \
803 ../common/m9x/___fex_i386.c \

805 SRCS_m9x_i386_i386 = \
806 ../common/m9x/lroundf.c \

808 SRCS_m9x_i386_amd64 = \
809 ../common/m9x/llrint.c \
810 ../common/m9x/llrintf.c \
811 ../common/m9x/llrintl.c \
812 ../common/m9x/nexttowardl.c \
813 ../common/m9x/remquo.c \
814 ../common/m9x/remquof.c \
815 ../common/m9x/round.c \
816 ../common/m9x/roundl.c \
817 ../common/m9x/scalbln.c \
818 ../common/m9x/scalblnf.c \
819 ../common/m9x/scalblnl.c \
820 ../common/m9x/trunc.c \
821 ../common/m9x/truncl.c \

823 # sparc
824 SRCS_m9x_sparc_sparc = \
825 ../common/m9x/lrint.c \
826 ../common/m9x/lrintf.c \
827 ../common/m9x/lrintl.c \
828 ../common/m9x/lround.c \
829 ../common/m9x/lroundf.c \
830 ../common/m9x/lroundl.c \

832 SRCS_m9x_sparc = \
833 ../common/m9x/___fex_sparc.c \
834 ../common/m9x/llrint.c \
835 ../common/m9x/llrintf.c \
836 ../common/m9x/llrintl.c \
837 ../common/m9x/nexttowardl.c \
838 ../common/m9x/remquo.c \
839 ../common/m9x/remquof.c \
840 ../common/m9x/remquol.c \
841 ../common/m9x/round.c \
842 ../common/m9x/roundl.c \
843 ../common/m9x/scalbln.c \
844 ../common/m9x/scalblnf.c \
845 ../common/m9x/scalblnl.c \
846 ../common/m9x/trunc.c \
847 ../common/m9x/truncl.c \

849 SRCS_m9x = \
850 ${SRCS_m9x_${MACH}} \

```

```

851 ${SRCS_m9x_sparc_${TARGET_ARCH}} \
852 ${SRCS_m9x_i386_${TARGET_ARCH}} \
853 ../common/m9x/___fex_hdr.c \
854 ../common/m9x/___fex_sym.c \
855 ../common/m9x/fdim.c \
856 ../common/m9x/fdimf.c \
857 ../common/m9x/fdiml.c \
858 ../common/m9x/feexcept.c \
859 ../common/m9x/fenv.c \
860 ../common/m9x/feround.c \
861 ../common/m9x/fex_handler.c \
862 ../common/m9x/fex_log.c \
863 ../common/m9x/fma.c \
864 ../common/m9x/fmaf.c \
865 ../common/m9x/fmal.c \
866 ../common/m9x/fmax.c \
867 ../common/m9x/fmaxf.c \
868 ../common/m9x/fmaxl.c \
869 ../common/m9x/fmin.c \
870 ../common/m9x/fminf.c \
871 ../common/m9x/fminl.c \
872 ../common/m9x/frexp.c \
873 ../common/m9x/frexp.c \
874 ../common/m9x/frexp.c \
875 ../common/m9x/ldexp.c \
876 ../common/m9x/ldexpf.c \
877 ../common/m9x/ldexpl.c \
878 ../common/m9x/llround.c \
879 ../common/m9x/llroundf.c \
880 ../common/m9x/llroundl.c \
881 ../common/m9x/modf.c \
882 ../common/m9x/modff.c \
883 ../common/m9x/modfl.c \
884 ../common/m9x/nan.c \
885 ../common/m9x/nanf.c \
886 ../common/m9x/nanl.c \
887 ../common/m9x/nearbyint.c \
888 ../common/m9x/nearbyintf.c \
889 ../common/m9x/nearbyintl.c \
890 ../common/m9x/nexttoward.c \
891 ../common/m9x/nexttowardf.c \
892 ../common/m9x/roundf.c \
893 ../common/m9x/tgamma.c \
894 ../common/m9x/tgammaf.c \
895 ../common/m9x/tgamma.c \
896 ../common/m9x/truncf.c \

898 SRCS_C_sparc = \
899 ../common/C/___tan.c \
900 ../common/C/___TBL_atan.c \
901 ../common/C/___TBL_exp2.c \
902 ../common/C/___TBL_log.c \
903 ../common/C/___TBL_log2.c \
904 ../common/C/___TBL_tan.c \
905 ../common/C/acos.c \
906 ../common/C/asin.c \
907 ../common/C/atan.c \
908 ../common/C/atan2.c \
909 ../common/C/ceil.c \
910 ../common/C/cos.c \
911 ../common/C/exp.c \
912 ../common/C/exp10.c \
913 ../common/C/exp2.c \
914 ../common/C/expml.c \
915 ../common/C/floor.c \
916 ../common/C/fmod.c \

```

```

917 ../common/C/hypot.c \
918 ../common/C/ilogb.c \
919 ../common/C/isnan.c \
920 ../common/C/log.c \
921 ../common/C/log10.c \
922 ../common/C/log2.c \
923 ../common/C/pow.c \
924 ../common/C/remainder.c \
925 ../common/C/rint.c \
926 ../common/C/scalbn.c \
927 ../common/C/sin.c \
928 ../common/C/sincos.c \
929 ../common/C/tan.c

931 SRCS_i386_i386 = \
932 ../common/C/_libx_errno.c

934 SRCS_sparc_sparc = \
935 $(SRCS_i386_i386)

937 SRCS_sparc_sparcv9 = \
938 ../common/C/copysign.c \
939 ../common/C/fabs.c \
940 ../common/C/nextafter.c

942 SRCS_i386_amd64 = \
943 ../common/C/_TBL_atan.c \
944 ../common/C/_TBL_exp2.c \
945 ../common/C/_TBL_log.c \
946 ../common/C/_TBL_log2.c \
947 ../common/C/_tan.c \
948 ../common/C/_TBL_tan.c \
949 ../common/C/copysign.c \
950 ../common/C/exp.c \
951 ../common/C/fabs.c \
952 ../common/C/ilogb.c \
953 ../common/C/isnan.c \
954 ../common/C/nextafter.c \
955 ../common/C/rint.c \
956 ../common/C/scalbn.c \
957 ../common/C/acos.c \
958 ../common/C/asin.c \
959 ../common/C/atan.c \
960 ../common/C/atan2.c \
961 ../common/C/ceil.c \
962 ../common/C/cos.c \
963 ../common/C/exp10.c \
964 ../common/C/exp2.c \
965 ../common/C/expm1.c \
966 ../common/C/floor.c \
967 ../common/C/hypot.c \
968 ../common/C/log.c \
969 ../common/C/log10.c \
970 ../common/C/log2.c \
971 ../common/C/pow.c \
972 ../common/C/sin.c \
973 ../common/C/sincos.c \
974 ../common/C/tan.c

976 SRCS_C = \
977 $(SRCS_C_$(MACH)) \
978 $(SRCS_C_i386_$(TARGET_ARCH)) \
979 ../common/C/_cos.c \
980 ../common/C/_lgamma.c \
981 ../common/C/_rem_pio2.c \
982 ../common/C/_rem_pio2m.c \

```

```

983 ../common/C/_sin.c \
984 ../common/C/_sincos.c \
985 ../common/C/_xpg6.c \
986 ../common/C/_lib_version.c \
987 ../common/C/_SVID_error.c \
988 ../common/C/_TBL_ipio2.c \
989 ../common/C/_TBL_sin.c \
990 ../common/C/acosh.c \
991 ../common/C/asinh.c \
992 ../common/C/atan2pi.c \
993 ../common/C/atanh.c \
994 ../common/C/cbrt.c \
995 ../common/C/cosh.c \
996 ../common/C/erf.c \
997 ../common/C/gamma.c \
998 ../common/C/gamma_r.c \
999 ../common/C/j0.c \
1000 ../common/C/j1.c \
1001 ../common/C/jn.c \
1002 ../common/C/lgamma.c \
1003 ../common/C/lgamma_r.c \
1004 ../common/C/loglp.c \
1005 ../common/C/logb.c \
1006 ../common/C/matherr.c \
1007 ../common/C/scalb.c \
1008 ../common/C/singam.c \
1009 ../common/C/significand.c \
1010 ../common/C/sincospi.c \
1011 ../common/C/sinh.c \
1012 ../common/C/sqrt.c \
1013 ../common/C/tanh.c

1015 SRCS = \
1016 $(SRCS_Q_$(MACH)) \
1017 $(SRCS_LD_$(MACH)) \
1018 $(SRCS_R) \
1019 $(SRCS_complex) \
1020 $(SRCS_C)

1022 .KEEP_STATE:

1024 all: $(LIBS)

1026 lint: lintcheck

```



```

*****
2668 Mon Oct 27 12:36:38 2014
new/usr/src/lib/libm/Makefile.libm.com
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
14 #
15 #
16 LIBMDIR      = $(SRC)/lib/libm
17 #
18 LIBMSRC      = $(LIBMDIR)/common
19 #
20 CPP_CMD      = $(CC) -E -Xs
21 #
22 ASSUFFIX_sparc = S
23 ASSUFFIX_i386  = s
24 ASSUFFIX      = $(ASSUFFIX_$(MACH))
25 #
26 # C99MODE of neither enabled nor disabled is "no_lib", whereby we expect
27 # C99-the-language, but don't modify the behaviour of library routines. This
28 # is VERY IMPORTANT, as -xc99=%all, for instance, would link us with
29 # values-xpg6, which would introduce an _xpg6 to our object with the C99
30 # flags set, causing us to default C99 libm behaviour on, breaking
31 # compatibility.
32 C99MODE      =
33 #
34 M4FLAGS      = -D__STDC__ -DPIC
35 M4FLAGS      = -D__STDC__ -DELFOBJ -DPIC
36 #
37 LDBLDIR_sparc = Q
38 LDBLDIR_i386  = LD
39 LDBLDIR      = $(LDBLDIR_$(MACH))
40 #
41 LM_IL        = $(LIBMDIR)/$(TARGET_ARCH)/src/locallibm.il
42 #
43 CFLAGS       += $(C_PICFLAGS) $(XSTRCONST) $(LM_IL)
44 CFLAGS64     += $(C_PICFLAGS) $(XSTRCONST) $(LM_IL)
45 #
46 CPPFLAGS     += -I$(LIBMSRC)/C \
47               += -DELFOBJ \
48               -DLIBM_MT_FEX_SYNC \
49               -I$(LIBMSRC)/C \
50               -I$(LIBMSRC)/$(LDBLDIR) -I$(LIBMDIR)/$(TARGET_ARCH)/src
51 #
52 # GCC needs __C99FEATURES__ such that the implementations of isunordered,
53 # isgreater, islessequal, etc, exist. This is basically equivalent to
54 # providing no -xc99 to Studio, in that it gets us the C99 language features,
55 # but not values-xpg6, the reason for which is outline with C99MODE.
56 CFLAGS       += -_gcc=-D__C99FEATURES__
57 CFLAGS64     += -_gcc=-D__C99FEATURES__

```

```

58 # libm depends on integer overflow characteristics
59 CFLAGS       += -_gcc=-fno-strict-overflow
60 CFLAGS64     += -_gcc=-fno-strict-overflow
61 #
62 $(DYNLIB)    := LDLIBS += -lc
63 #
64 $(LINTLIB)   := SRCS = $(LIBMSRC)/$(LINTSRC)
65 #
66 CLEANFILES  += pics/*.s pics/*.S
67 #
68 FPDEF_amd64 = -DARCH_amd64
69 FPDEF_sparc  = -DCG89 -DARCH_v8plus -DFPADD_TRAPS_INCOMPLETE_ON_NAN
70 FPDEF_sparcv9 = -DARCH_v9 -DFPADD_TRAPS_INCOMPLETE_ON_NAN
71 FPDEF        = $(FPDEF_$(TARGET_ARCH))
72 #
73 ASFLAGS     = -P -D_ASM $(FPDEF)
74 #
75 XARCH_sparc = v8plus
76 XARCH_sparcv9 = v9
77 XARCH_i386   = f80387
78 XARCH_amd64  = amd64
79 XARCH        = $(XARCH_$(TARGET_ARCH))
80 #
81 ASOPT_sparc  = -xarch=$(XARCH) $(AS_PICFLAGS)
82 ASOPT_sparcv9 = -xarch=$(XARCH) $(AS_PICFLAGS)
83 ASOPT_i386   =
84 ASOPT_amd64  = -xarch=$(XARCH) $(AS_PICFLAGS)
85 ASOPT        = $(ASOPT_$(TARGET_ARCH))
86 #
87 ASFLAGS     += $(ASOPT)
88 #
89 CPPFLAGS_sparc = -DFPADD_TRAPS_INCOMPLETE_ON_NAN \
90                 -DFDTOS_TRAPS_INCOMPLETE_IN_FNS_MODE
91 CPPFLAGS     += $(CPPFLAGS_$(MACH))
92 ASFLAGS     += $(CPPFLAGS)

```

new/usr/src/lib/libm/common/C/copysign.c

1

```
*****
1200 Mon Oct 27 12:36:39 2014
new/usr/src/lib/libm/common/C/copysign.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak copysign = __copysign
32 #endif

32 #include "libm.h"

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

38     hx = ((int *) &x)[HIWORD];
39     hy = ((int *) &y)[HIWORD];
40     return (hx ^ hy) >= 0 ? (x) : (-x);
41 }
_____unchanged_portion_omitted_____
```

```

*****
5483 Mon Oct 27 12:36:39 2014
new/usr/src/lib/libm/common/C/hypot.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak hypot = __hypot
32 #endif

32 /* INDENT OFF */
33 /*
34  * Hypot(x, y)
35  * by K.C. Ng for SUN 4.0 libm, updated 3/11/2003.
36  * Method :
37  * A. When rounding is rounded-to-nearest:
38  *   If  $z = x^2 + y^2$  has error less than  $\sqrt{2} / 2$  ulp than
39  *    $\sqrt{z}$  has error less than 1 ulp.
40  *   So, compute  $\sqrt{x^2+y^2}$  with some care as follows:
41  *   Assume  $x > y > 0$ ;
42  *   1. Check whether save and set rounding to round-to-nearest
43  *   2. if  $x > 2y$  use
44  *        $xh*xh+(y*y+((x-xh)*(x+xh)))$  for  $x^2+y^2$ 
45  *   where  $xh = x$  with lower 32 bits cleared; else
46  *   3. if  $x \leq 2y$  use
47  *        $x2h*yh+((x-y)*(x-y)+(x2h*(y-yh)+(x2-x2h)*y))$ 
48  *   where  $x2 = 2*x$ ,  $x2h = 2x$  with lower 32 bits cleared,  $yh = y$  with
49  *   lower 32 bits chopped.
50  *
51  * B. When rounding is not rounded-to-nearest:
52  *   The following (magic) formula will yield an error less than 1 ulp.
53  *    $z = \sqrt{x^2 + y^2}$ 
54  *    $\text{hypot}(x, y) = x + (y / ((x + z) / y))$ 
55  *
56  * NOTE: DO NOT remove parenthesis!
57  *
58  * Special cases:

```

```

59  *   hypot(x, y) is INF if x or y is +INF or -INF; else
60  *   hypot(x, y) is NAN if x or y is NAN.
61  *
62  * Accuracy:
63  *   hypot(x, y) returns  $\sqrt{x^2+y^2}$  with error less than 1 ulps
64  *   (units in the last place)
65  */

67 #include "libm.h"

69 static const double
70     zero = 0.0,
71     oneplu = 1.00000000000000022204e+00, /* 0x3ff00000 1 = 1+2**-52 */
72     twom53 = 1.11022302462515654042e-16, /* 0x3ca00000 0 = 2**-53 */
73     twom768 = 6.441148769597133308e-232, /* 2-768 */
74     two768 = 1.552518092300708935e+231; /* 2768 */

76 /* INDENT ON */

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

84     ix = ((int *) &x)[HIWORD] & ~0x80000000;
85     lx = ((int *) &x)[LOWORD];
86     iy = ((int *) &y)[HIWORD] & ~0x80000000;
87     ly = ((int *) &y)[LOWORD];

88 /*
89  * Force ax = |x| ~>~ ay = |y|
90  */
91     if (iy > ix) {
92         ax = fabs(y);
93         ay = fabs(x);
94         i = ix;
95         ix = iy;
96         iy = i;
97         lx = ly;
98         ly = lx;
99     } else {
100         ax = fabs(x);
101         ay = fabs(y);
102     }
103     nx = ix >> 20;
104     ny = iy >> 20;
105     j = nx - ny;
106 /*
107  * x >= 2500 (x*x or y*y may overflow)
108  */
109     if (nx >= 0x5f3) {
110         if (nx == 0x7ff) { /* inf or NaN, signal of sNaN */
111             if (((ix - 0x7ff00000) | lx) == 0)
112                 return (ax == ay ? ay : ax);
113             else if (((iy - 0x7ff00000) | ly) == 0)
114                 return (ax == ax ? ax : ay);
115             else
116                 return (ax * ay); /* + -> * for Cheetah */
117         } else if (j > 32) { /* x >> y */
118             if (j <= 53)
119                 ay *= twom53;
120             ax += ay;
121             if (((int *) &ax)[HIWORD] == 0x7ff00000)
122                 ax = _SVID_libm_err(x, y, 4);
123             return (ax);
124         }

```

```

125     }
126     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 oneplu*oneplu with oneplu+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 (oneplu * oneplu != oneplu + 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 Mon Oct 27 12:36:39 2014
new/usr/src/lib/libm/common/C/ilogb.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak ilogb = __ilogb
32 #endif

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

35 #if defined(__x86)
37 #if defined(USE_FPSCALE) || defined(__x86)
36 static const double two52 = 4503599627370496.0;
37 #else
38 /*
39  * v: high part of a non-zero subnormal |x|; w: low part of |x|
40  */
41 static int
42 ilogb_subnormal(unsigned v, unsigned w) {
43     int r = -1022 - 52;

44     if (v)
45         r += 32;
46     else
47         v = w;
48     if (v & 0xffff0000)
49         r += 16, v >>= 16;
50     if (v & 0xf00)
51         r += 8, v >>= 8;
52     if (v & 0xf0)
53         r += 4, v >>= 4;
54     v <<= 1;
55     return (r + ((0xffffaa50 >> v) & 0x3));
56 }
57 }

```

```

58 #endif /* defined(__x86) */
60 #endif /* defined(USE_FPSCALE) */

60 static int
61 raise_invalid(int v) { /* SUSv3 requires ilogb(0,+/-Inf,NaN) raise invalid */
62 #ifndef lint
63     if ((__xpg6 & _C99SUSv3_ilogb_0InfNaN_raises_invalid) != 0) {
64         static const double zero = 0.0;
65         volatile double dummy;

66         dummy = zero / zero;

67     }
68 #endif
69 return (v);
71 }

73 int
74 ilogb(double x) {
75     int *px = (int *) &x, k = px[HIWORD] & ~0x80000000;

76     if (k < 0x00100000) {
77         if ((px[LOWORD] | k) == 0)
78             return (raise_invalid(0x80000001));
79         else {
80             #if defined(__x86)
81             #if defined(USE_FPSCALE) || defined(__x86)
82                 x *= two52;
83                 return ((px[HIWORD] & 0x7ff00000) >> 20) - 1075;
84             #else
85                 return (ilogb_subnormal(k, px[LOWORD]));
86             #endif
87         }
88     } else if (k < 0x7ff00000)
89         return ((k >> 20) - 1023);
90     else
91         return (raise_invalid(0x7fffffff));
92 }

```

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

```

*****
5457 Mon Oct 27 12:36:40 2014
new/usr/src/lib/libm/common/C/libm_protos.h
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #ifndef _C_LIBM_PROTOS_H
30 #define _C_LIBM_PROTOS_H

32 /*
33 * Many symbols used to be namespaced with __libm to prevent collisions. All
34 * but these two were otherwise scoped local and directly bound, so that
35 * collision could not occur.
36 *
37 * For reasons unknown, these two are global (but private).
38 */
39 #ifndef LIBM_OPT_BUILD
40 #define TBL_cos __libmopt_TBL_cos
41 #define TBL_exp2_512 __libmopt_TBL_exp2_512
42 #define TBL_ipio2_inf __libmopt_TBL_ipio2_inf
43 #define TBL_jlog_n1 __libmopt_TBL_jlog_n1
44 #define TBL_jlog_n2 __libmopt_TBL_jlog_n2
45 #define TBL_jlog_p1 __libmopt_TBL_jlog_p1
46 #define TBL_jlog_p2 __libmopt_TBL_jlog_p2
47 #define TBL_log10 __libmopt_TBL_log10
48 #define TBL_log2_14 __libmopt_TBL_log2_14
49 #define TBL_log2_9 __libmopt_TBL_log2_9
50 #define TBL_sin __libmopt_TBL_sin
51 #define TBL_sincos __libmopt_TBL_sincos
52 #define TBL_xexp __libmopt_TBL_xexp
53 #define TBL_xlog __libmopt_TBL_xlog
54 #define k_cos __libmopt_k_cos
55 #define k_sin __libmopt_k_sin
56 #define k_sincos __libmopt_k_sincos
57 #define reduction __libmopt_reduction
58 #define rem_pio2 __libmopt_rem_pio2
59 #define rem_pio2m __libmopt_rem_pio2m
60 #else /* defined(LIBM_OPT_BUILD) */

```

```

54 #ifndef LIBM_BUILD
55 #define _SVID_libm_err __libm_SVID_libm_err /* not used by -lsunmath */
56 #define TBL_atan __libm_TBL_atan
57 #define TBL_atanl __libm_TBL_atanl
58 #define TBL_atan_hi __libm_TBL_atan_hi /* not used by -lsunmath */
59 #define TBL_atan_lo __libm_TBL_atan_lo /* not used by -lsunmath */
60 #define TBL_exp2_hi __libm_TBL_exp2_hi /* not used by -lsunmath */
61 #define TBL_exp2_lo __libm_TBL_exp2_lo /* not used by -lsunmath */
62 #define TBL_ipio2_inf __libm_TBL_ipio2_inf
63 #define TBL_log __libm_TBL_log
64 #define TBL_log2_hi __libm_TBL_log2_hi /* not used by -lsunmath */
65 #define TBL_log2_lo __libm_TBL_log2_lo /* not used by -lsunmath */
66 #define TBL_log_hi __libm_TBL_log_hi /* not used by -lsunmath */
67 #define TBL_log_lo __libm_TBL_log_lo /* not used by -lsunmath */
68 #define TBL_sincos __libm_TBL_sincos
69 #define TBL_sincosx __libm_TBL_sincosx
70 #define TBL_tan_hi __libm_TBL_tan_hi /* not used by -lsunmath */
71 #define TBL_tan_lo __libm_TBL_tan_lo /* not used by -lsunmath */
72 #define k_cexp __libm_k_cexp /* C99 libm */
73 #define k_cexpl __libm_k_cexpl /* C99 libm */
74 #define k_clog_r __libm_k_clog_r /* C99 libm */
75 #define k_clog_rl __libm_k_clog_rl /* C99 libm */
76 #define k_atan2 __libm_k_atan2 /* C99 libm */
77 #define k_atan2l __libm_k_atan2l /* C99 libm */
78 #define k_cos __libm_k_cos
79 #define k_lgamma __libm_k_lgamma
80 #define k_sin __libm_k_sin
81 #define k_sincos __libm_k_sincos
82 #define k_tan __libm_k_tan
83 #define reduction __libm_reduction /* i386 only */
39 #define rem_pio2 __libm_rem_pio2
40 #define rem_pio2m __libm_rem_pio2m
86 #define k_cosf __libm_k_cosf /* C99 libm */
87 #define k_cosl __libm_k_cosl /* C99 libm */
88 #define k_lgamma __libm_k_lgamma /* C99 libm */
89 #define k_sincosf __libm_k_sincosf /* C99 libm */
90 #define k_sincosl __libm_k_sincosl /* C99 libm */
91 #define k_sinf __libm_k_sinf /* C99 libm */
92 #define k_sinl __libm_k_sinl /* C99 libm */
93 #define k_tanf __libm_k_tanf /* C99 libm */
94 #define k_tanl __libm_k_tanl /* C99 libm */
95 #define poly_libmq __libm_poly_libmq /* C99 libm */
96 #define rem_pio2l __libm_rem_pio2l /* C99 libm */
97 #define TBL_atanl_hi __libm_TBL_atanl_hi /* C99 libm */
98 #define TBL_atanl_lo __libm_TBL_atanl_lo /* C99 libm */
99 #define TBL_cosl_hi __libm_TBL_cosl_hi /* C99 libm */
100 #define TBL_cosl_lo __libm_TBL_cosl_lo /* C99 libm */
101 #define TBL_expl_hi __libm_TBL_expl_hi /* C99 libm */
102 #define TBL_expl_lo __libm_TBL_expl_lo /* C99 libm */
103 #define TBL_expml1 __libm_TBL_expml1 /* C99 libm */
104 #define TBL_expml1x __libm_TBL_expml1x /* C99 libm */
105 #define TBL_ipio2l_inf __libm_TBL_ipio2l_inf /* C99 libm */
106 #define TBL_logl_hi __libm_TBL_logl_hi /* C99 libm */
107 #define TBL_logl_lo __libm_TBL_logl_lo /* C99 libm */
108 #define TBL_r_atan_hi __libm_TBL_r_atan_hi /* C99 libm */
109 #define TBL_r_atan_lo __libm_TBL_r_atan_lo /* C99 libm */
110 #define TBL_sinl_hi __libm_TBL_sinl_hi /* C99 libm */
111 #define TBL_sinl_lo __libm_TBL_sinl_lo /* C99 libm */
112 #define TBL_tanl_hi __libm_TBL_tanl_hi /* C99 libm */
113 #define TBL_tanl_lo __libm_TBL_tanl_lo /* C99 libm */
114 #endif /* defined(LIBM_BUILD) */
115 #endif /* defined(LIBM_OPT_BUILD) */

42 #ifndef _ASM
43 #ifdef __STDC__
44 #define __P(p) p

```

```

45 #else
46 #define __P(p) ()
47 #endif

49 #include <sys/ieeeefp.h>

51 extern double _SVID_libm_err __P((double, double, int));
52 extern double __k_cos __P((double, double));
53 extern double __k_cos_ __P((double *));
54 extern double __k_lgamma __P((double, int *));
55 extern double __k_sin __P((double, double));
56 extern double __k_sin_ __P((double *));
57 extern double __k_sincos __P((double, double, double *));
58 extern double __k_sincos_ __P((double *, double *));
59 extern double __k_tan __P((double, double, int));
60 extern double __k_cexp __P((double, int *));
61 extern long double __k_cexpl __P((long double, int *));
62 extern double __k_clog_r __P((double, double, double *));
63 extern long double __k_clog_rl __P((long double, long double, long double *));
64 extern double __k_atan2 __P((double, double, double *));
65 extern long double __k_atan2l __P((long double, long double, long double *));
66 extern int __rem_pio2 __P((double, double *));
67 extern int __rem_pio2m __P((double *, double *, int, int, int, const int *));

69 /*
70  * entry points that are in-lined
71  */
72 extern double copysign __P((double, double));
73 extern int finite __P((double));
74 extern enum fp_class_type fp_class __P((double));
75 extern double infinity __P((void));
76 extern int isinf __P((double));
77 extern int signbit __P((double));

79 /*
80  * new C99 entry points
81  */
82 extern double fdim __P((double, double));
83 extern double fma __P((double, double, double));
84 extern double fmax __P((double, double));
85 extern double fmin __P((double, double));
86 extern double frexp __P((double, int *));
87 extern double ldexp __P((double, int));
88 extern double modf __P((double, double *));
89 extern double nan __P((const char *));
90 extern double nearbyint __P((double));
91 extern double nexttoward __P((double, long double));
92 extern double remquo __P((double, double, int *));
93 extern double round __P((double));
94 extern double scalbln __P((double, long int));
95 extern double tgamma __P((double));
96 extern double trunc __P((double));
97 extern float fdimf __P((float, float));
98 extern float fmaf __P((float, float, float));
99 extern float fmaxf __P((float, float));
100 extern float fminf __P((float, float));
101 extern float frexpf __P((float, int *));
102 extern float ldexpf __P((float, int));
103 extern float modff __P((float, float *));
104 extern float nanf __P((const char *));
105 extern float nearbyintf __P((float));
106 extern float nextafterf __P((float, float));
107 extern float nexttowardf __P((float, long double));
108 extern float remquof __P((float, float, int *));
109 extern float roundf __P((float));
110 extern float scalblnf __P((float, long int));

```

```

111 extern float tgammaf __P((float));
112 extern float truncf __P((float));
113 extern long double frexpl(long double, int *);
114 extern long double fdiml __P((long double, long double));
115 extern long double fmal __P((long double, long double, long double));
116 extern long double fmaxl __P((long double, long double));
117 extern long double fminl __P((long double, long double));
118 extern long double ldexpl __P((long double, int));
119 extern long double modfl __P((long double, long double *));
120 extern long double nanl __P((const char *));
121 extern long double nearbyintl __P((long double));
122 extern long double nextafterl __P((long double, long double));
123 extern long double nexttowardl __P((long double, long double));
124 extern long double remquol __P((long double, long double, int *));
125 extern long double roundl __P((long double));
126 extern long double scalblnl __P((long double, long int));
127 extern long double tgamma __P((long double));
128 extern long double trunc __P((long double));
129 extern long int lrint __P((double));
130 extern long int lrintf __P((float));
131 extern long int lrintl __P((long double));
132 extern long int lround __P((double));
133 extern long int lroundf __P((float));
134 extern long int lroundl __P((long double));
135 extern long long int llrint __P((double));
136 extern long long int llrintf __P((float));
137 extern long long int llrintl __P((long double));
138 extern long long int llround __P((double));
139 extern long long int llroundf __P((float));
140 extern long long int llroundl __P((long double));
141 #endif /* _ASM */

143 #endif /* _C_LIBM_PROTOS_H */

```

```

*****
23508 Mon Oct 27 12:36:40 2014
new/usr/src/lib/libm/common/C/libm_synonyms.h
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #ifndef _LIBM_SYNONYMS_H
31 #define _LIBM_SYNONYMS_H

33 #if !defined(lint)
33 #if defined(ELFOBJ) && !defined(lint)

35 #define cabs          __cabs          /* C99 <complex.h> */
36 #define cabsf        __cabsf        /* C99 <complex.h> */
37 #define cabsl        __cabsl        /* C99 <complex.h> */
38 #define cacos        __cacos        /* C99 <complex.h> */
39 #define cacosf       __cacosf       /* C99 <complex.h> */
40 #define cacosl       __cacosl       /* C99 <complex.h> */
41 #define cacosh       __cacosh       /* C99 <complex.h> */
42 #define cacoshf     __cacoshf     /* C99 <complex.h> */
43 #define cacoshl     __cacoshl     /* C99 <complex.h> */
44 #define carg         __carg         /* C99 <complex.h> */
45 #define cargf       __cargf       /* C99 <complex.h> */
46 #define cargl       __cargl       /* C99 <complex.h> */
47 #define casin       __casin       /* C99 <complex.h> */
48 #define casinf      __casinf      /* C99 <complex.h> */
49 #define casinl     __casinl     /* C99 <complex.h> */
50 #define casinh      __casinh      /* C99 <complex.h> */
51 #define casinhf    __casinhf    /* C99 <complex.h> */
52 #define casinhl    __casinhl    /* C99 <complex.h> */
53 #define catan       __catan       /* C99 <complex.h> */
54 #define catanf     __catanf     /* C99 <complex.h> */
55 #define catanl     __catanl     /* C99 <complex.h> */
56 #define catanh     __catanh     /* C99 <complex.h> */
57 #define catanhf   __catanhf   /* C99 <complex.h> */
58 #define catanhl   __catanhl   /* C99 <complex.h> */
59 #define ccos       __ccos       /* C99 <complex.h> */

```

```

60 #define ccosf      __ccosf      /* C99 <complex.h> */
61 #define ccosl      __ccosl      /* C99 <complex.h> */
62 #define ccosh      __ccosh      /* C99 <complex.h> */
63 #define ccoshf     __ccoshf     /* C99 <complex.h> */
64 #define ccoshl     __ccoshl     /* C99 <complex.h> */
65 #define cexp       __cexp       /* C99 <complex.h> */
66 #define cexpf      __cexpf      /* C99 <complex.h> */
67 #define cexpl      __cexpl      /* C99 <complex.h> */
68 #define cimag      __cimag      /* C99 <complex.h> */
69 #define cimagf     __cimagf     /* C99 <complex.h> */
70 #define cimagl     __cimagl     /* C99 <complex.h> */
71 #define clog       __clog       /* C99 <complex.h> */
72 #define clogf      __clogf      /* C99 <complex.h> */
73 #define clogl      __clogl      /* C99 <complex.h> */
74 #define conj       __conj       /* C99 <complex.h> */
75 #define conjf      __conjf      /* C99 <complex.h> */
76 #define conjl      __conjl      /* C99 <complex.h> */
77 #define cpow       __cpow       /* C99 <complex.h> */
78 #define cpowf      __cpowf      /* C99 <complex.h> */
79 #define cpowl      __cpowl      /* C99 <complex.h> */
80 #define cproj      __cproj      /* C99 <complex.h> */
81 #define cprojf     __cprojf     /* C99 <complex.h> */
82 #define cprojl     __cprojl     /* C99 <complex.h> */
83 #define creal      __creal      /* C99 <complex.h> */
84 #define crealf     __crealf     /* C99 <complex.h> */
85 #define creall     __creall     /* C99 <complex.h> */
86 #define csin       __csin       /* C99 <complex.h> */
87 #define csinf      __csinf      /* C99 <complex.h> */
88 #define csinl      __csinl      /* C99 <complex.h> */
89 #define csinh      __csinh      /* C99 <complex.h> */
90 #define csinhf     __csinhf     /* C99 <complex.h> */
91 #define csinhl     __csinhl     /* C99 <complex.h> */
92 #define csqrt      __csqrt      /* C99 <complex.h> */
93 #define csqrtf     __csqrtf     /* C99 <complex.h> */
94 #define csqrtl     __csqrtl     /* C99 <complex.h> */
95 #define ctan       __ctan       /* C99 <complex.h> */
96 #define ctanf      __ctanf      /* C99 <complex.h> */
97 #define ctanl      __ctanl      /* C99 <complex.h> */
98 #define ctanh      __ctanh      /* C99 <complex.h> */
99 #define ctanhf     __ctanhf     /* C99 <complex.h> */
100 #define ctanhl     __ctanhl     /* C99 <complex.h> */
101 #define abrupt_underflow_ __abrupt_underflow_
102 #define acos       __acos       /* C99 <complex.h> */
103 #define acosd      __acosd      /* C99 <complex.h> */
104 #define acosdf     __acosdf     /* C99 <complex.h> */
105 #define acosdl     __acosdl     /* C99 <complex.h> */
106 #define acosf      __acosf      /* C99 <complex.h> */
107 #define acosh      __acosh      /* C99 <complex.h> */
108 #define acoshf     __acoshf     /* C99 <complex.h> */
109 #define acoshl     __acoshl     /* C99 <complex.h> */
110 #define acosl      __acosl      /* C99 <complex.h> */
111 #define acospl     __acospl     /* C99 <complex.h> */
112 #define acospf     __acospf     /* C99 <complex.h> */
113 #define acospi     __acospi     /* C99 <complex.h> */
114 #define acospif    __acospif    /* C99 <complex.h> */
115 #define acospil    __acospil    /* C99 <complex.h> */
116 #define acospl     __acospl     /* C99 <complex.h> */
117 #define aint       __aint       /* C99 <complex.h> */
118 #define aintf      __aintf      /* C99 <complex.h> */
119 #define aintl      __aintl      /* C99 <complex.h> */
120 #define antint     __antint     /* C99 <complex.h> */
121 #define antintf    __antintf    /* C99 <complex.h> */
122 #define antintl    __antintl    /* C99 <complex.h> */
123 #define annuity    __annuity    /* C99 <complex.h> */
124 #define annuityf   __annuityf   /* C99 <complex.h> */
125 #define annuityyl  __annuityyl  /* C99 <complex.h> */

```



```

126 #define asin          __asin
127 #define asind         __asind
128 #define asindf        __asindf
129 #define asindl        __asindl
130 #define asinf         __asinf
131 #define asinh         __asinh
132 #define asinhf        __asinhf
133 #define asinhl        __asinhl
134 #define asinl         __asinl
135 #define asinp         __asinp
136 #define asinpf        __asinpf
137 #define asinpi        __asinpi
138 #define asinpif       __asinpif
139 #define asinpil       __asinpil
140 #define asinpl        __asinpl
141 #define atan          __atan
142 #define atan2         __atan2
143 #define atan2d        __atan2d
144 #define atan2df       __atan2df
145 #define atan2dl       __atan2dl
146 #define atan2f        __atan2f
147 #define atan2l        __atan2l
148 #define atan2pi       __atan2pi
149 #define atan2pif      __atan2pif
150 #define atan2pil      __atan2pil
151 #define atand         __atand
152 #define atandf        __atandf
153 #define atandl        __atandl
154 #define atanf         __atanf
155 #define atanh         __atanh
156 #define atanhf        __atanhf
157 #define atanhhl       __atanhhl
158 #define atanl         __atanl
159 #define atanp         __atanp
160 #define atanpf        __atanpf
161 #define atanpi        __atanpi
162 #define atanpif       __atanpif
163 #define atanpil       __atanpil
164 #define atanpl        __atanpl
165 #define cbrt         __cbrt
166 #define cbrtf         __cbrtf
167 #define cbrtl         __cbrtl
168 #define ceil         __ceil
169 #define ceilf         __ceilf
170 #define ceill         __ceill
171 #define compound      __compound
172 #define compoundf     __compoundf
173 #define compoundl     __compoundl
174 #define convert_external __convert_external
175 #define convert_external_ __convert_external_
176 #define copysign      __copysign
177 #define copysignf    __copysignf
178 #define copysignl    __copysignl
179 #define cos           __cos
180 #define cosd         __cosd
181 #define cosdf        __cosdf
182 #define cosdl        __cosdl
183 #define cosf         __cosf
184 #define cosh         __cosh
185 #define coshf        __coshf
186 #define coshl        __coshl
187 #define cosl         __cosl
188 #define cosp         __cosp
189 #define cospf        __cospf
190 #define cospi        __cospi
191 #define cospif       __cospif

```

```

192 #define cospil        __cospil
193 #define cospl         __cospl
194 #define d_acos        __d_acos
195 #define d_acosd       __d_acosd
196 #define d_acosh       __d_acosh
197 #define d_acosp       __d_acosp
198 #define d_acospi      __d_acospi
199 #define d_addran      __d_addran
200 #define d_addrans     __d_addrans
201 #define d_aint        __d_aint
202 #define d_anint       __d_anint
203 #define d_annuity     __d_annuity
204 #define d_asin        __d_asin
205 #define d_asind       __d_asind
206 #define d_asinh       __d_asinh
207 #define d_asinp       __d_asinp
208 #define d_asinpi      __d_asinpi
209 #define d_atan2       __d_atan2
210 #define d_atan2d      __d_atan2d
211 #define d_atan2pi     __d_atan2pi
212 #define d_atan        __d_atan
213 #define d_atand       __d_atand
214 #define d_atanh       __d_atanh
215 #define d_atanp       __d_atanp
216 #define d_atanpi      __d_atanpi
217 #define d_cbrt        __d_cbrt
218 #define d_ceil        __d_ceil
219 #define d_compound    __d_compound
220 #define d_copysign    __d_copysign
221 #define d_cos         __d_cos
222 #define d_cosd        __d_cosd
223 #define d_cosh        __d_cosh
224 #define d_cosp        __d_cosp
225 #define d_cospi       __d_cospi
226 #define d_erf         __d_erf
227 #define d_erfc        __d_erfc
228 #define d_exp10       __d_exp10
229 #define d_exp2        __d_exp2
230 #define d_exp         __d_exp
231 #define d_expl        __d_expl
232 #define d_fabs        __d_fabs
233 #define d_floor       __d_floor
234 #define d_fmod        __d_fmod
235 #define d_get_addrans __d_get_addrans
236 #define d_hypot       __d_hypot
237 #define d_infinity    __d_infinity
238 #define d_init_addrans __d_init_addrans
239 #define d_j0          __d_j0
240 #define d_j1          __d_j1
241 #define d_jn          __d_jn
242 #define d_lcrans      __d_lcrans
243 #define d_lcrans      __d_lcrans
244 #define d_lgamma      __d_lgamma
245 #define d_lgamma_r    __d_lgamma_r
246 #define d_log10       __d_log10
247 #define d_loglp       __d_loglp
248 #define d_log2        __d_log2
249 #define d_log         __d_log
250 #define d_logb        __d_logb
251 #define d_max_normal  __d_max_normal
252 #define d_max_subnormal __d_max_subnormal
253 #define d_min_normal  __d_min_normal
254 #define d_min_subnormal __d_min_subnormal
255 #define d_mwcrans     __d_mwcrans
256 #define d_mwcrans     __d_mwcrans
257 #define d_nextafter   __d_nextafter

```

```

258 #define d_pow_           __d_pow_
259 #define d_quiet_nan_     __d_quiet_nan_
260 #define d_remainder_     __d_remainder_
261 #define d_rint_          __d_rint_
262 #define d_scalb_         __d_scalb_
263 #define d_scalbn_        __d_scalbn_
264 #define d_set_addrans_   __d_set_addrans_
265 #define d_shufrens_      __d_shufrens_
266 #define d_signaling_nan_ __d_signaling_nan_
267 #define d_significand_   __d_significand_
268 #define d_sin_           __d_sin_
269 #define d_sincos_        __d_sincos_
270 #define d_sincosd_       __d_sincosd_
271 #define d_sincosp_       __d_sincosp_
272 #define d_sincospi_      __d_sincospi_
273 #define d_sind_          __d_sind_
274 #define d_sinh_          __d_sinh_
275 #define d_sinp_          __d_sinp_
276 #define d_sinpi_        __d_sinpi_
277 #define d_sqrt_          __d_sqrt_
278 #define d_tan_           __d_tan_
279 #define d_tand_          __d_tand_
280 #define d_tanh_          __d_tanh_
281 #define d_tanp_          __d_tanp_
282 #define d_tanpi_        __d_tanpi_
283 #define d_y0_            __d_y0_
284 #define d_y1_            __d_y1_
285 #define d_yn_            __d_yn_
286 #define drem             __drem
287 #define erf               __erf
288 #define erfc              __erfc
289 #define erfcf             __erfcf
290 #define erfcl             __erfcl
291 #define erff              __erff
292 #define erfl             __erfl
293 #define exp               __exp
294 #define exp10             __exp10
295 #define exp10f            __exp10f
296 #define exp10l            __exp10l
297 #define exp2              __exp2
298 #define exp2f             __exp2f
299 #define exp2l            __exp2l
300 #define expf              __expf
301 #define expl              __expl
302 #define expml             __expml
303 #define expmlf            __expmlf
304 #define expmll            __expmll
305 #define fabs              __fabs
306 #define fabsf             __fabsf
307 #define fabsl             __fabsl
308 #define fdim              __fdim           /* C99 */
309 #define fdimf             __fdimf          /* C99 */
310 #define fdiml             __fdiml          /* C99 */
311 #define finitef           __finitf
312 #define finitel           __finitel
313 #define floor             __floor
314 #define floorf            __floorf
315 #define floorl            __floorl
316 #define fma               __fma           /* C99 */
317 #define fmaf              __fmaf          /* C99 */
318 #define fmal              __fmal          /* C99 */
319 #define fmax              __fmax          /* C99 */
320 #define fmaxf             __fmaxf         /* C99 */
321 #define fmaxl             __fmaxl         /* C99 */
322 #define fmin              __fmin          /* C99 */
323 #define fminf             __fminf         /* C99 */

```

```

324 #define fminl            __fminl           /* C99 */
325 #define fmod             __fmod
326 #define fmodf            __fmodf
327 #define fmodl            __fmodl
328 #define fp_class         __fp_class
329 #define fp_classf        __fp_classf
330 #define fp_classl        __fp_classl
331 #define frexp            __frexp           /* S10 */
332 #define frexpf           __frexpf         /* S10 */
333 #define frexpl           __frexpl         /* S10 */
334 #define gamma            __gamma
335 #define gamma_r          __gamma_r
336 #define gammaf           __gammaf
337 #define gammaf_r         __gammaf_r
338 #define gammal           __gammal
339 #define gammal_r         __gammal_r
340 #define gradual_underflow_ __gradual_underflow_
341 #define hypot            __hypot
342 #define hypotf           __hypotf
343 #define hypotl           __hypotl
344 #define i_addran_        __i_addran_
345 #define i_addrans_       __i_addrans_
346 #define i_get_addrans_   __i_get_addrans_
347 #define i_get_lcrans_    __i_get_lcrans_
348 #define i_get_mwcrans_   __i_get_mwcrans_
349 #define i_init_addrans_  __i_init_addrans_
350 #define i_init_lcrans_   __i_init_lcrans_
351 #define i_init_mwcrans_  __i_init_mwcrans_
352 #define i_lcran_         __i_lcran_
353 #define i_lcrans_        __i_lcrans_
354 #define i_llmwcran_      __i_llmwcran_
355 #define i_llmwcrans_     __i_llmwcrans_
356 #define i_mwcran_        __i_mwcran_
357 #define i_mwcrans_       __i_mwcrans_
358 #define i_set_addrans_   __i_set_addrans_
359 #define i_set_lcrans_    __i_set_lcrans_
360 #define i_set_mwcrans_   __i_set_mwcrans_
361 #define i_shufrens_      __i_shufrens_
362 #define id_finite_       __id_finite_
363 #define id_fp_class_     __id_fp_class_
364 #define id_ilogb_        __id_ilogb_
365 #define id_rint_         __id_rint_
366 #define id_isinf_        __id_isinf_
367 #define id_isnan_        __id_isnan_
368 #define id_isnormal_     __id_isnormal_
369 #define id_issubnormal_  __id_issubnormal_
370 #define id_iszero_       __id_iszero_
371 #define id_nint_         __id_nint_
372 #define id_signbit_      __id_signbit_
373 #define ieee_flags       __ieee_flags
374 #define ieee_handler     __ieee_handler
375 #define ieee_handler_    __ieee_handler_
376 #define ieee_handlers    __ieee_handlers
377 #define ieee_handlers_   __ieee_handlers_
378 #define ieee_retrospective __ieee_retrospective
379 #define ieee_retrospective_ __ieee_retrospective_
380 #define ilogb            __ilogb
381 #define ilogbf           __ilogbf
382 #define ilogbl           __ilogbl
383 #define infinity         __infinity
384 #define infinityf        __infinityf
385 #define infinityl        __infinityl
386 #define iq_finite_       __iq_finite_
387 #define iq_fp_class_     __iq_fp_class_
388 #define iq_ilogb_        __iq_ilogb_
389 #define iq_isinf_        __iq_isinf_

```

```

390 #define iq_isnan_           __iq_isnan_
391 #define iq_isnormal_       __iq_isnormal_
392 #define iq_issubnormal_   __iq_issubnormal_
393 #define iq_iszero_        __iq_iszero_
394 #define iq_signbit_       __iq_signbit_
395 #define ir_finite_        __ir_finite_
396 #define ir_fp_class_      __ir_fp_class_
397 #define ir_ilogb_        __ir_ilogb_
398 #define ir_rint_         __ir_rint_
399 #define ir_isinf_        __ir_isinf_
400 #define ir_isnan_        __ir_isnan_
401 #define ir_isnormal_      __ir_isnormal_
402 #define ir_issubnormal_  __ir_issubnormal_
403 #define ir_iszero_       __ir_iszero_
404 #define ir_nint_         __ir_nint_
405 #define ir_signbit_      __ir_signbit_
406 #define irint_           __irint_
407 #define irintf_          __irintf_
408 #define irintl_          __irintl_
409 #define isinf_           __isinf_
410 #define isinff_          __isinff_
411 #define isinfl_          __isinfl_
412 #define isnan_          __isnan_
413 #define isnanf_          __isnanf_
414 #define isnanl_          __isnanl_
415 #define isnormal_        __isnormal_
416 #define isnormalf_       __isnormalf_
417 #define isnormall_       __isnormall_
418 #define issubnormal_     __issubnormal_
419 #define issubnormalf_    __issubnormalf_
420 #define issubnormall_    __issubnormall_
421 #define iszero_          __iszero_
422 #define iszerof_         __iszerof_
423 #define iszerol_         __iszerol_
424 #define j0_              __j0_
425 #define j0f_             __j0f_
426 #define j0l_             __j0l_
427 #define j1_              __j1_
428 #define j1f_             __j1f_
429 #define j1l_             __j1l_
430 #define jn_              __jn_
431 #define jnf_             __jnf_
432 #define jnl_             __jnl_
433 #define ldexp_           __ldexp_           /* S10 */
434 #define ldexpf_          __ldexpf_          /* S10 */
435 #define ldexpl_          __ldexpl_          /* S10 */
436 #define lgamma_          __lgamma_
437 #define lgamma_r_        __lgamma_r_
438 #define lgammaf_         __lgammaf_
439 #define lgammaf_r_       __lgammaf_r_
440 #define lgammal_         __lgammal_
441 #define lgammal_r_       __lgammal_r_
442 #define llrint_          __llrint_          /* C99 */
443 #define llrintf_         __llrintf_         /* C99 */
444 #define llrintl_         __llrintl_         /* C99 */
445 #define llround_         __llround_         /* C99 */
446 #define llroundf_        __llroundf_        /* C99 */
447 #define llroundl_        __llroundl_        /* C99 */
448 #define lrint_           __lrint_           /* C99 */
449 #define lrintf_          __lrintf_          /* C99 */
450 #define lrintl_          __lrintl_          /* C99 */
451 #define lround_          __lround_          /* C99 */
452 #define lroundf_         __lroundf_         /* C99 */
453 #define lroundl_         __lroundl_         /* C99 */
454 #define log_             __log_
455 #define log10_           __log10_

```

```

456 #define log10f_          __log10f_
457 #define log10l_          __log10l_
458 #define log1p_           __log1p_
459 #define log1pf_          __log1pf_
460 #define log1pl_          __log1pl_
461 #define log2_            __log2_
462 #define log2f_           __log2f_
463 #define log2l_           __log2l_
464 #define logb_            __logb_
465 #define logbf_           __logbf_
466 #define logbl_           __logbl_
467 #define logf_            __logf_
468 #define logl_            __logl_
469 #define max_normal_      __max_normal_
470 #define max_normalf_     __max_normalf_
471 #define max_normall_     __max_normall_
472 #define max_subnormal_   __max_subnormal_
473 #define max_subnormalf_  __max_subnormalf_
474 #define max_subnormall_  __max_subnormall_
475 #define min_normal_      __min_normal_
476 #define min_normalf_     __min_normalf_
477 #define min_normall_     __min_normall_
478 #define min_subnormal_   __min_subnormal_
479 #define min_subnormalf_  __min_subnormalf_
480 #define min_subnormall_  __min_subnormall_
481 #define modf_            __modf_           /* S10 */
482 #define modff_           __modff_          /* S10 */
483 #define modfl_           __modfl_          /* S10 */
484 #define nan_             __nan_            /* C99 */
485 #define nanf_            __nanf_           /* C99 */
486 #define nanl_            __nanl_           /* C99 */
487 #define nearbyint_       __nearbyint_      /* C99 */
488 #define nearbyintf_      __nearbyintf_     /* C99 */
489 #define nearbyintl_      __nearbyintl_     /* C99 */
490 #define nextafter_        __nextafter_
491 #define nextafterf_       __nextafterf_
492 #define nextafterl_       __nextafterl_
493 #define nexttoward_       __nexttoward_     /* C99 */
494 #define nexttowardf_      __nexttowardf_    /* C99 */
495 #define nexttowardl_     __nexttowardl_    /* C99 */
496 #define nint_            __nint_
497 #define nintf_           __nintf_
498 #define nintl_           __nintl_
499 #define nonstandard_arithmetic_ __nonstandard_arithmetic_
500 #define nonstandard_arithmetic_ __nonstandard_arithmetic_
501 #define pow_             __pow_
502 #define pow_di_          __pow_di_
503 #define pow_li_          __pow_li_
504 #define pow_ri_          __pow_ri_
505 #define powf_            __powf_
506 #define powl_            __powl_
507 #define q_copysign_      __q_copysign_
508 #define q_fabs_          __q_fabs_
509 #define q_fmod_          __q_fmod_
510 #define q_infinity_      __q_infinity_
511 #define q_max_normal_    __q_max_normal_
512 #define q_max_subnormal_ __q_max_subnormal_
513 #define q_min_normal_    __q_min_normal_
514 #define q_min_subnormal_ __q_min_subnormal_
515 #define q_nextafter_     __q_nextafter_
516 #define q_quiet_nan_     __q_quiet_nan_
517 #define q_remainder_     __q_remainder_
518 #define q_scalbn_        __q_scalbn_
519 #define q_signaling_nan_ __q_signaling_nan_
520 #define quiet_nan_       __quiet_nan_
521 #define quiet_nanf_      __quiet_nanf_

```

```

522 #define quiet_nanl          __quiet_nanl
523 #define r_acos_             _r_acos_
524 #define r_acosd_           _r_acosd_
525 #define r_acosh_           _r_acosh_
526 #define r_acosp_           _r_acosp_
527 #define r_acospi_          _r_acospi_
528 #define r_addran_          _r_addran_
529 #define r_addrans_         _r_addrans_
530 #define r_aint_            _r_aint_
531 #define r_anint_           _r_anint_
532 #define r_annuity_         _r_annuity_
533 #define r_asin_            _r_asin_
534 #define r_asind_           _r_asind_
535 #define r_asinh_           _r_asinh_
536 #define r_asinp_           _r_asinp_
537 #define r_asinpi_         _r_asinpi_
538 #define r_atan2_           _r_atan2_
539 #define r_atan2d_          _r_atan2d_
540 #define r_atan2pi_         _r_atan2pi_
541 #define r_atan_            _r_atan_
542 #define r_atand_           _r_atand_
543 #define r_atanh_           _r_atanh_
544 #define r_atanp_           _r_atanp_
545 #define r_atanpi_          _r_atanpi_
546 #define r_cbrt_           _r_cbrt_
547 #define r_ceil_           _r_ceil_
548 #define r_compound_        _r_compound_
549 #define r_copysign_        _r_copysign_
550 #define r_cos_             _r_cos_
551 #define r_cosd_            _r_cosd_
552 #define r_cosh_            _r_cosh_
553 #define r_cosp_            _r_cosp_
554 #define r_cospi_          _r_cospi_
555 #define r_erf_             _r_erf_
556 #define r_erfc_           _r_erfc_
557 #define r_exp10_          _r_exp10_
558 #define r_exp2_           _r_exp2_
559 #define r_exp_            _r_exp_
560 #define r_expml_          _r_expml_
561 #define r_fabs_           _r_fabs_
562 #define r_floor_          _r_floor_
563 #define r_fmod_           _r_fmod_
564 #define r_get_addrans_    _r_get_addrans_
565 #define r_hypot_          _r_hypot_
566 #define r_infinity_       _r_infinity_
567 #define r_init_addrans_   _r_init_addrans_
568 #define r_j0_             _r_j0_
569 #define r_j1_             _r_j1_
570 #define r_jn_             _r_jn_
571 #define r_lcran_          _r_lcran_
572 #define r_lcrans_         _r_lcrans_
573 #define r_lgamma_         _r_lgamma_
574 #define r_lgamma_r_       _r_lgamma_r_
575 #define r_log10_          _r_log10_
576 #define r_loglp_          _r_loglp_
577 #define r_log2_           _r_log2_
578 #define r_log_            _r_log_
579 #define r_logb_           _r_logb_
580 #define r_max_normal_     _r_max_normal_
581 #define r_max_subnormal_  _r_max_subnormal_
582 #define r_min_normal_     _r_min_normal_
583 #define r_min_subnormal_  _r_min_subnormal_
584 #define r_mwcran_         _r_mwcran_
585 #define r_mwcrans_        _r_mwcrans_
586 #define r_nextafter_      _r_nextafter_
587 #define r_pow_            _r_pow_

```

```

588 #define r_quiet_nan_       _r_quiet_nan_
589 #define r_remainder_       _r_remainder_
590 #define r_rint_           _r_rint_
591 #define r_scalb_          _r_scalb_
592 #define r_scalbn_         _r_scalbn_
593 #define r_set_addrans_    _r_set_addrans_
594 #define r_shufans_        _r_shufans_
595 #define r_signaling_nan_  _r_signaling_nan_
596 #define r_significand_    _r_significand_
597 #define r_sin_            _r_sin_
598 #define r_sincos_         _r_sincos_
599 #define r_sincosd_        _r_sincosd_
600 #define r_sincosp_        _r_sincosp_
601 #define r_sincospi_       _r_sincospi_
602 #define r_sind_           _r_sind_
603 #define r_sinh_           _r_sinh_
604 #define r_sinp_           _r_sinp_
605 #define r_sinpi_          _r_sinpi_
606 #define r_sqrt_           _r_sqrt_
607 #define r_tan_            _r_tan_
608 #define r_tand_           _r_tand_
609 #define r_tanh_           _r_tanh_
610 #define r_tanp_           _r_tanp_
611 #define r_tanpi_          _r_tanpi_
612 #define r_y0_            _r_y0_
613 #define r_y1_            _r_y1_
614 #define r_yn_            _r_yn_
615 #define remainder         __remainder
616 #define remainderf        __remainderf
617 #define remainderl        __remainderl
618 #define remquo             _remquo /* C99 */
619 #define remquoof           _remquoof /* C99 */
620 #define remquoofl          _remquoofl /* C99 */
621 #define rint               _rint
622 #define rintf              _rintf
623 #define rintl              _rintl
624 #define round              _round /* C99 */
625 #define roundf             _roundf /* C99 */
626 #define roundl             _roundl /* C99 */
627 #define scalb              _scalb
628 #define scalbf             _scalbf
629 #define scalbl             _scalbl
630 #define scalbln            _scalbln /* C99 */
631 #define scalblnlf          _scalblnlf /* C99 */
632 #define scalblnl          _scalblnl /* C99 */
633 #define scalbn             _scalbn
634 #define scalbnf            _scalbnf
635 #define scalbnl            _scalbnl
636 #define sigfpe             _sigfpe
637 #define sigfpe_           _sigfpe_
638 #define signaling_nan      _signaling_nan
639 #define signaling_nanf     _signaling_nanf
640 #define signaling_nanl     _signaling_nanl
641 #define signbit            _signbit
642 #define signbitf           _signbitf
643 #define signbitl           _signbitl
644 #define signgam            _signgam
645 #define signgamf           _signgamf
646 #define signgaml           _signgaml
647 #define significand        _significand
648 #define significandf       _significandf
649 #define significandl       _significandl
650 #define sin                _sin
651 #define sincos             _sincos
652 #define sincosd            _sincosd
653 #define sincosdf           _sincosdf

```

```

654 #define sincosdl      __sincosdl
655 #define sincosf      __sincosf
656 #define sincosl      __sincosl
657 #define sincosp      __sincosp
658 #define sincospf     __sincospf
659 #define sincospi     __sincospi
660 #define sincospif    __sincospif
661 #define sincospil    __sincospil
662 #define sincospl     __sincospl
663 #define sind         __sind
664 #define sindf        __sindf
665 #define sindl        __sindl
666 #define sinf         __sinf
667 #define sinh         __sinh
668 #define sinhf        __sinhf
669 #define sinhl        __sinhl
670 #define sinl         __sinl
671 #define sinp         __sinp
672 #define sinpf        __sinpf
673 #define sinpi        __sinpi
674 #define sinpif       __sinpif
675 #define sinpil       __sinpil
676 #define sinpl        __sinpl
677 #define smwcran_     __smwcran_
678 #define sqrt         __sqrt
679 #define sqrtf        __sqrtf
680 #define sqrtl        __sqrtl
681 #define standard_arithmetic __standard_arithmetic
682 #define standard_arithmetic_ __standard_arithmetic_
683 #define tan          __tan
684 #define tand         __tand
685 #define tandf        __tandf
686 #define tandl        __tandl
687 #define tanf         __tanf
688 #define tanh         __tanh
689 #define tanhf        __tanhf
690 #define tanhl        __tanhl
691 #define tanl         __tanl
692 #define tanp         __tanp
693 #define tanpf        __tanpf
694 #define tanpi        __tanpi
695 #define tanpif       __tanpif
696 #define tanpil       __tanpil
697 #define tanpl        __tanpl
698 #define tgamma       __tgamma      /* C99 */
699 #define tgammaf      __tgammaf     /* C99 */
700 #define tgammal      __tgammal     /* C99 */
701 #define trunc        __trunc        /* C99 */
702 #define truncf       __truncf       /* C99 */
703 #define truncL      __truncL       /* C99 */
704 #define u_addrans_   __u_addrans_
705 #define u_lcrans_    __u_lcrans_
706 #define u_llmwcran_  __u_llmwcran_
707 #define u_llmwcrans_ __u_llmwcrans_
708 #define u_mwcran_    __u_mwcran_
709 #define u_mwcrans_   __u_mwcrans_
710 #define u_shufrans_  __u_shufrans_
711 #define y0           __y0
712 #define y0f          __y0f
713 #define y0l          __y0l
714 #define y1           __y1
715 #define y1f          __y1f
716 #define y1l          __y1l
717 #define yn           __yn
718 #define ynf          __ynf
719 #define ynl          __ynl

```

```

721 /*
722 * these are libdl entry points
723 */
724 #define dlclose        _dlclose
725 #define dlopen         _dlopen
726 #define dlsym          _dlsym

728 /*
729 * these are libc entry points
730 */
731 #define finite         _finite
732 #define fpclass        _fpclass
733 #define isnand         _isnand
734 #define sigaction      _sigaction
735 #define sigemptyset    _sigemptyset
736 #define unordered     _unordered
737 #define write          _write
738 #ifdef _REENTRANT
739 #define mutex_lock     _mutex_lock
740 #define mutex_unlock   _mutex_unlock
741 #define thr_getspecific _thr_getspecific
742 #define thr_keycreate  _thr_keycreate
743 #define thr_main       _thr_main
744 #define thr_setspecific _thr_setspecific
745 #endif

747 #endif /* !defined(lint) */
747 #endif /* defined(ELFOBJ) && !defined(lint) */

749 #endif /* _LIBM_SYNONYMS_H */

```

new/usr/src/lib/libm/common/C/logb.c

1

```
*****
2118 Mon Oct 27 12:36:41 2014
new/usr/src/lib/libm/common/C/logb.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak logb = __logb
31 #pragma weak _logb = __logb
33 #endif

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

37 #if defined(__x86)
39 #if defined(USE_FPSCALE) || defined(__x86)
38 static const double two52 = 4503599627370496.0;
39 #else
40 /*
41  * v: high part of a non-zero subnormal |x|; w: low part of |x|
42  */
43 static int
44 ilogb_subnormal(unsigned v, unsigned w) {
45     int r = -1022 - 52;

47     if (v)
48         r += 32;
49     else
50         v = w;
51     if (v & 0xffff0000)
52         r += 16, v >>= 16;
53     if (v & 0xff00)
54         r += 8, v >>= 8;
55     if (v & 0xf0)
56         r += 4, v >>= 4;
57     v <<= 1;
```

new/usr/src/lib/libm/common/C/logb.c

2

```
58     return (r + ((0xffffaa50 >> v) & 0x3));
59 }
60 #endif /* defined(__x86) */
62 #endif /* defined(USE_FPSCALE) */

62 double
63 logb(double x) {
64     int *px = (int *) &x, k = px[HIWORD] & ~0x80000000;

66     if (k < 0x00100000) {
67         if ((px[LOWORD] | k) == 0)
68             return (_SVID_libm_err(x, x, 45));
69         else if ((__xpg6 & _C99SUSv3_logb) != 0) {
70 #if defined(__x86)
72 #if defined(USE_FPSCALE) || defined(__x86)
71             x *= two52;
72             return ((double) (((px[HIWORD] & 0x7ff00000) >> 20)
73                 - 1075));
74 #else
75             return ((double) ilogb_subnormal(k, px[LOWORD]));
76 #endif
77         } else
78             return (-1022.0);
79     } else if (k < 0x7ff00000)
80         return ((double) ((k >> 20) - 1023));
81     else
82         return (x * x);
83 }

unchanged_portion_omitted
```

new/usr/src/lib/libm/common/C/pow.c

1

```
*****
10175 Mon Oct 27 12:36:41 2014
new/usr/src/lib/libm/common/C/pow.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak pow = __pow
32 #endif

32 /*
33  * pow(x,y) return x**y
34  *
35  * Method: Let  $x = 2^n * (1+f)$ 
36  * 1. Compute and return  $\log_2(x)$  in two pieces:
37  *     $\log_2(x) = w1 + w2,$ 
38  *    where w1 has 24 bits trailing zero.
39  * 2. Perform  $y * \log_2(x)$  by simulating muti-precision arithmetic
40  * 3. Return  $x**y = \exp_2(y * \log(x))$ 
41  *
42  * Special cases:
43  * 1. (anything) ** +-0 is 1
44  * 1'. 1 ** (anything) is 1 (C99; 1 ** +-INF/NAN used to be NAN)
45  * 2. (anything) ** 1 is itself
46  * 3. (anything except 1) ** NAN is NAN ("except 1" is C99)
47  * 4. NAN ** (anything except 0) is NAN
48  * 5. +-(|x| > 1) ** +INF is +INF
49  * 6. +-(|x| > 1) ** -INF is +0
50  * 7. +-(|x| < 1) ** +INF is +0
51  * 8. +-(|x| < 1) ** -INF is +INF
52  * 9. -1 ** +-INF is 1 (C99; -1 ** +-INF used to be NAN)
53  * 10. +0 ** (+anything except 0, NAN) is +0
54  * 11. -0 ** (+anything except 0, NAN, odd integer) is +0
55  * 12. +0 ** (-anything except 0, NAN) is +INF
56  * 13. -0 ** (-anything except 0, NAN, odd integer) is +INF
57  * 14. -0 ** (odd integer) = -(+0 ** (odd integer))
58  * 15. +INF ** (+anything except 0,NAN) is +INF
```

new/usr/src/lib/libm/common/C/pow.c

2

```
59 * 16. +INF ** (-anything except 0,NAN) is +0
60 * 17. -INF ** (anything) = -0 ** (-anything)
61 * 18. (-anything) ** (integer) is (-1)**(integer)*(+anything**integer)
62 * 19. (-anything except 0 and inf) ** (non-integer) is NAN
63 *
64 * Accuracy:
65 * pow(x,y) returns x**y nearly rounded. In particular
66 * pow(integer,integer)
67 * always returns the correct integer provided it is representable.
68 */

70 #include "libm.h"
71 #include "xpg6.h" /* __xpg6 */
72 #define _C99SUSv3_pow_treats_Inf_as_an_even_int

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

76 extern const double _TBL_log2_hi[], _TBL_log2_lo[];
77 static const double
78 two53 = 9007199254740992.0,
79 Al_hi = 2.8853900432586669921875,
80 Al_lo = 3.8519259825035041963606002e-8,
81 A1 = 2.885390081777926817222541963606002026086e+0000,
82 A2 = 9.617966939207270828380543979852286255862e-0001,
83 A3 = 5.770807680887875964868853124873696201995e-0001,
84 B0_hi = 2.8853900432586669921875,
85 B0_lo = 3.8519259822532793056374320585e-8,
86 B0 = 2.885390081777926814720293056374320585689e+0000,
87 B1 = 9.617966939259755138949202350396200257632e-0001,
88 B2 = 5.770780163585687000782112776448797953382e-0001,
89 B3 = 4.121985488948771523290174512461778354953e-0001,
90 B4 = 3.207590534812432970433641789022666850193e-0001;

92 static double
93 log2_x(double x, double *w) {
94     double f, s, z, qn, h, t;
95     int *px = (int *) &x;
96     int *pz = (int *) &z;
97     int i, j, ix, ni;

99     n = 0;
100    ix = px[HIWORD];
101    if (ix >= 0x3fef03f1 && ix < 0x3ff08208) { /* 65/63 > x > 63/65 */
102        double f1, v;
103        f = x - one;
104        if (((ix - 0x3ff00000) | px[LOWORD]) == 0) {
105            *w = zero;
106            return (zero); /* log2(1) = +0 */
107        }
108        qn = one / (two + f);
109        s = f * qn; /* |s| < 2**-6 */
110        v = s * s;
111        h = (double) ((float) s);
112        f1 = (double) ((float) f);
113        t = qn * (((f - two * h) - h * f1) - h * (f - f1)); /* s = h+t */
114        f1 = h * B0_lo + s * (v * (B1 + v * (B2 + v * (B3 + v * B4))));
115        t = f1 + t * B0;
116        h *= B0_hi;
117        s = (double) ((float) (h + t));
118        *w = t - (s - h);
119        return (s);
120    }
121    }
122    if (ix < 0x00100000) { /* subnormal x */
123        x *= two53;
124        n = -53;
```

```
125         ix = px[HIWORD];
126     }
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_
```



```

*****
2763 Mon Oct 27 12:36:42 2014
new/usr/src/lib/libm/common/C/scalbn.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 scalbn = __scalbn

31 #include "libm.h"

33 static const double
34     one = 1.0,
35     huge = 1.0e300,
36     tiny = 1.0e-300,
37     twom54 = 5.5511151231257827021181583404541015625e-17;

39 #if defined(__x86)
39 #if defined(USE_FPSCALE) || defined(__x86)
40 static const double two52 = 4503599627370496.0;
41 #else
42 /*
43  * Normalize non-zero subnormal x and return biased exponent of x in [-51,0]
44  */
45 static int
46 ilogb_biased(unsigned *px) {
47     int s = 52;
48     unsigned v = px[HIWORD] & ~0x80000000, w = px[LOWORD], t = v;

50     if (t)
51         s -= 32;
52     else
53         t = w;
54     if (t & 0xffff0000)
55         s -= 16, t >>= 16;
56     if (t & 0xff00)
57         s -= 8, t >>= 8;
58     if (t & 0xf0)
59         s -= 4, t >>= 4;

```

```

60     t <= 1;
61     s -= (0xfffffaa50 >> t) & 0x3;
62     if (s < 32) {
63         v = (v << s) | w >> (32 - s);
64         w <= s;
65     } else {
66         v = w << (s - 32);
67         w = 0;
68     }
69     px[HIWORD] = (px[HIWORD] & 0x80000000) | v;
70     px[LOWORD] = w;
71     return (1 - s);
72 }
73 #endif /* defined(__x86) */
73 #endif /* defined(USE_FPSCALE) */

75 double
76 scalbn(double x, int n) {
77     int *px, ix, hx, k;

79     px = (int *)&x;
80     ix = px[HIWORD];
81     hx = ix & ~0x80000000;
82     k = hx >> 20;

84     if (k == 0x7ff) /* x is inf or NaN */
85         return (x * one);

87     if (k == 0) {
88         if ((hx | px[LOWORD]) == 0 || n == 0)
89             return (x);
90 #if defined(__x86)
90 #if defined(USE_FPSCALE) || defined(__x86)
91         x *= two52;
92         ix = px[HIWORD];
93         k = ((ix & ~0x80000000) >> 20) - 52;
94 #else
95         k = ilogb_biased((unsigned *)px);
96         ix = px[HIWORD];
97 #endif
98         /* now k is in the range -51..0 */
99         k += n;
100        if (k > n) /* integer overflow occurred */
101            k = -100;
102    } else {
103        /* k is in the range 1..1023 */
104        k += n;
105        if (k < n) /* integer overflow occurred */
106            k = 0x7ff;
107    }

109    if (k > 0x7fe)
110        return (huge * ((ix < 0)? -huge : huge));
111    if (k < 1) {
112        if (k <= -54)
113            return (tiny * ((ix < 0)? -tiny : tiny));
114        k += 54;
115        px[HIWORD] = (ix & ~0x7ff00000) | (k << 20);
116        return (x * twom54);
117    }
118    px[HIWORD] = (ix & ~0x7ff00000) | (k << 20);
119    return (x);
120 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/C/significand.c

1

\*\*\*\*\*

1455 Mon Oct 27 12:36:42 2014

new/usr/src/lib/libm/common/C/significand.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak significand = __significand
32 #endif
33
34 #include "libm.h"
35
36 double
37 significand(double x) {
38     int ix = ((int *) &x)[HIWORD] & ~0x80000000;
39
40     /* weed out 0/+-Inf/NaN because C99 ilogb raises invalid on them */
41     if ((ix | ((int *) &x)[LOWORD]) == 0 || ix >= 0x7ff00000)
42         return ((ix & 0x80000) != 0 ? x : x + x);
43     /* assumes sparc-like QNaN */
44 #else
45     return (x + x);
46 #endif
47     else
48         return (scalbn(x, -ilogb(x)));
49 }
50
51 _____unchanged_portion_omitted_____
```

```

*****
1220 Mon Oct 27 12:36:42 2014
new/usr/src/lib/libm/common/C/sqrt.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 sqrt = __sqrt

31 #include "libm.h"

33 #ifdef __INLINE

34 extern double __inline_sqrt(double);

36 double
37 sqrt(double x) {
38     double z = __inline_sqrt(x);

40     if (isnan(x))
41         return (z);
42     return ((x < 0.0)? _SVID_libm_err(x, x, 26) : z);
43 }

46 #else /* defined(__INLINE) */

48 /*
49  * Warning: This correctly rounded sqrt is extremely slow because it computes
50  * the sqrt bit by bit using integer arithmetic.
51  */

53 static const double big = 1.0e30, small = 1.0e-30;

55 double
56 sqrt(double x)
57 {
58     double z;
59     unsigned r, t1, s1, ix1, q1;
60     int ix0, s0, j, q, m, n, t;

```

```

61     int *px = (int *)&x, *pz = (int *)&z;

63     ix0 = px[HIWORD];
64     ix1 = px[LOWORD];
65     if ((ix0 & 0x7ff00000) == 0x7ff00000) { /* x is inf or NaN */
66         if (ix0 == 0xfff00000 && ix1 == 0)
67             return (_SVID_libm_err(x, x, 26));
68         return (x + x);
69     }
70     if (((ix0 & 0x7fffffff) | ix1) == 0) /* x is zero */
71         return (x);

73     /* extract exponent and significand */
74     m = ilogb(x);
75     z = scalbn(x, -m);
76     ix0 = (pz[HIWORD] & 0x000fffff) | 0x00100000;
77     ix1 = pz[LOWORD];
78     n = m >> 1;
79     if (n + n != m) {
80         ix0 = (ix0 << 1) | (ix1 >> 31);
81         ix1 <<= 1;
82         m -= 1;
83     }

85     /* generate sqrt(x) bit by bit */
86     ix0 = (ix0 << 1) | (ix1 >> 31);
87     ix1 <<= 1;
88     q = q1 = s0 = s1 = 0;
89     r = 0x00200000;

91     for (j = 1; j <= 22; j++) {
92         t = s0 + r;
93         if (t <= ix0) {
94             s0 = t + r;
95             ix0 -= t;
96             q += r;
97         }
98         ix0 = (ix0 << 1) | (ix1 >> 31);
99         ix1 <<= 1;
100        r >>= 1;
101    }

103    r = 0x80000000;
104    for (j = 1; j <= 32; j++) {
105        t1 = s1 + r;
106        t = s0;
107        if (t < ix0 || (t == ix0 && t1 <= ix1)) {
108            s1 = t1 + r;
109            if ((t1 & 0x80000000) == 0x80000000 &&
110                (s1 & 0x80000000) == 0)
111                s0 += 1;
112            ix0 -= t;
113            if (ix1 < t1)
114                ix0 -= 1;
115            ix1 -= t1;
116            q1 += r;
117        }
118        ix0 = (ix0 << 1) | (ix1 >> 31);
119        ix1 <<= 1;
120        r >>= 1;
121    }

123    /* round */
124    if ((ix0 | ix1) == 0)
125        goto done;
126    z = big - small; /* trigger inexact flag */

```

```
127     if (z < big)
128         goto done;
129     if (q1 == 0xffffffff) {
130         q1 = 0;
131         q += 1;
132         goto done;
133     }
134     z = big + small;
135     if (z > big) {
136         if (q1 == 0xffffffff)
137             q += 1;
138         q1 += 2;
139         goto done;
140     }
141     q1 += (q1 & 1);
142 done:
143     pz[HIWORD] = (q >> 1) + 0x3fe00000;
144     pz[LOWORD] = q1 >> 1;
145     if ((q & 1) == 1)
146         pz[LOWORD] |= 0x80000000;
147     return (scalbn(z, n));
148 }
149
150 #endif /* defined(__INLINE) */
```

new/usr/src/lib/libm/common/LD/acoshl.c

1

\*\*\*\*\*

1471 Mon Oct 27 12:36:43 2014

new/usr/src/lib/libm/common/LD/acoshl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 #if defined(ELFOBJ)
30 #pragma weak acoshl = __acoshl
32 #endif

32 #include "libm.h"

34 static const long double
35     zero      = 0.0L,
36     ln2       = 6.931471805599453094172321214581765680755e-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 (loglpl(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
```

new/usr/src/lib/libm/common/LD/asinhl.c

1

```
*****
1610 Mon Oct 27 12:36:43 2014
new/usr/src/lib/libm/common/LD/asinhl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak asinhl = __asinhl
32 #endif

32 #include "libm.h"

34 static const long double
35     ln2      = 6.931471805599453094172321214581765680755e-0001L,
36     one      = 1.0L,
37     big      = 1.0e+20L,
38     tiny     = 1.0e-20L;

40 long double
41 asinhl(long double x) {
42     long double t, w;
43 #ifndef lint
44     volatile long double dummy;
45 #endif

47     w = fabsl(x);
48     if (isnanl(x))
49         return (x + x); /* x is NaN */
50     if (w < tiny) {
51 #ifndef lint
52         dummy = x + big; /* inexact if x != 0 */
53 #endif
54         return (x); /* tiny x */
55     } else if (w < big) {
56         t = one / w;
57         return (copysignl(loglpl(w + w / (t + sqrtl(one + t * t))), x));
58     } else
```

new/usr/src/lib/libm/common/LD/asinhl.c

2

```
59         return (copysignl(logl(w) + ln2, x));
60     }
_____unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/LD/cbrtl.c

1

```
*****
1752 Mon Oct 27 12:36:44 2014
new/usr/src/lib/libm/common/LD/cbrtl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak cbrtl = __cbrtl
32 #endif

32 #include "libm.h"
33 #include "longdouble.h"

35 static const double d_one = 1.0;

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

44     if (!finitel(x))
45         return (x + x);
46     if (iszerol(x))
47         return (x);
48     n0 = 0;
49     if (*((int *) &d_one) == 0)
50         n0 = 1;
51     sx = signbitl(x);
52     x = fabsl(x);
53     n = ilogbl(x);
54     m = n / 3;
55     m3 = m + m + m;
56     y = scalbnl(x, -m3);
57     dx = (double) y;
58     dy = cbrt(dx);
```

new/usr/src/lib/libm/common/LD/cbrtl.c

2

```
59     py[1 - n0] += 2;
60     if (py[1 - n0] == 0)
61         py[n0] += 1;

63     /* one step newton iteration to 113 bits with error < 0.667ulps */
64     t = (long double) dy;
65     t = scalbnl(t, m);
66     s = t * t;
67     r = x / s;
68     w = t + t;
69     r = (r - t) / (w + r);
70     t += t * r;

72     return (sx == 0 ? t : -t);
73 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/LD/coshl.c

1

```
*****
2788 Mon Oct 27 12:36:44 2014
new/usr/src/lib/libm/common/LD/coshl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak coshl = __coshl
32 #endif

32 #include "libm.h"
33 #include "longdouble.h"

35 /*
36 * COSH(X)
37 * RETURN THE HYPERBOLIC COSINE OF X
38 *
39 * Method :
40 * 1. Replace x by |x| (COSH(x) = COSH(-x)).
41 * 2.
42 *
43 * 0 <= x <= 0.3465 : COSH(x) := 1 + -----
44 *                                     [ EXP(x) - 1 ]^2
45 *                                     2*EXP(x)
46 *
47 * 0.3465 <= x <= thresh : COSH(x) := -----
48 *                                     EXP(x) + 1/EXP(x)
49 *                                     2
50 * thresh <= x <= lnovft : COSH(x) := EXP(x)/2
51 * lnovft <= x < INF : COSH(x) := SCALBN(EXP(x-MEPl*ln2),ME)
52 *
53 * here
54 * 0.3465 a number that is near one half of ln2.
55 * thresh a number such that
56 *          EXP(thresh)+EXP(-thresh)=EXP(thresh)
57 * lnovft logarithm of the overflow threshold
58 *          = MEPl*ln2 chopped to machine precision.
```

new/usr/src/lib/libm/common/LD/coshl.c

2

```
59 * ME maximum exponent
60 * MEPl maximum exponent plus 1
61 *
62 * Special cases:
63 * COSH(x) is |x| if x is +INF, -INF, or NaN.
64 * only COSH(0)=1 is exact for finite x.
65 */

67 static const long double C[] = {
68 0.5L,
69 1.0L,
70 0.3465L,
71 45.0L,
72 1.135652340629414394879149e+04L,
73 7.004447686242549087858985e-16L,
74 2.710505431213761085018632e-20L,
75 }; /* 2^-65 */

_____unchanged_portion_omitted_____
```



new/usr/src/lib/libm/common/LD/finitel.c

1

```
*****
1446 Mon Oct 27 12:36:44 2014
new/usr/src/lib/libm/common/LD/finitel.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
```

```
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 #if defined(ELFOBJ)
30 #pragma weak finitel = __finitel
32 #endif

32 #include "libm.h"

34 #if defined(__sparc)
35 int
36 finitel(long double x) {
37     int *px = (int *) &x;
38     return ((px[0] & ~0x80000000) < 0x7fff0000);
39 }
unchanged_portion_omitted
```

```

*****
3742 Mon Oct 27 12:36:45 2014
new/usr/src/lib/libm/common/LD/hypot1.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak hypot1 = __hypot1
32 #endif

32 /*
33 * hypot(x,y)
34 * Method :
35 * If z=x*x+y*y has error less than sqrt(2)/2 ulp than sqrt(z) has
36 * error less than 1 ulp.
37 * So, compute sqrt(x*x+y*y) with some care as follows:
38 * Assume x>y>0:
39 * 1. save and set rounding to round-to-nearest
40 * 2. if x > 2y use
41 *    x1*x1+(y*y+(x2*(x+x2))) for x*x+y*y
42 * where x1 = x with lower 32 bits cleared, x2 = x-x1; else
43 * 3. if x <= 2y use
44 *    t1*y1+((x-y)*(x-y)+(t1*y2+t2*y))
45 * where t1 = 2x with lower 64 bits cleared, t2 = 2x-t1, y1 = y with
46 * lower 32 bits cleared, y2 = y-y1.
47 *
48 * NOTE: DO NOT remove parenthesis!
49 *
50 * Special cases:
51 * hypot(x,y) is INF if x or y is +INF or -INF; else
52 * hypot(x,y) is NAN if x or y is NAN.
53 *
54 * Accuracy:
55 * hypot(x,y) returns sqrt(x^2+y^2) with error less than 1 ulps (units
56 * in the last place)
57 */

```

```

59 #include "libm.h"

61 #if defined(__x86)
62 extern enum fp_direction_type __swap87RD(enum fp_direction_type);

64 #define k          0x7fff

66 long double
67 hypot1(long double x, long double y) {
68     long double t1, t2, y1, y2, w;
69     int *px = (int *) &x, *py = (int *) &y;
70     int *pt1 = (int *) &t1, *py1 = (int *) &y1;
71     enum fp_direction_type rd;
72     int j, nx, ny, nz;

74     px[2] &= 0x7fff;          /* clear sign bit and padding bits of x and y */
75     py[2] &= 0x7fff;
76     nx = px[2];              /* biased exponent of x and y */
77     ny = py[2];
78     if (ny > nx) {
79         w = x;
80         x = y;
81         y = w;
82         nz = ny;
83         ny = nx;
84         nx = nz;
85     }                          /* force nx >= ny */
86     if (nx - ny >= 66)
87         return (x + y); /* x / y >= 2**65 */
88     if (nx < 0x5fff3 && ny > 0x205b) { /* medium x,y */
89         /* save and set RD to Rounding to nearest */
90         rd = __swap87RD(fp_nearest);
91         w = x - y;
92         if (w > y) {
93             pt1[2] = px[2];
94             pt1[1] = px[1];
95             pt1[0] = 0;
96             t2 = x - t1;
97             x = sqrtl(t1 * t1 - (y * (-y) - t2 * (x + t1)));
98         } else {
99             x += x;
100            py1[2] = py[2];
101            py1[1] = py[1];
102            py1[0] = 0;
103            y2 = y - y1;
104            pt1[2] = px[2];
105            pt1[1] = px[1];
106            pt1[0] = 0;
107            t2 = x - t1;
108            x = sqrtl(t1 * y1 - (w * (-w) - (t2 * y1 + y2 * x)));
109        }
110        if (rd != fp_nearest)
111            __swap87RD(rd); /* restore rounding mode */
112        return (x);
113    } else {
114        if (nx == k || ny == k) { /* x or y is INF or NaN */
115            /* since nx >= ny; nx is always k within this block */
116            if (px[1] == 0x80000000 && px[0] == 0)
117                return (x);
118            else if (ny == k && py[1] == 0x80000000 && py[0] == 0)
119                return (y);
120            else
121                return (x + y);
122        }
123        if (ny == 0) {
124            if (y == 0.L || x == 0.L)

```

```
125         return (x + y);
126         ptl[2] = 0x3fff + 16381;
127         ptl[1] = 0x80000000;
128         ptl[0] = 0;
129         pyl[2] = 0x3fff - 16381;
130         pyl[1] = 0x80000000;
131         pyl[0] = 0;
132         x *= t1;
133         y *= t1;
134         return (y1 * hypot1(x, y));
135     }
136     j = nx - 0x3fff;
137     px[2] -= j;
138     py[2] -= j;
139     ptl[2] = nx;
140     ptl[1] = 0x80000000;
141     ptl[0] = 0;
142     return (t1 * hypot1(x, y));
143 }
144 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/LD/isnanl.c

1

\*\*\*\*\*

1585 Mon Oct 27 12:36:45 2014

new/usr/src/lib/libm/common/LD/isnanl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak isnanl = __isnanl
32 #endif
33
34 #include "libm.h"
35
36 #if defined(__sparc)
37 int
38 isnanl(long double x) {
39     int *px = (int *) &x;
40     return ((px[0] & ~0x80000000) >= 0x7fff0000 &&
41         ((px[0] & ~0xffff0000) | px[1] | px[2] | px[3]) != 0);
42 }
43
44 unchanged_portion_omitted
```

new/usr/src/lib/libm/common/LD/jnl.c

1

```
*****
6885 Mon Oct 27 12:36:46 2014
new/usr/src/lib/libm/common/LD/jnl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak jnl = __jnl
31 #pragma weak ynl = __ynl
33 #endif
34 /*
35 * floating point Bessel's function of the 1st and 2nd kind
36 * of order n: jn(n,x),yn(n,x);
37 *
38 * Special cases:
39 * y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
40 * y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
41 * Note 2. About jn(n,x), yn(n,x)
42 * For n=0, j0(x) is called,
43 * for n=1, j1(x) is called,
44 * for n>1, forward recursion is used starting
45 * from values of j0(x) and j1(x).
46 * for n>x, a continued fraction approximation to
47 * j(n,x)/j(n-1,x) is evaluated and then backward
48 * recursion is used starting from a supposed value
49 * for j(n,x). The resulting value of j(0,x) is
50 * compared with the actual value to correct the
51 * supposed value of j(n,x).
52 *
53 * yn(n,x) is similar in all respects, except
54 * that forward recursion is used for all
55 * values of n>1.
56 */
58 #include "libm.h"
```

new/usr/src/lib/libm/common/LD/jnl.c

2

```
59 #include "longdouble.h"
60 #include <float.h> /* LDBL_MAX */
62 #define GENERIC long double
64 static const GENERIC
65 invsqrtpi = 5.641895835477562869480794515607725858441e-0001L,
66 two = 2.0L,
67 zero = 0.0L,
68 one = 1.0L;
70 GENERIC
71 jnl(n, x) int n; GENERIC x; {
72     int i, sgn;
73     GENERIC a, b, temp = 0, z, w;
75     /*
76      * J(-n,x) = (-1)^n * J(n, x), J(n, -x) = (-1)^n * J(n, x)
77      * Thus, J(-n,x) = J(n,-x)
78      */
79     if (n < 0) {
80         n = -n;
81         x = -x;
82     }
83     if (n == 0) return (j0l(x));
84     if (n == 1) return (j1l(x));
85     if (x != x) return x+x;
86     if ((n&1) == 0)
87         sgn = 0; /* even n */
88     else
89         sgn = signbitl(x); /* old n */
90     x = fabsl(x);
91     if (x == zero || !finitel(x)) b = zero;
92     else if ((GENERIC)n <= x) {
93         /*
94          * Safe to use
95          * J(n+1,x)=2n/x *J(n,x)-J(n-1,x)
96          */
97         if (x > 1.0e91L) {
98             /*
99              * x >> n**2
100             * Jn(x) = cos(x-(2n+1)*pi/4)*sqrt(2/x*pi)
101             * Yn(x) = sin(x-(2n+1)*pi/4)*sqrt(2/x*pi)
102             * Let s=sin(x), c=cos(x),
103             * xn=x-(2n+1)*pi/4, sqt2 = sqrt(2), then
104             *
105             *          n      sin(xn)*sqt2      cos(xn)*sqt2
106             * -----
107             *          0      s-c              c+s
108             *          1     -s-c              -c+s
109             *          2     -s+c              -c-s
110             *          3      s+c              c-s
111             */
112             switch (n&3) {
113                 case 0: temp = cosl(x)+sinl(x); break;
114                 case 1: temp = -cosl(x)+sinl(x); break;
115                 case 2: temp = -cosl(x)-sinl(x); break;
116                 case 3: temp = cosl(x)-sinl(x); break;
117             }
118             b = invsqrtpi*temp/sqrtl(x);
119         } else {
120             a = j0l(x);
121             b = j1l(x);
122             for (i = 1; i < n; i++) {
123                 temp = b;
124                 b = b*((GENERIC)(i+1)/x) - a; /* avoid underflow */
```

```

125         a = temp;
126     }
127 } else {
128     if (x < 1e-17L) { /* use J(n,x) = 1/n!(x/2)^n */
129         b = powl(0.5L*x, (GENERIC) n);
130         if (b != zero) {
131             for (a = one, i = 1; i <= n; i++) a *= (GENERIC)i;
132             b = b/a;
133         }
134     } else {
135         /*
136         * use backward recurrence
137         *
138         * J(n,x)/J(n-1,x) =  $\frac{x}{2n} - \frac{x^2}{2(n+1)} - \frac{x^2}{2(n+2)} - \dots$ 
139         *
140         * (for large x) =  $\frac{1}{2n} - \frac{1}{2(n+1)} - \frac{1}{2(n+2)} - \dots$ 
141         *
142         *
143         *
144         *
145         *
146         *
147         * Let w = 2n/x and h=2/x, then the above quotient
148         * is equal to the continued fraction:
149         *
150         *  $\frac{1}{w - \frac{1}{w+h - \frac{1}{w+2h - \dots}}}$ 
151         *
152         *
153         *
154         *
155         *
156         *
157         * To determine how many terms needed, let
158         * Q(0) = w, Q(1) = w(w+h) - 1,
159         * Q(k) = (w+k*h)*Q(k-1) - Q(k-2),
160         * When Q(k) > 1e4 good for single
161         * When Q(k) > 1e9 good for double
162         * When Q(k) > 1e17 good for quaduple
163         */
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(fabs(v*tmp));
192     if (tmp < 1.1356523406294143949491931077970765e+04L) {
193         for (i = n-1; i > 0; i--) {
194             temp = b;
195             b = ((i+1)/x)*b - a;
196             a = temp;
197         }
198     } else {
199         for (i = n-1; i > 0; i--) {
200             temp = b;
201             b = ((i+1)/x)*b - a;
202             a = temp;
203         }
204         if (b > 1e1000L) {
205             a /= b;
206             t /= b;
207             b = 1.0;
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

new/usr/src/lib/libm/common/LD/loglpl.c

1

```
*****
1592 Mon Oct 27 12:36:46 2014
new/usr/src/lib/libm/common/LD/loglpl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak loglpl = __loglpl
32 #endif

32 /*
33 * loglpl(x)
34 * Kahan's trick based on log(1+x)/x being a slow varying function.
35 */

37 #include "libm.h"

39 #if defined(__x86)
40 #define __swapRD __swap87RD
41 #endif
42 extern enum fp_direction_type __swapRD(enum fp_direction_type);

44 long double
45 loglpl(long double x) {
46     long double y;
47     enum fp_direction_type rd;

49     if (x != x)
50         return (x + x);
51     if (x < -1.L)
52         return (logl(x));
53     rd = __swapRD(fp_nearest);
54     y = 1.L + x;
55     if (y != 1.L) {
56         if (y == x)
57             x = logl(x);
58         else
```

new/usr/src/lib/libm/common/LD/loglpl.c

2

```
59         x *= logl(y) / (y - 1.L);
60     }
61     if (rd != fp_nearest)
62         (void) __swapRD(rd);
63     return (x);
64 }
unchanged_portion_omitted
```

new/usr/src/lib/libm/common/LD/logbl.c

1

```
*****
2468 Mon Oct 27 12:36:47 2014
new/usr/src/lib/libm/common/LD/logbl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak logbl = __logbl
32 #endif

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)
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) {
```

new/usr/src/lib/libm/common/LD/logbl.c

2

```
59 #pragma STDC FENV_ACCESS ON
60 static const long double zero = 0.0L;
61 return (v / zero);
62 }
_____unchanged_portion_omitted_____
```



```

*****
2732 Mon Oct 27 12:36:47 2014
new/usr/src/lib/libm/common/LD/nextafterl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nextafterl = __nextafterl
32 #endif

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

35 #if defined(__sparc)
36 #define n0      0
37 #define n1      1
38 #define n2      2
39 #define n3      3
40 #define X86PDNRM1(x)
41 #define INC(px) { \
42     if (++px[n3] == 0) \
43         if (++px[n2] == 0) \
44             if (++px[n1] == 0) \
45                 ++px[n0]; \
46 }
47 #define DEC(px) { \
48     if (--px[n3] == 0xffffffff) \
49         if (--px[n2] == 0xffffffff) \
50             if (--px[n1] == 0xffffffff) \
51                 --px[n0]; \
52 }
53 #elif defined(__x86)
54 #define n0      2
55 #define n1      1
56 #define n2      0
57 #define n3      0
58 /*

```

```

59  * if pseudo-denormal, replace by the equivalent normal
60  */
61 #define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
62     0x80000000) != 0) \
63     ((int *) &x)[2] |= 1
64 #define INC(px) { \
65     if (++px[n2] == 0) \
66         if (++px[n1] & ~0x80000000) == 0) \
67             px[n1] = 0x80000000, ++px[n0]; \
68 }
69 #define DEC(px) { \
70     if (--px[n2] == 0xffffffff) \
71         if (--px[n1] == 0x7fffffff) \
72             if (--px[n0] & 0x7fff) != 0) \
73                 px[n1] |= 0x80000000; \
74 }
75 #endif

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

82     if (x == y)
83         return (y);          /* C99 requirement */
84     if (x != x || y != y)
85         return (x * y);

87     if (ISZEROL(x)) {        /* x == 0.0 */
88         px[n0] = py[n0] & XSGNMSK;
89         px[n1] = px[n2] = 0;
90         px[n3] = 1;
91     } else {
92         X86PDNRM1(x);
93         if ((px[n0] & XSGNMSK) == 0) { /* x > 0.0 */
94             if (x > y) /* x > y */
95                 DEC(px)
96             else
97                 INC(px)
98         } else {
99             if (x < y) /* x < y */
100                 DEC(px)
101             else
102                 INC(px)
103         }
104     }
105 #ifndef lint
106     {
107         volatile long double dummy;
108         int k = XBIASED_EXP(x);

110         if (k == 0)
111             dummy = LDBL_MIN * copysign(LDBL_MIN, x);
112         else if (k == 0x7fff)
113             dummy = LDBL_MAX * copysign(LDBL_MAX, x);
114     }
115 #endif
116     return (x);
117 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/LD/significandl.c

1

```
*****
1226 Mon Oct 27 12:36:48 2014
new/usr/src/lib/libm/common/LD/significandl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak significandl = __significandl
32 #endif

32 #include "libm.h"

34 long double
35 significandl(long double x) {
36     if (ISZEROL(x) || XBIASED_EXP(x) == 0x7fff) /* 0/+Inf/NaN */
37         return (x + x);
38     else
39         return (scalbnl(x, -ilogbl(x)));
40 }
unchanged_portion_omitted
```

```

*****
2601 Mon Oct 27 12:36:48 2014
new/usr/src/lib/libm/common/LD/tanh1.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak tanhl = __tanhl
32 #endif

32 /*
33 * tanhl(x) returns the hyperbolic tangent of x
34 *
35 * Method :
36 * 1. reduce x to non-negative: tanhl(-x) = - tanhl(x).
37 * 2.
38 * 0 < x <= small : tanhl(x) := x
39 * -expm1(-2x)
40 * small < x <= 1 : tanhl(x) := -----
41 * expm1(-2x) + 2
42 * 2
43 * 1 <= x <= threshold : tanhl(x) := 1 - -----
44 * expm1(2x) + 2
45 * threshold < x <= INF : tanhl(x) := 1.
46 *
47 * where
48 * single : small = 1.e-5 threshold = 11.0
49 * double : small = 1.e-10 threshold = 22.0
50 * quad : small = 1.e-20 threshold = 45.0
51 *
52 * Note: threshold was chosen so that
53 * f1(1.0+2/(expm1(2*threshold)+2)) == 1.
54 *
55 * Special cases:
56 * tanhl(NaN) is NaN;
57 * only tanhl(0.0)=0.0 is exact for finite argument.
58 */

```

```

60 #include "libm.h"
61 #include "longdouble.h"

63 static const long double small = 1.0e-20L, one = 1.0, two = 2.0,
64 #ifndef lint
65     big = 1.0e+20L,
66 #endif
67     threshold = 45.0L;

69 long double
70 tanhl(long double x) {
71     long double t, y, z;
72     int signx;
73 #ifndef lint
74     volatile long double dummy;
75 #endif

77     if (isnanl(x))
78         return (x + x);          /* x is NaN */
79     signx = signbitl(x);
80     t = fabsl(x);
81     z = one;
82     if (t <= threshold) {
83         if (t > one)
84             z = one - two / (expm1(t + t) + two);
85         else if (t > small) {
86             y = expm1(-t - t);
87             z = -y / (y + two);
88         } else {
89 #ifndef lint
90             dummy = t + big;
91                                     /* inexact if t != 0 */
92 #endif
93             return (x);
94         }
95     } else if (!finitel(t))
96         return (copysignl(one, x));
97     else
98         return (signx ? -z + small * small : z - small * small);
99     return (signx ? -z : z);
100 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/Q/acoshl.c

1

\*\*\*\*\*

1471 Mon Oct 27 12:36:48 2014

new/usr/src/lib/libm/common/Q/acoshl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 #if defined(ELFOBJ)
30 #pragma weak acoshl = __acoshl
32 #endif

32 #include "libm.h"

34 static const long double
35     zero      = 0.0L,
36     ln2       = 6.931471805599453094172321214581765680755e-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 (loglpl(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_
```

new/usr/src/lib/libm/common/Q/asinhl.c

1

```
*****
1590 Mon Oct 27 12:36:49 2014
new/usr/src/lib/libm/common/Q/asinhl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak asinhl = __asinhl
32 #endif

32 #include "libm.h"

34 static const long double
35     ln2      = 6.931471805599453094172321214581765680755e-0001L,
36     one      = 1.0L,
37     big      = 1.0e+20L,
38     tiny     = 1.0e-20L;

40 long double
41 asinhl(long double x) {
42     long double t, w;
43     volatile long double dummy;

45     w = fabsl(x);
46     if (isnanl(x))
47         return (x + x); /* x is NaN */
48     if (w < tiny) {
49 #ifndef lint
50         dummy = x + big; /* inexact if x != 0 */
51 #endif
52         return (x); /* tiny x */
53     } else if (w < big) {
54         t = one / w;
55         return (copysignl(loglpl(w + w / (t + sqrtl(one + t * t))), x));
56     } else
57         return (copysignl(logl(w) + ln2, x));
58 }
```

new/usr/src/lib/libm/common/Q/asinl.c

1

```
*****
2010 Mon Oct 27 12:36:49 2014
new/usr/src/lib/libm/common/Q/asinl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak asinl = __asinl
32 #endif

32 /*
33 * asinl(x) = atan2l(x,sqrt(1-x*x));
34 *
35 * For better accuracy, 1-x*x is computed as follows
36 * 1-x*x if x < 0.5,
37 * 2*(1-|x|)-(1-|x|)*(1-|x|) if x >= 0.5.
38 *
39 * Special cases:
40 * if x is NaN, return x itself;
41 * if |x|>1, return NaN with invalid signal.
42 */

44 #include "libm.h"

46 static const long double zero = 0.0L, small = 1.0e-20L, half = 0.5L, one = 1.0L;
47 #ifndef lint
48 static const long double big = 1.0e+20L;
49 #endif

51 long double
52 asinl(long double x) {
53     long double t, w;
54     volatile long double dummy;

56     w = fabsl(x);
57     if (isnanl(x))
58         return (x + x);
```

new/usr/src/lib/libm/common/Q/asinl.c

2

```
59     else if (w <= half) {
60         if (w < small) {
61             #ifndef lint
62                 dummy = w + big;
63             #endif
64             #endif
65             return (x);
66         } else
67             return (atanl(x / sqrtl(one - x * x)));
68     } else if (w < one) {
69         t = one - w;
70         w = t + t;
71         return (atanl(x / sqrtl(w - t * t)));
72     } else if (w == one)
73         return (atan2l(x, zero)); /* asin(++1) = +- PI/2 */
74     else
75         return (zero / zero); /* |x| > 1: invalid */
76 }
unchanged_portion_omitted
```

new/usr/src/lib/libm/common/Q/copysignl.c

1

```
*****
1237 Mon Oct 27 12:36:50 2014
new/usr/src/lib/libm/common/Q/copysignl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak copysignl = __copysignl
32 #endif

32 #include "libm.h"

34 long double
35 copysignl(long double x, long double y) {
36     int *px = (int *) &x;
37     int *py = (int *) &y;

39     px[HIXWORD] = (px[HIXWORD] & ~XSGNMSK) | (py[HIXWORD] & XSGNMSK);
40     return (x);
41 }
_____unchanged_portion_omitted_____
```

```
*****
5996 Mon Oct 27 12:36:50 2014
new/usr/src/lib/libm/common/Q/expmll.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak expmll = __expmll
32 #endif
33 #if !defined(__sparc)
34 #error Unsupported architecture
35 #endif
36
37 /*
38 * expmll(x)
39 *
40 * Table driven method
41 * Written by K.C. Ng, June 1995.
42 * Algorithm :
43 * 1. expmll(x) = x if x<2**-114
44 * 2. if |x| <= 0.0625 = 1/16, use approximation
45 * expmll(x) = x + x*P/(2-P)
46 * where
47 * P = x - z*(P1+z*(P2+z*(P3+z*(P4+z*(P5+z*(P6+z*P7))))), z = x*x;
48 * (this formula is derived from
49 * 2-P+x = R = x*(exp(x)+1)/(exp(x)-1) ~ 2 + x*x/6 - x^4/360 + ...)
50 *
51 * P1 = 1.6666666666666666666666666666666666666666666666638500528074603030e-0001
52 * P2 = -2.7777777777777777777777777777777777777777777777759668391122822266551158e-0003
53 * P3 = 6.61375661375661375657437408890138814721051293054e-0005
54 * P4 = -1.65343915343915303310185228411892601606669528828e-0006
55 * P5 = 4.17535139755122945763580609663414647067443411178e-0008
56 * P6 = -1.05683795988668526689182102605260986731620026832e-0009
57 * P7 = 2.67544168821852702827123344217198187229611470514e-0011
58 *
59 * Accuracy: |R-x*(exp(x)+1)/(exp(x)-1)|<=2**-119.13

```

```
59 *
60 * 3. For 1/16 < |x| < 1.125, choose x(-i) ~ +/- (i+4.5)/64, i=0,..,67
61 * since
62 * exp(x) = exp(xi+(x-xi))= exp(xi)*exp((x-xi))
63 * we have
64 * expmll(x) = expmll(xi)+(exp(xi))*(expmll(x-xi))
65 * where
66 * |s=x-xi| <= 1/128
67 * and
68 * expmll(s)=2s/(2-R), R= s-s^2*(T1+s^2*(T2+s^2*(T3+s^2*(T4+s^2*T5)))
69 *
70 * T1 = 1.666666666666666666666666666666666666666666666660876387437e-1L,
71 * T2 = -2.7777777777777777777777777777777777777777777777707812093173478756e-3L,
72 * T3 = 6.613756613756613482074280932874221202424e-5L,
73 * T4 = -1.6534391533921399541696098227422235851120e-6L,
74 * T5 = 4.175314851769539751387852116610973796053e-8L;
75 *
76 * 4. For |x| >= 1.125, return exp(x)-1.
77 * (see algorithm for exp)
78 *
79 * Special cases:
80 * expmll(INF) is INF, expmll(NaN) is NaN;
81 * expmll(-INF)= -L;
82 * for finite argument, only expmll(0)=0 is exact.
83 *
84 * Accuracy:
85 * according to an error analysis, the error is always less than
86 * 2 ulp (unit in the last place).
87 *
88 * Misc. info.
89 * For 113 bit long double
90 * if x > 1.135652340629414394949193107797076342845e+4
91 * then expmll(x) overflow;
92 *
93 * Constants:
94 * Only decimal values are given. We assume that the compiler will convert
95 * from decimal to binary accurately enough to produce the correct
96 * hexadecimal values.
97 *
98 #include "libm.h"
99
100 extern const long double _TBL_expl_hi[], _TBL_expl_lo[];
101 extern const long double _TBL_expmlx[], _TBL_expml[];
102
103 static const long double
104 zero = +0.0L,
105 one = +1.0L,
106 two = +2.0L,
107 ln2_64 = +1.083042469624914545964425189778400898568e-2L,
108 ovflthresold = +1.135652340629414394949193107797076342845e+4L,
109 invln2_32 = +4.616624130844682903551758979206054839765e+1L,
110 ln2_32hi = +2.166084939249829091928849858592451515688e-2L,
111 ln2_32lo = +5.209643502595475652782654157501186731779e-27L,
112 huge = +1.0e4000L,
113 tiny = +1.0e-4000L,
114 P1 = +1.6666666666666666666666666666666666666666666666666666638500528074603030e-0001L,
115 P2 = -2.77777777777777777777777777777777777777777777777777777777777759668391122822266551158e-0003L,
116 P3 = +6.61375661375661375657437408890138814721051293054e-0005L,
117 P4 = -1.65343915343915303310185228411892601606669528828e-0006L,
118 P5 = +4.17535139755122945763580609663414647067443411178e-0008L,
119 P6 = -1.05683795988668526689182102605260986731620026832e-0009L,
120 P7 = +2.67544168821852702827123344217198187229611470514e-0011L,
121 /* rational approximation coeffs for [-(ln2)/64,(ln2)/64] */
122 T1 = +1.6666666666666666666666666666666666666666666666666666666666666666666666666666660876387437e-1L,
123 T2 = -2.7777777777777777777777777777777777777777777777707812093173478756e-3L,
124 T3 = +6.613756613756613482074280932874221202424e-5L,
```



```

125     T4 = -1.653439153392139954169609822742235851120e-6L,
126     T5 = +4.175314851769539751387852116610973796053e-8L;

128 long double
129 expm1(long double x) {
130     int hx, ix, j, k, m;
131     long double t, r, s, w;

133     hx = ((int *) &x)[HIXWORD];
134     ix = hx & ~0x80000000;
135     if (ix >= 0x7fff0000) {
136         if (x != x)
137             return (x + x); /* NaN */
138         if (x < zero)
139             return (-one); /* -inf */
140         return (x); /* +inf */
141     }
142     if (ix < 0x3fff4000) { /* |x| < 1.25 */
143         if (ix < 0x3ffb0000) { /* |x| < 0.0625 */
144             if (ix < 0x3f8d0000) {
145                 if ((int) x == 0)
146                     return (x); /* |x| < 2^-114 */
147             }
148             t = x * x;
149             r = (x - t * (P1 + t * (P2 + t * (P3 + t * (P4 + t *
150                 (P5 + t * (P6 + t * P7))))));
151             return (x + (x * r) / (two - r));
152         }
153         /* compute i = [64*x] */
154         m = 0x4009 - (ix >> 16);
155         j = ((ix & 0x0000ffff) | 0x10000) >> m; /* j=4,...,67 */
156         if (hx < 0)
157             j += 82; /* negative */
158         s = x - _TBL_expml1x[j];
159         t = s * s;
160         r = s - t * (T1 + t * (T2 + t * (T3 + t * (T4 + t * T5)));
161         r = (s + s) / (two - r);
162         w = _TBL_expml1[j];
163         return (w + (w + one) * r);
164     }
165     if (hx > 0) {
166         if (x > ovflthreshold)
167             return (huge * huge);
168         k = (int) (invln2_32 * (x + ln2_64));
169     } else {
170         if (x < -80.0)
171             return (tiny - x / x);
172         k = (int) (invln2_32 * (x - ln2_64));
173     }
174     j = k & 0x1f;
175     m = k >> 5;
176     t = (long double) k;
177     x = (x - t * ln2_32hi) - t * ln2_32lo;
178     t = x * x;
179     r = (x - t * (T1 + t * (T2 + t * (T3 + t * (T4 + t * T5)))) - two;
180     x = _TBL_expl_hi[j] - ((_TBL_expl_hi[j] * (x + x)) / r -
181         _TBL_expl_lo[j]);
182     return (scalbnl(x, m) - one);
183 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/Q/finitel.c

1

\*\*\*\*\*

1446 Mon Oct 27 12:36:51 2014

new/usr/src/lib/libm/common/Q/finitel.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 #if defined(ELFOBJ)
30 #pragma weak finitel = __finitel
32 #endif
```

```
32 #include "libm.h"
```

```
34 #if defined(__sparc)
35 int
36 finitel(long double x) {
37     int *px = (int *) &x;
38     return ((px[0] & ~0x80000000) < 0x7fff0000);
39 }
```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/Q/ieee\_func1.c

1

```
*****
2732 Mon Oct 27 12:36:51 2014
new/usr/src/lib/libm/common/Q/ieee_func1.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELF_OBJ)
30 #pragma weak isinfl = __isinfl
31 #pragma weak isnormall = __isnormall
32 #pragma weak issubnormall = __issubnormall
33 #pragma weak iszerol = __iszerol
34 #pragma weak signbitl = __signbitl
36 #endif

36 #include "libm.h"

38 #if defined(__sparc)
39 int
40 isinfl(long double x) {
41     int *px = (int *) &x;
42     return ((px[0] & ~0x80000000) == 0x7fff0000 && px[1] == 0 &&
43            px[2] == 0 && px[3] == 0);
44 }
  unchanged_portion_omitted_

```

new/usr/src/lib/libm/common/Q/ilogbl.c

1

```
*****
2452 Mon Oct 27 12:36:51 2014
new/usr/src/lib/libm/common/Q/ilogbl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak ilogbl = __ilogbl
32 #endif

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

35 #if defined(__sparc)
36 #define ISNORMALL(k, x) (k != 0x7fff) /* assuming k != 0 */
37 #define X86PDNRM(k, x)
38 #define XSCALE_OFFSET 0x406f /* 0x3fff + 112 */
39 static const long double xscale = 5192296858534827628530496329220096.0L;
40 /* 2^112 */
41 #elif defined(__x86)
42 /*
43 * if pseudo-denormal, replace by the equivalent normal
44 */
45 #define X86PDNRM(k, x) if (k == 0 && (((int *) &x)[1] & 0x80000000) != 0) \
46 ((int *) &x)[2] |= k = 1
47 #if defined(HANDLE_UNSUPPORTED)
48 #define ISNORMALL(k, x) (k != 0x7fff && (((int *) &x)[1] & 0x80000000) != 0)
49 #else
50 #define ISNORMALL(k, x) (k != 0x7fff)
51 #endif
52 #define XSCALE_OFFSET 0x403e /* 0x3fff + 63 */
53 static const long double xscale = 9223372036854775808.0L; /* 2^63 */
54 #endif

56 static int
57 raise_invalid(int v) { /* SUSv3 requires ilogbl(0,+/-Inf,NaN) raise invalid */
58 #ifndef lint
```

new/usr/src/lib/libm/common/Q/ilogbl.c

2

```
59 if ((__xpg6 & _C99SUSv3_ilogb_0InfNaN_raises_invalid) != 0) {
60     static const double zero = 0.0;
61     volatile double dummy;

63     dummy = zero / zero;
64 }
65 #endif
66 return (v);
67 }
_____unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/Q/isnanl.c

1

\*\*\*\*\*

1581 Mon Oct 27 12:36:52 2014

new/usr/src/lib/libm/common/Q/isnanl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak isnanl = __isnanl
32 #endif
33
34 #include "libm.h"
35
36 #if defined(__sparc)
37 int
38 isnanl(long double x) {
39     int *px = (int *) &x;
40     return ((px[0] & ~0x80000000) >= 0x7fff0000 &&
41             ((px[0] & ~0xffff0000) | px[1] | px[2] | px[3]) != 0);
42 }
43
44 unchanged_portion_omitted
```

new/usr/src/lib/libm/common/Q/jnl.c

1

```
*****
7136 Mon Oct 27 12:36:52 2014
new/usr/src/lib/libm/common/Q/jnl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak jnl = __jnl
31 #pragma weak ynl = __ynl
33 #endif
34 /*
35 * floating point Bessel's function of the 1st and 2nd kind
36 * of order n: jn(n,x),yn(n,x);
37 *
38 * Special cases:
39 *   y0(0)=y1(0)=yn(n,0) = -inf with division by zero signal;
40 *   y0(-ve)=y1(-ve)=yn(n,-ve) are NaN with invalid signal.
41 * Note 2. About jn(n,x), yn(n,x)
42 *   For n=0, j0(x) is called,
43 *   for n=1, j1(x) is called,
44 *   for n<x, forward recursion us used starting
45 *   from values of j0(x) and j1(x).
46 *   for n>x, a continued fraction approximation to
47 *   j(n,x)/j(n-1,x) is evaluated and then backward
48 *   recursion is used starting from a supposed value
49 *   for j(n,x). The resulting value of j(0,x) is
50 *   compared with the actual value to correct the
51 *   supposed value of j(n,x).
52 *   yn(n,x) is similar in all respects, except
53 *   that forward recursion is used for all
54 *   values of n>1.
55 *
56 */
58 #include "libm.h"
```

new/usr/src/lib/libm/common/Q/jnl.c

2

```
59 #include "longdouble.h"
60 #include <float.h> /* LDBL_MAX */
62 #define GENERIC long double
64 static const GENERIC
65 invsqrtpi = 5.641895835477562869480794515607725858441e-0001L,
66 two = 2.0L,
67 zero = 0.0L,
68 one = 1.0L;
70 GENERIC
71 jnl(n, x) int n; GENERIC x; {
72     int i, sgn;
73     GENERIC a, b, temp, z, w;
74
75     /*
76      * J(-n,x) = (-1)^n * J(n, x), J(n, -x) = (-1)^n * J(n, x)
77      * Thus, J(-n,x) = J(n,-x)
78      */
79     if (n < 0) {
80         n = -n;
81         x = -x;
82     }
83     if (n == 0)
84         return (j0l(x));
85     if (n == 1)
86         return (j1l(x));
87     if (x != x)
88         return (x+x);
89     if ((n&1) == 0)
90         sgn = 0; /* even n */
91     else
92         sgn = signbitl(x); /* old n */
93     x = fabsl(x);
94     if (x == zero || !finitel(x)) b = zero;
95     else if ((GENERIC)n <= x) {
96         /*
97          * Safe to use
98          * J(n+1,x)=2n/x *J(n,x)-J(n-1,x)
99          */
100         if (x > 1.0e91L) {
101             /*
102              * x >> n**2
103              * Jn(x) = cos(x-(2n+1)*pi/4)*sqrt(2/x*pi)
104              * Yn(x) = sin(x-(2n+1)*pi/4)*sqrt(2/x*pi)
105              * Let s=sin(x), c=cos(x),
106              * xn=x-(2n+1)*pi/4, sqrt2 = sqrt(2), then
107              *
108              *          n      sin(xn)*sqrt2      cos(xn)*sqrt2
109              *          -----
110              *          0      s-c                  c+s
111              *          1      -s-c                 -c+s
112              *          2      -s+c                 -c-s
113              *          3      s+c                  c-s
114              */
115             switch (n&3) {
116                 case 0: temp = cosl(x)+sinl(x); break;
117                 case 1: temp = -cosl(x)+sinl(x); break;
118                 case 2: temp = -cosl(x)-sinl(x); break;
119                 case 3: temp = cosl(x)-sinl(x); break;
120             }
121             b = invsqrtpi*temp/sqrtl(x);
122         } else {
123             a = j0l(x);
124             b = j1l(x);
```

```

125         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          *
142          * 
$$J(n,x)/J(n-1,x) = \frac{x}{2n} - \frac{x^2}{2(n+1)} - \frac{x^2}{2(n+2)} \dots$$

143          *
144          *
145          * (for large x) =  $\frac{1}{2n} - \frac{1}{2(n+1)} - \frac{1}{2(n+2)} \dots$ 
146          *
147          *
148          * 
$$\frac{1}{x} - \frac{1}{x} - \frac{1}{x} - \dots$$

149          *
150          *
151          * Let w = 2n/x and h=2/x, then the above quotient
152          * is equal to the continued fraction:
153          *
154          * 
$$= \frac{1}{w - \frac{1}{w+h - \frac{1}{w+2h - \dots}}}$$

155          *
156          *
157          *
158          *
159          *
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 1.1356523406294143949491931077970765006170e+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 < 1.1356523406294143949491931077970765e+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

new/usr/src/lib/libm/common/Q/log101.c

1

```
*****
3044 Mon Oct 27 12:36:53 2014
new/usr/src/lib/libm/common/Q/log101.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak log10l = __log10l
32 #endif

32 /*
33 * log10l(X)
34 *
35 * Method :
36 * Let log10_2hi = leading 98(SPARC)/49(x86) bits of log10(2) and
37 * log10_2lo = log10(2) - log10_2hi,
38 * ivln10 = 1/log(10) rounded.
39 * Then
40 * n = ilogb(x),
41 * if (n<0) n = n+1;
42 * x = scalbn(x,-n);
43 * LOG10(x) := n*log10_2hi + (n*log10_2lo + ivln10*log(x))
44 *
45 * Note1:
46 * For fear of destroying log10(10**n)=n, the rounding mode is
47 * set to Round-to-Nearest.
48 *
49 * Special cases:
50 * log10(x) is NaN with signal if x < 0;
51 * log10(+INF) is +INF with no signal; log10(0) is -INF with signal;
52 * log10(NaN) is that NaN with no signal;
53 * log10(10**N) = N for N=0,1,...,22.
54 *
55 * Constants:
56 * The hexadecimal values are the intended ones for the following constants.
57 * The decimal values may be used, provided that the compiler will convert
58 * from decimal to binary accurately enough to produce the hexadecimal values
```

new/usr/src/lib/libm/common/Q/log101.c

2

```
59 * shown.
60 */

62 #include "libm.h"
63 #include "longdouble.h"

65 #if defined(__x86)
66 #define __swapRD __swap87RD
67 #endif
68 extern enum fp_direction_type __swapRD(enum fp_direction_type);

70 static const long double
71 zero = 0.0L,
72 ivln10 = 4.342944819032518276511289189166050822944e-0001L,
73 one = 1.0L,
74 #if defined(__x86)
75 log10_2hi = 3.010299956639803653501985536422580480576e-01L,
76 log10_2lo = 8.298635403410822349787106337291183585413e-16L;
77 #elif defined(__sparc)
78 log10_2hi = 3.010299956639811952137388947242098603469e-01L,
79 log10_2lo = 2.831664213089468167896664371953210945664e-31L;
80 #else
81 #error Unknown Architecture!
82 #endif

84 long double
85 log10l(long double x) {
86     long double y, z;
87     enum fp_direction_type rd;
88     int n;

90     if (!finitel(x))
91         return (x + fabsl(x)); /* x is +-INF or NaN */
92     else if (x > zero) {
93         n = ilogbl(x);
94         if (n < 0)
95             n += 1;
96         rd = __swapRD(fp_nearest);
97         y = n;
98         x = scalbnl(x, -n);
99         z = y * log10_2lo + ivln10 * logl(x);
100        z += y * log10_2hi;
101        if (rd != fp_nearest)
102            (void) __swapRD(rd);
103        return (z);
104    } else if (x == zero) /* -INF */
105        return (-one / zero);
106    else /* x < 0, return NaN */
107        return (zero / zero);
108 }

_____unchanged_portion_omitted_____
```



new/usr/src/lib/libm/common/Q/logbl.c

1

```
*****
2454 Mon Oct 27 12:36:53 2014
new/usr/src/lib/libm/common/Q/logbl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak logbl = __logbl
32 #endif

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)
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) {
```

new/usr/src/lib/libm/common/Q/logbl.c

2

```
59 #pragma STDC FENV_ACCESS ON
60 static const long double zero = 0.0L;
61 return (v / zero);
62 }
_____unchanged_portion_omitted_____
```

```

*****
2732 Mon Oct 27 12:36:53 2014
new/usr/src/lib/libm/common/Q/nextafter1.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nextafter1 = __nextafter1
32 #endif

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

35 #if defined(__sparc)
36 #define n0      0
37 #define n1      1
38 #define n2      2
39 #define n3      3
40 #define X86PDNRM1(x)
41 #define INC(px) { \
42     if (++px[n3] == 0) \
43         if (++px[n2] == 0) \
44             if (++px[n1] == 0) \
45                 ++px[n0]; \
46 }
47 #define DEC(px) { \
48     if (--px[n3] == 0xffffffff) \
49         if (--px[n2] == 0xffffffff) \
50             if (--px[n1] == 0xffffffff) \
51                 --px[n0]; \
52 }
53 #elif defined(__x86)
54 #define n0      2
55 #define n1      1
56 #define n2      0
57 #define n3      0
58 /*

```

```

59 * if pseudo-denormal, replace by the equivalent normal
60 */
61 #define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
62     0x80000000) != 0) \
63     ((int *) &x)[2] |= 1
64 #define INC(px) { \
65     if (++px[n2] == 0) \
66         if (++px[n1] & ~0x80000000) == 0) \
67             px[n1] = 0x80000000, ++px[n0]; \
68 }
69 #define DEC(px) { \
70     if (--px[n2] == 0xffffffff) \
71         if (--px[n1] == 0x7fffffff) \
72             if (--px[n0] & 0x7fff) != 0) \
73                 px[n1] |= 0x80000000; \
74 }
75 #endif

77 long double
78 nextafter1(long double x, long double y) {
79     int *px = (int *) &x;
80     int *py = (int *) &y;

82     if (x == y)
83         return (y);          /* C99 requirement */
84     if (x != x || y != y)
85         return (x * y);

87     if (ISZEROL(x)) {        /* x == 0.0 */
88         px[n0] = py[n0] & XSGNMSK;
89         px[n1] = px[n2] = 0;
90         px[n3] = 1;
91     } else {
92         X86PDNRM1(x);
93         if ((px[n0] & XSGNMSK) == 0) { /* x > 0.0 */
94             if (x > y) /* x > y */
95                 DEC(px)
96             else
97                 INC(px)
98         } else {
99             if (x < y) /* x < y */
100                 DEC(px)
101             else
102                 INC(px)
103         }
104     }
105 #ifndef lint
106     {
107         volatile long double dummy;
108         int k = XBIASED_EXP(x);

110         if (k == 0)
111             dummy = LDBL_MIN * copysign(LDBL_MIN, x);
112         else if (k == 0x7fff)
113             dummy = LDBL_MAX * copysign(LDBL_MAX, x);
114     }
115 #endif
116     return (x);
117 }

```

unchanged\_portion\_omitted



new/usr/src/lib/libm/common/Q/scalbnl.c

1

```
*****
2490 Mon Oct 27 12:36:54 2014
new/usr/src/lib/libm/common/Q/scalbnl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak scalbnl = __scalbnl
32 #endif

32 #include "libm.h"
33 #include <float.h>          /* LDBL_MAX, LDBL_MIN */
34 #include <stdlib.h>         /* abs */

36 #if defined(__sparc)
37 #define XSET_EXP(k, x)  (((int *) &x)[0] = (((int *) &x)[0] & ~0x7fff0000) | \
38                        (k << 16))
39 #define ISINFNANL(k, x) (k == 0x7fff)
40 #define XTWOT_OFFSET    113
41 static const long double xtwot = 10384593717069655257060992658440192.0L,
42                        twomt1 = 4.814824860968089632639944856462318296E-35L; /* 2^113 */
43                        /* 2^-114 */
44 #elif defined(__x86)
45 #define XSET_EXP(k, x)  (((int *) &x)[2] = (((int *) &x)[2] & ~0x7fff) | k
46 #if defined(HANDLE_UNSUPPORTED)
47 #define ISINFNANL(k, x) (k == 0x7fff || k != 0 && \
48                        (((int *) &x)[1] & 0x80000000) == 0)
49 #else
50 #define ISINFNANL(k, x) (k == 0x7fff)
51 #endif
52 #define XTWOT_OFFSET    64
53 static const long double xtwot = 18446744073709551616.0L, /* 2^64 */
54                        twomt1 = 2.7105054312137610850186E-20L; /* 2^-65 */
55 #endif

57 long double
58 scalbnl(long double x, int n) {
```

new/usr/src/lib/libm/common/Q/scalbnl.c

2

```
59     int k = XBIASED_EXP(x);
61     if (ISINFNANL(k, x))
62         return (x + x);
63     if (ISZEROL(x) || n == 0)
64         return (x);
65     if (k == 0) {
66         x *= xtwot;
67         k = XBIASED_EXP(x) - XTWOT_OFFSET;
68     }
69     if (((unsigned) abs(n)) >= 131072) /* cast to unsigned for -2^31 */
70         n >>= 1; /* avoid subsequent integer overflow */
71     k += n;
72     if (k > 0x7ffe)
73         return (LDBL_MAX * copysignl(LDBL_MAX, x));
74     if (k <= -XTWOT_OFFSET - 1)
75         return (LDBL_MIN * copysignl(LDBL_MIN, x));
76     if (k > 0) {
77         XSET_EXP(k, x);
78         return (x);
79     }
80     k += XTWOT_OFFSET + 1;
81     XSET_EXP(k, x);
82     return (x * twomt1);
83 }
unchanged_portion_omitted
```

new/usr/src/lib/libm/common/Q/significandl.c

1

```
*****
1226 Mon Oct 27 12:36:55 2014
new/usr/src/lib/libm/common/Q/significandl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak significandl = __significandl
32 #endif

32 #include "libm.h"

34 long double
35 significandl(long double x) {
36     if (ISZEROL(x) || XBIASED_EXP(x) == 0x7fff) /* 0/+Inf/NaN */
37         return (x + x);
38     else
39         return (scalbnl(x, -ilogbl(x)));
40 }
unchanged_portion_omitted
```

```

*****
2581 Mon Oct 27 12:36:55 2014
new/usr/src/lib/libm/common/Q/tanh1.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak tanhl = __tanhl
32 #endif

32 /*
33 * tanhl(x) returns the hyperbolic tangent of x
34 *
35 * Method :
36 * 1. reduce x to non-negative: tanhl(-x) = - tanhl(x).
37 * 2.
38 * 0 < x <= small : tanhl(x) := x
39 * -expm1(-2x)
40 * small < x <= 1 : tanhl(x) := -----
41 * expm1(-2x) + 2
42 * 2
43 * 1 <= x <= threshold : tanhl(x) := 1 - -----
44 * expm1(2x) + 2
45 * threshold < x <= INF : tanhl(x) := 1.
46 *
47 * where
48 * single : small = 1.e-5 threshold = 11.0
49 * double : small = 1.e-10 threshold = 22.0
50 * quad : small = 1.e-20 threshold = 45.0
51 *
52 * Note: threshold was chosen so that
53 * fl(1.0+2/(expm1(2*threshold)+2)) == 1.
54 *
55 * Special cases:
56 * tanhl(NaN) is NaN;
57 * only tanhl(0.0)=0.0 is exact for finite argument.
58 */

```

```

60 #include "libm.h"
61 #include "longdouble.h"

63 static const long double small = 1.0e-20L, one = 1.0, two = 2.0,
64 #ifndef lint
65     big = 1.0e+20L,
66 #endif
67     threshold = 45.0L;

69 long double
70 tanhl(long double x) {
71     long double t, y, z;
72     int signx;
73     volatile long double dummy;

75     if (isnanl(x))
76         return (x + x);          /* x is NaN */
77     signx = signbitl(x);
78     t = fabsl(x);
79     z = one;
80     if (t <= threshold) {
81         if (t > one)
82             z = one - two / (expm1(t + t) + two);
83         else if (t > small) {
84             y = expm1(-t - t);
85             z = -y / (y + two);
86         } else {
87 #ifndef lint
88             dummy = t + big;
89                                     /* inexact if t != 0 */
90 #endif
91             return (x);
92         }
93     } else if (!finitel(t))
94         return (copysignl(one, x));
95     else
96         return (signx ? -z + small * small : z - small * small);
97     return (signx ? -z : z);
98 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/R/copysignf.c

1

\*\*\*\*\*

1189 Mon Oct 27 12:36:56 2014

new/usr/src/lib/libm/common/R/copysignf.c

5262 libm needs to be carefully undef'd  
5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak copysignf = __copysignf
32 #endif
33
34 #include "libm.h"
35
36 float
37 copysignf(float x, float y) {
38     float w;
39
40     *(int *) &w = (*(int *) &x & ~0x80000000) | (*(int *) &y & 0x80000000);
41     return (w);
42 }
43
44 unchanged_portion_omitted
```

new/usr/src/lib/libm/common/R/ilogbf.c

1

```
*****
2102 Mon Oct 27 12:36:56 2014
new/usr/src/lib/libm/common/R/ilogbf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak ilogbf = __ilogbf
32 #endif
```

```
32 #include "libm.h"
33 #include "xpg6.h" /* __xpg6 */
```

```
35 #if defined(__x86)
37 #if defined(USE_FPSCALE) || defined(__x86)
36 static const float two25 = 33554432.0F;
37 #else
```

```
38 /*
39  * v: a non-zero subnormal |x|
40  */
41 static int
42 ilogbf_subnormal(unsigned v) {
43     int r = -126 - 23;

44     if (v & 0xffff0000)
45         r += 16, v >>= 16;
46     if (v & 0xff00)
47         r += 8, v >>= 8;
48     if (v & 0xf0)
49         r += 4, v >>= 4;
50     v <<= 1;
51     return (r + ((0xffffaa50 >> v) & 0x3));
52 }
53 }
```

```
54 #endif /* defined(__x86) */
56 #endif /* defined(USE_FPSCALE) */
```

```
56 static int
```

new/usr/src/lib/libm/common/R/ilogbf.c

2

```
57 raise_invalid(int v) { /* SUSv3 requires ilogbf(0,+/-Inf,NaN) raise invalid */
58 #ifndef lint
59     if ((__xpg6 & _C99SUSv3_ilogb_0InfNaN_raises_invalid) != 0) {
60         static const double zero = 0.0;
61         volatile double dummy;

62         dummy = zero / zero;
63     }
64 #endif
65 return (v);
66 }

69 int
70 ilogbf(float x) {
71     int k = *((int *) &x) & ~0x80000000;

72     if (k < 0x00800000) {
73         if (k == 0)
74             return (raise_invalid(0x80000001));
75         else {
76             #if defined(__x86)
77             #if defined(USE_FPSCALE) || defined(__x86)
78                 x *= two25;
79                 return (((*(int *) &x) & 0x7f800000) >> 23) - 152;
80             #else
81                 return (ilogbf_subnormal(k));
82             #endif
83         }
84     } else if (k < 0x7f800000)
85         return ((k >> 23) - 127);
86     else
87         return (raise_invalid(0x7fffffff));
88 }
```

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_



new/usr/src/lib/libm/common/R/isnanf.c

1

```
*****
1156 Mon Oct 27 12:36:57 2014
new/usr/src/lib/libm/common/R/isnanf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak isnanf = __isnanf
31 #pragma weak _isnanf = __isnanf
33 #endif

33 #include "libm.h"

35 int
36 isnanf(float x) {
37     return ((* (int *) &x & ~0x80000000) > 0x7f800000);
38 }
_____unchanged_portion_omitted_____
```

```

*****
2080 Mon Oct 27 12:36:57 2014
new/usr/src/lib/libm/common/R/logbf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak logbf = __logbf
32 #endif

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

36 #if defined(__x86)
38 #if defined(USE_FPSCALE) || 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 }
55 #endif /* defined(__x86) */
57 #endif /* defined(USE_FPSCALE) */

```

```

57 static float
58 raise_division(float t) {
59 #pragma STDC FENV_ACCESS ON
60     static const float zero = 0.0F;
61     return (t / zero);
62 }

64 float
65 logbf(float x) {
66     int k = *((int *) &x) & ~0x80000000;

68     if (k < 0x00800000) {
69         if (k == 0)
70             return (raise_division(-1.0F));
71         else if ((__xpg6 & _C99SUSv3_logb) != 0) {
72 #if defined(__x86)
74 #if defined(USE_FPSCALE) || defined(__x86)
73             x *= two25;
74             return ((float) (((*(int *) &x) & 0x7f800000) >> 23) -
75                     152));
76 #else
77             return ((float) ilogbf_subnormal(k));
78 #endif
79         } else
80             return (-126.F);
81     } else if (k < 0x7f800000)
82         return ((float) ((k >> 23) - 127));
83     else
84         return (x * x);
85 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/R/nextafterf.c

1

```
*****
1937 Mon Oct 27 12:36:58 2014
new/usr/src/lib/libm/common/R/nextafterf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nextafterf = __nextafterf
32 #endif

32 #include "libm.h"

34 float
35 nextafterf(float x, float y) {
36     float w;
37     int *pw = (int *) &w;
38     int *px = (int *) &x;
39     int *py = (int *) &y;
40     int ix, iy, iz;

42     ix = px[0];
43     iy = py[0];
44     if ((ix & ~0x80000000) > 0x7f800000)
45         return (x * y); /* + -> * for Cheetah */
46     if ((iy & ~0x80000000) > 0x7f800000)
47         return (y * x); /* + -> * for Cheetah */
48     if (ix == iy || (ix | iy) == 0x80000000)
49         return (y); /* C99 requirement */
50     if ((ix & ~0x80000000) == 0)
51         iz = 1 | (iy & 0x80000000);
52     else if (ix > 0) {
53         if (ix > iy)
54             iz = ix - 1;
55         else
56             iz = ix + 1;
57     } else {
58         if (iy < 0 && ix < iy)
```

new/usr/src/lib/libm/common/R/nextafterf.c

2

```
59             iz = ix + 1;
60         else
61             iz = ix - 1;
62     }
63     pw[0] = iz;
64     ix = iz & 0x7f800000;
65     if (ix == 0x7f800000) {
66         /* raise overflow */
67         volatile float t;

69         *(int *) &t = 0x7f7fffff;
70         t *= t;
71     } else if (ix == 0) {
72         /* raise underflow */
73         volatile float t;

75         *(int *) &t = 0x00800000;
76         t *= t;
77     }
78     return (w);
79 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/R/remainderf.c

1

\*\*\*\*\*

1381 Mon Oct 27 12:36:58 2014

new/usr/src/lib/libm/common/R/remainderf.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak remainderf = __remainderf
32 #endif
33
32 #include "libm.h"
33
34 float
35 remainderf(float x, float y) {
36     if (isnanf(x) || isnanf(y))
37         return (x * y);
38     if (y == 0.0f || (*(int *) &x & ~0x80000000) == 0x7f800000) {
39         /* y is 0 or x is infinite; raise invalid and return NaN */
40         y = 0.0f;
41         *(int *) &x = 0x7f800000;
42         return (x * y);
43     }
44     return ((float) remainder((double) x, (double) y));
45 }
46
unchanged_portion_omitted
```

```

*****
3780 Mon Oct 27 12:36:58 2014
new/usr/src/lib/libm/common/R/rintf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak aintf = __aintf
31 #pragma weak anintf = __anintf
32 #pragma weak irintf = __irintf
33 #pragma weak nintf = __nintf
34 #pragma weak rintf = __rintf
36 #endif

36 /* INDENT OFF */
37 /*
38 * aintf(x)    return x chopped to integral value
39 * anintf(x)   return sign(x)*(|x|+0.5) chopped to integral value
40 * irintf(x)   return rint(x) in integer format
41 * nintf(x)    return aint(x) in integer format
42 * rintf(x)    return x rounded to integral according to the rounding direction
43 *
44 * NOTE: rintf(x), aintf(x) and anintf(x) return results with the same sign as
45 * x's, including 0.0.
46 */

48 #include "libm.h"

50 static const float xf[] = {
51 /* ZEROF */      0.0f,
52 /* TWO_23F */   8.3886080000e6f,
53 /* MTWO_23F */ -8.3886080000e6f,
54 /* ONEF */      1.0f,
55 /* MONEF */     -1.0f,
56 /* HALFF */     0.5f,
57 /* MHALFF */   -0.5f,
58 /* HUGE */     1.0e30f,

```

```

59 };
_____unchanged_portion_omitted_____

```

```

*****
2388 Mon Oct 27 12:36:59 2014
new/usr/src/lib/libm/common/R/scalbnf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak scalbnf = __scalbnf
32 #endif

32 #include "libm.h"
33 #include <float.h>          /* FLT_MAX, FLT_MIN */
34 #include <stdlib.h>         /* abs */

36 static const float twom25f = 2.98023223876953125e-8F;
37 #if defined(__x86)
39 #if defined(USE_FPSCALE) || defined(__x86)
38 static const float two23f = 8388608.0F;
39 #else
40 /*
41  * v: a non-zero subnormal |x|; returns [-22, 0]
42  */
43 static int
44 ilogbf_biased(unsigned v) {
45     int r = -22;

47     if (v & 0xffff0000)
48         r += 16, v >>= 16;
49     if (v & 0xff00)
50         r += 8, v >>= 8;
51     if (v & 0xf0)
52         r += 4, v >>= 4;
53     v <<= 1;
54     return (r + ((0xffffaa50 >> v) & 0x3));
55 }
56 #endif /* defined(__x86) */
58 #endif /* defined(USE_FPSCALE) */

```

```

58 float
59 scalbnf(float x, int n) {
60     int *px = (int *) &x, ix, k;

62     ix = *px & ~0x80000000;
63     k = ix >> 23;
64     if (k == 0xff)
65 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
66         return (ix > 0x7f800000 ? x * x : x);
67 #else
68         return (x + x);
69 #endif
70     if (ix == 0 || n == 0)
71         return (x);
72     if (k == 0) {
73 #if defined(__x86)
74 #if defined(USE_FPSCALE) || defined(__x86)
74         x *= two23f;
75         k = ((*px & ~0x80000000) >> 23) - 23;
76 #else
77         k = ilogbf_biased(ix);
78         *px = (*px & 0x80000000) | (ix << (-k + 1));
79 #endif
80     }
81     if ((unsigned) abs(n) >= 131072) /* cast to unsigned for -2^31 */
82         n >>= 1; /* avoid subsequent integer overflow */
83     k += n;
84     if (k > 0xfe)
85         return (FLT_MAX * copysignf(FLT_MAX, x));
86     if (k <= -25)
87         return (FLT_MIN * copysignf(FLT_MIN, x));
88     if (k > 0) {
89         *px = (*px & ~0x7f800000) | (k << 23);
90         return (x);
91     }
92     k += 25;
93     *px = (*px & ~0x7f800000) | (k << 23);
94     return (x * twom25f);
95 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/R/significandf.c

1

\*\*\*\*\*

1337 Mon Oct 27 12:36:59 2014

new/usr/src/lib/libm/common/R/significandf.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 #if defined(ELFOBJ)
30 #pragma weak significandf = __significandf
32 #endif
```

```
32 #include "libm.h"
```

```
34 float
35 significandf(float x) {
36     int ix = *(int *) &x & ~0x80000000;
37
38     if (ix == 0 || ix >= 0x7f800000) /* 0/+--Inf/NaN */
39 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
40         return (ix > 0x7f800000 ? x * x : x);
41 #else
42         return (x + x);
43 #endif
44     else
45         return (scalbnf(x, -ilogbf(x)));
46 }
```

unchanged\_portion\_omitted

```

*****
1138 Mon Oct 27 12:37:00 2014
new/usr/src/lib/libm/common/R/sqrtf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 sqrtf = __sqrtf

31 #include "libm.h"

33 #ifdef __INLINE

34 extern float __inline_sqrtf(float);

36 float
37 sqrtf(float x) {
38     return (__inline_sqrtf(x));
39 }

42 #else /* defined(__INLINE) */

44 static const float huge = 1.0e35F, tiny = 1.0e-35F, zero = 0.0f;

46 float
47 sqrtf(float x) {
48     float    dz, w;
49     int      *pw = (int *)&w;
50     int      ix, j, r, q, m, n, s, t;

52     w = x;
53     ix = *pw[0];
54     if (ix <= 0) {
55         /* x is <= 0 or nan */
56         j = ix & 0x7fffffff;
57         if (j == 0)
58             return (w);
59         return ((w * zero) / zero);
60     }

```

```

62     if ((ix & 0x7f800000) == 0x7f800000) {
63         /* x is +inf or nan */
64         return (w * w);
65     }

67     m = ir_ilogb_(&w);
68     n = -m;
69     w = r_scalbn_(&w, (int *)&n);
70     ix = (pw[0] & 0x007fffff) | 0x00800000;
71     n = m / 2;
72     if ((n + n) != m) {
73         ix = ix + ix;
74         m -= 1;
75         n = m / 2;
76     }

78     /* generate sqrt(x) bit by bit */
79     ix <<= 1;
80     q = s = 0;
81     r = 0x01000000;
82     for (j = 1; j <= 25; j++) {
83         t = s + r;
84         if (t <= ix) {
85             s = t + r;
86             ix -= t;
87             q += r;
88         }
89         ix <<= 1;
90         r >>= 1;
91     }
92     if (ix == 0)
93         goto done;

95     /* raise inexact and determine the ambient rounding mode */
96     dz = huge - tiny;
97     if (dz < huge)
98         goto done;
99     dz = huge + tiny;
100    if (dz > huge)
101        q += 1;
102    q += (q & 1);

104 done:
105    pw[0] = (q >> 1) + 0x3f000000;
106    return (r_scalbn_(&w, (int *)&n));
107 }

109 #endif /* defined(__INLINE) */

```



new/usr/src/lib/libm/common/m9x/\_\_\_fex\_hdlr.c

1

```
*****
21422 Mon Oct 27 12:37:00 2014
new/usr/src/lib/libm/common/m9x/___fex_hdlr.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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"
31 #undef lint
32 #include <signal.h>
33 #include <siginfo.h>
34 #include <ucontext.h>
35 #include <stdio.h>
36 #include <stdlib.h>
37 #include <unistd.h>
38 #include <thread.h>
39 #include <math.h>
40 #if defined(__SUNPRO_C)
41 #include <sunmath.h>
42 #endif
43 #include <fenv.h>
44 #include "fex_handler.h"
45 #include "fenv_inlines.h"

47 #if defined(__sparc) && !defined(__sparcv9)
48 #include <sys/proefs.h>
49 #endif

51 /* 2.x signal.h doesn't declare sigemptyset or sigismember
52  * if they're #defined (see sys/signal.h) */
53 extern int sigemptyset(sigset_t *);
54 extern int sigismember(const sigset_t *, int);

56 /* external globals */
57 void (*__mt_fex_sync)() = NULL; /* for synchronization with libmtsk */
58 #pragma weak __mt_fex_sync

60 #ifndef LIBM_MT_FEX_SYNC
```

new/usr/src/lib/libm/common/m9x/\_\_\_fex\_hdlr.c

2

```
60 void (*__libm_mt_fex_sync)() = NULL; /* new, improved version of above */
61 #pragma weak __libm_mt_fex_sync
62 #endif

63 /* private variables */
64 static fex_handler_t main_handlers;
65 static int handlers_initialized = 0;
66 static thread_key_t handlers_key;
67 static mutex_t handlers_key_lock = DEFAULTMUTEX;

69 static struct sigaction oact = { 0, SIG_DFL };
70 static mutex_t hdlr_lock = DEFAULTMUTEX;
71 static int hdlr_installed = 0;

73 /* private const data */
74 static const int te_bit[FEX_NUM_EXC] = {
75     1 << fp_trap_inexact,
76     1 << fp_trap_division,
77     1 << fp_trap_underflow,
78     1 << fp_trap_overflow,
79     1 << fp_trap_invalid,
80     1 << fp_trap_invalid,
81     1 << fp_trap_invalid,
82     1 << fp_trap_invalid,
83     1 << fp_trap_invalid,
84     1 << fp_trap_invalid,
85     1 << fp_trap_invalid,
86     1 << fp_trap_invalid
87 };
    unchanged_portion_omitted

148 #ifdef LIBM_MT_FEX_SYNC
146 /*
147  * The following function may be used for synchronization with any
148  * internal project that manages multiple threads
149 */
150 enum __libm_mt_fex_sync_actions {
151     __libm_mt_fex_start_master = 0,
152     __libm_mt_fex_start_slave,
153     __libm_mt_fex_finish_master,
154     __libm_mt_fex_finish_slave
155 };
    unchanged_portion_omitted
207 #endif

205 #if defined(__sparc)

207 /*
208  * Code for setting or clearing interval mode on US-III and above.
209  * This is embedded as data so we don't have to mark the library
210  * as a v8plusb/v9b object. (I could have just used one entry and
211  * modified the second word to set the bits I want, but that would
212  * have required another mutex.)
213 */
214 static const unsigned int siam[][2] = {
215     { 0x81c3e008, 0x81b01020 }, /* retl, siam 0 */
216     { 0x81c3e008, 0x81b01024 }, /* retl, siam 4 */
217     { 0x81c3e008, 0x81b01025 }, /* retl, siam 5 */
218     { 0x81c3e008, 0x81b01026 }, /* retl, siam 6 */
219     { 0x81c3e008, 0x81b01027 }, /* retl, siam 7 */
220 };
    unchanged_portion_omitted

799 /*
800  * Update the trap enable bits according to the selected modes
801 */
```

```
802 void
803 ___fex_update_te()
804 {
805     struct fex_handler_data *thr_handlers;
806     struct sigaction        act, tmpact;
807     sigset_t                blocked;
808     unsigned long           fsr;
809     int                      te;

811     /* determine which traps are needed */
812     thr_handlers = ___fex_get_thr_handlers();
813     ___fenv_getfsr(&fsr);
814     te = ___fex_te_needed(thr_handlers, fsr);

816     /* install ___fex_hdlr as necessary */
817     if (!hdlr_installed && te) {
818         act.sa_handler = ___fex_hdlr;
819         sigemptyset(&act.sa_mask);
820         act.sa_flags = SA_SIGINFO;
821         sigaction(SIGFPE, &act, &tmpact);
822         if (tmpact.sa_handler != ___fex_hdlr)
823         {
824             mutex_lock(&hdlr_lock);
825             oact = tmpact;
826             mutex_unlock(&hdlr_lock);
827         }
828         hdlr_installed = 1;
829     }

831     /* set the new trap enable bits (only if SIGFPE is not blocked) */
832     if (sigprocmask(0, NULL, &blocked) == 0 &&
833         !sigismember(&blocked, SIGFPE)) {
834         ___fenv_set_te(fsr, te);
835         ___fenv_setfsr(&fsr);
836     }

838     /* synchronize with libm_tsk */
839     ___mt_fex_sync = ___fex_sync_with_libm_tsk;

845 #ifdef LIBM_MT_FEX_SYNC
841     /* synchronize with other projects */
842     ___libm_mt_fex_sync = ___fex_sync_with_threads;
848 #endif
843 }
_____unchanged_portion_omitted_____
```

```
*****
1460 Mon Oct 27 12:37:01 2014
new/usr/src/lib/libm/common/m9x/fdim.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak fdim = __fdim
32 #endif
33
34 /*
35 * fdim(x,y) returns x - y if x > y, +0 if x <= y, and NaN if x and
36 * y are unordered.
37 *
38 * fdim(x,y) raises overflow or inexact if x > y and x - y overflows
39 * or is inexact. It raises invalid if either operand is a signaling
40 * NaN. Otherwise, it raises no exceptions.
41 */
42
43 #include "libm.h" /* for islessequal macro */
44
45 double
46 __fdim(double x, double y) {
47     if (islessequal(x, y)) {
48         x = 0.0;
49         y = -x;
50     }
51     return (x - y);
52 }
53
54 unchanged_portion_omitted
```

new/usr/src/lib/libm/common/m9x/fdimf.c

1

\*\*\*\*\*

1522 Mon Oct 27 12:37:01 2014

new/usr/src/lib/libm/common/m9x/fdimf.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 #if defined(ELFOBJ)
```

```
30 #pragma weak fdimf = __fdimf
```

```
32 #endif
```

```
32 #include "libm.h" /* for islessequal macro */
```

```
34 float
35 __fdimf(float x, float y) {
36     /*
37      * On SPARC v8plus/v9, this could be implemented as follows
38      * (assuming %f0 = x, %f1 = y, return value left in %f0):
39      *
40      * fcmps    %fcc0,%f0,%f1
41      * st       %g0,[scratch] ! use fzero instead of st/ld
42      * ld       [scratch],%f2 ! if VIS is available
43      * fnegs   %f2,%f3
44      * fmovsle %fcc0,%f2,%f0
45      * fmovsle %fcc0,%f3,%f1
46      * fsubs  %f0,%f1,%f0
47      */
48     if (islessequal(x, y)) {
49         x = 0.0f;
50         y = -x;
51     }
52     return (x - y);
53 }
```

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

new/usr/src/lib/libm/common/m9x/fdiml.c

1

```
*****
1196 Mon Oct 27 12:37:02 2014
new/usr/src/lib/libm/common/m9x/fdiml.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fdiml = __fdiml
32 #endif

32 #include "libm.h"      /* for islessequal macro */

34 long double
35 __fdiml(long double x, long double y) {
36     if (islessequal(x, y)) {
37         x = 0.01;
38         y = -x;
39     }
40     return (x - y);
41 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/fma.c

1

```
*****
11289 Mon Oct 27 12:37:02 2014
new/usr/src/lib/libm/common/m9x/fma.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fma = __fma
32 #endif

32 #include "libm.h"
33 #include "fma.h"
34 #include "fenv_inlines.h"

36 #if defined(__sparc)

38 static const union {
39     unsigned i[2];
40     double d;
41 } C[] = {
    _____unchanged_portion_omitted_____

```

```

*****
5636 Mon Oct 27 12:37:02 2014
new/usr/src/lib/libm/common/m9x/fmaf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmaf = __fmaf
32 #endif

32 #include "libm.h"
33 #include "fma.h"
34 #include "fenv_inlines.h"

36 #if defined(__sparc)

38 /*
39  * fmaf for SPARC: 32-bit single precision, big-endian
40 */
41 float
42 __fmaf(float x, float y, float z) {
43     union {
44         unsigned i[2];
45         double d;
46     } xy, zz;
47     unsigned u, s;
48     int exy, ez;

50     /*
51      * the following operations can only raise the invalid exception,
52      * and then only if either x*y is of the form Inf*0 or one of x,
53      * y, or z is a signaling NaN
54      */
55     xy.d = (double) x * y;
56     zz.d = (double) z;

58     /*

```

```

59     * if the sum xy + z will be exact, just compute it and cast the
60     * result to float
61     */
62     exy = (xy.i[0] >> 20) & 0x7fff;
63     ez = (zz.i[0] >> 20) & 0x7fff;
64     if ((ez - exy <= 4 && exy - ez <= 28) || exy == 0x7fff || exy == 0 ||
65         ez == 0x7fff || ez == 0) {
66         return ((float) (xy.d + zz.d));
67     }

69     /*
70     * collapse the tail of the smaller summand into a "sticky bit"
71     * so that the sum can be computed without error
72     */
73     if (ez > exy) {
74         if (ez - exy < 31) {
75             u = xy.i[1];
76             s = 2 << (ez - exy);
77             if (u & (s - 1))
78                 u |= s;
79             xy.i[1] = u & ~(s - 1);
80         } else if (ez - exy < 51) {
81             u = xy.i[0];
82             s = 1 << (ez - exy - 31);
83             if ((u & (s - 1)) | xy.i[1])
84                 u |= s;
85             xy.i[0] = u & ~(s - 1);
86             xy.i[1] = 0;
87         } else {
88             /* collapse all of xy into a single bit */
89             xy.i[0] = (xy.i[0] & 0x80000000) | ((ez - 51) << 20);
90             xy.i[1] = 0;
91         }
92     } else {
93         if (exy - ez < 31) {
94             u = zz.i[1];
95             s = 2 << (exy - ez);
96             if (u & (s - 1))
97                 u |= s;
98             zz.i[1] = u & ~(s - 1);
99         } else if (exy - ez < 51) {
100            u = zz.i[0];
101            s = 1 << (exy - ez - 31);
102            if ((u & (s - 1)) | zz.i[1])
103                u |= s;
104            zz.i[0] = u & ~(s - 1);
105            zz.i[1] = 0;
106        } else {
107            /* collapse all of zz into a single bit */
108            zz.i[0] = (zz.i[0] & 0x80000000) | ((exy - 51) << 20);
109            zz.i[1] = 0;
110        }
111    }

113     return ((float) (xy.d + zz.d));
114 }
_____unchanged_portion_omitted_____

```

new/usr/src/lib/libm/common/m9x/fmal.c

1

```
*****
28108 Mon Oct 27 12:37:03 2014
new/usr/src/lib/libm/common/m9x/fmal.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmal = __fmal
32 #endif

32 #include "libm.h"
33 #include "fma.h"
34 #include "fenv_inlines.h"

36 #if defined(__sparc)

38 static const union {
39     unsigned i[2];
40     double d;
41 } C[] = {
    _____unchanged_portion_omitted_____

```



new/usr/src/lib/libm/common/m9x/fmax.c

1

```
*****
1982 Mon Oct 27 12:37:03 2014
new/usr/src/lib/libm/common/m9x/fmax.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmax = __fmax
32 #endif

32 /*
33 * fmax(x,y) returns the larger of x and y. If just one of the
34 * arguments is NaN, fmax returns the other argument. If both
35 * arguments are NaN, fmax returns NaN.
36 *
37 * See fmaxf.c for a discussion of implementation trade-offs.
38 */

40 #include "libm.h" /* for isgreaterequal macro */
41 #include <fenv.h>

43 double
44 __fmax(double x, double y) {
45     union {
46         unsigned i[2];
47         double d;
48     } xx, yy;
49     unsigned s;

51     /* if y is nan, replace it by x */
52     if (y != y)
53         y = x;

55     /* if x is nan, replace it by y */
56     if (x != x)
57         x = y;
```

new/usr/src/lib/libm/common/m9x/fmax.c

2

```
59     /* At this point, x and y are either both numeric, or both NaN */
60     if (!isnan(x) && !isgreaterequal(x, y))
61         x = y;

63     /*
64      * clear the sign of the result if either x or y has its sign clear
65      */
66     xx.d = x;
67     yy.d = y;
68 #if defined(__sparc)
69     s = ~(xx.i[0] & yy.i[0]) & 0x80000000;
70     xx.i[0] &= ~s;
71 #elif defined(__x86)
72     s = ~(xx.i[1] & yy.i[1]) & 0x80000000;
73     xx.i[1] &= ~s;
74 #else
75 #error Unknown architecture
76 #endif

78     return (xx.d);
79 }
_____unchanged_portion_omitted_____
```

```

*****
4298 Mon Oct 27 12:37:04 2014
new/usr/src/lib/libm/common/m9x/fmaxf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmaxf = __fmaxf
32 #endif

32 /*
33 * fmax(x,y) returns the larger of x and y. If just one of the
34 * arguments is NaN, fmax returns the other argument. If both
35 * arguments are NaN, fmax returns NaN (ideally, one of the
36 * argument NaNs).
37 *
38 * C99 does not require that fmax(-0,+0) = fmax(+0,-0) = +0, but
39 * ideally fmax should satisfy this.
40 *
41 * C99 makes no mention of exceptions for fmax. I suppose ideally
42 * either fmax never raises any exceptions or else it raises the
43 * invalid operation exception if and only if some argument is a
44 * signaling NaN. In the former case, fmax should always return
45 * one of its arguments. In the latter, fmax shouldn't return a
46 * signaling NaN, although when both arguments are signaling NaNs,
47 * this ideal is at odds with the stipulation that fmax should
48 * always return one of its arguments.
49 *
50 * Commutativity of fmax follows from the properties listed above
51 * except when both arguments are NaN. In that case, fmax may be
52 * declared commutative by fiat because there is no portable way
53 * to tell different NaNs apart. Ideally fmax would be truly com-
54 * mutative for all arguments.
55 *
56 * On SPARC V8, fmax must involve tests and branches. Ideally,
57 * an implementation on SPARC V9 should avoid branching, using
58 * conditional moves instead where necessary, and be as efficient

```

```

59 * as possible in its use of other resources.
60 *
61 * It appears to be impossible to attain all of the aforementioned
62 * ideals simultaneously. The implementation below satisfies the
63 * following (on SPARC):
64 *
65 * 1. fmax(x,y) returns the larger of x and y if neither x nor y
66 * is NaN and the non-NaN argument if just one of x or y is NaN.
67 * If both x and y are NaN, fmax(x,y) returns x unchanged.
68 * 2. fmax(-0,+0) = fmax(+0,-0) = +0.
69 * 3. If either argument is a signaling NaN, fmax raises the invalid
70 * operation exception. Otherwise, it raises no exceptions.
71 */

73 #include "libm.h" /* for isgreaterequal macro */

75 float
76 __fmaxf(float x, float y) {
77     /*
78      * On SPARC v8plus/v9, this could be implemented as follows
79      * (assuming %f0 = x, %f1 = y, return value left in %f0):
80      *
81      * fcmps      %fcc0,%f1,%f1
82      * fmovsu     %fcc0,%f0,%f1
83      * fcmps      %fcc0,%f0,%f1
84      * fmovsul    %fcc0,%f1,%f0
85      * st         %f0,[x]
86      * st         %f1,[y]
87      * ld         [x],%l0
88      * ld         [y],%l1
89      * and        %l0,%l1,%l2
90      * sethi     %hi(0x80000000),%l3
91      * andn      %l3,%l2,%l2
92      * andn      %l0,%l2,%l0
93      * st         %l0,[x]
94      * ld         [x],%f0
95      *
96      * If VIS instructions are available, use this code instead:
97      *
98      * fcmps      %fcc0,%f1,%f1
99      * fmovsu     %fcc0,%f0,%f1
100     * fcmps      %fcc0,%f0,%f1
101     * fmovsul    %fcc0,%f1,%f0
102     * fands      %f0,%f1,%f2
103     * fzeros     %f3
104     * fnegs      %f3,%f3
105     * fandnot2s %f3,%f2,%f2
106     * fandnot2s %f0,%f2,%f0
107     *
108     * If VIS 3.0 instructions are available, use this:
109     *
110     * flcmps     %fcc0,%f0,%f1
111     * fmovslg    %fcc0,%f1,%f0 ! move if %fcc0 is 1 or 2
112     */

114     union {
115         unsigned i;
116         float f;
117     } xx, yy;
118     unsigned s;

120     /* if y is nan, replace it by x */
121     if (y != y)
122         y = x;

124     /* if x is nan, replace it by y */

```

```
125     if (x != x)
126         x = y;

128     /* At this point, x and y are either both numeric, or both NaN */
129     if (!isnan(x) && !isgreater(x, y))
130         x = y;

132     /*
133     * clear the sign of the result if either x or y has its sign clear
134     */
135     xx.f = x;
136     yy.f = y;
137     s = ~(xx.i & yy.i) & 0x80000000;
138     xx.i &= ~s;

140     return (xx.f);
141 }
unchanged_portion_omitted
```

```

*****
1847 Mon Oct 27 12:37:04 2014
new/usr/src/lib/libm/common/m9x/fmaxl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmaxl = __fmaxl
32 #endif

32 #include "libm.h"      /* for isgreaterequal macro */

34 long double
35 __fmaxl(long double x, long double y) {
36     union {
37 #if defined(__sparc)
38         unsigned i[4];
39 #elif defined(__x86)
40         unsigned i[3];
41 #else
42 #error Unknown architecture
43 #endif
44         long double ld;
45     } xx, yy;
46     unsigned s;

48     /* if y is nan, replace it by x */
49     if (y != y)
50         y = x;

52     /* if x is nan, replace it by y */
53     if (x != x)
54         x = y;

56     /* At this point, x and y are either both numeric, or both NaN */
57     if (!isnan(x) && !isgreaterequal(x, y))
58         x = y;

```

```

60     /*
61      * clear the sign of the result if either x or y has its sign clear
62      */
63     xx.ld = x;
64     yy.ld = y;
65 #if defined(__sparc)
66     s = ~(xx.i[0] & yy.i[0]) & 0x80000000;
67     xx.i[0] &= ~s;
68 #elif defined(__x86)
69     s = ~(xx.i[2] & yy.i[2]) & 0x8000;
70     xx.i[2] &= ~s;
71 #else
72 #error Unknown architecture
73 #endif

75     return (xx.ld);
76 }

```

unchanged portion omitted

new/usr/src/lib/libm/common/m9x/fmin.c

1

```
*****
1979 Mon Oct 27 12:37:05 2014
new/usr/src/lib/libm/common/m9x/fmin.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fmin = __fmin
32 #endif

32 /*
33 * fmin(x,y) returns the smaller of x and y. If just one of the
34 * arguments is NaN, fmin returns the other argument. If both
35 * arguments are NaN, fmin returns NaN.
36 *
37 * See fmaxf.c for a discussion of implementation trade-offs.
38 */

40 #include "libm.h" /* for islessequal macro */

42 #include "fenv_inlines.h"
43 #include <stdio.h>
44 #include <sys/isa_defs.h>

46 double
47 __fmin(double x, double y) {
48     union {
49         unsigned i[2];
50         double d;
51     } xx, yy;
52     unsigned s;
53
54     /* if y is nan, replace it by x */
55     if (y != y)
56         y = x;

58     /* if x is nan, replace it by y */
```

new/usr/src/lib/libm/common/m9x/fmin.c

2

```
59     if (x != x)
60         x = y;

62     /* At this point, x and y are either both numeric, or both NaN */
63     if (!isnan(x) && !islessequal(x, y))
64         x = y;

66     /*
67      * set the sign of the result if either x or y has its sign set
68      */
69     xx.d = x;
70     yy.d = y;
71 #if defined(_BIG_ENDIAN)
72     s = (xx.i[0] | yy.i[0]) & 0x80000000;
73     xx.i[0] |= s;
74 #else
75     s = (xx.i[1] | yy.i[1]) & 0x80000000;
76     xx.i[1] |= s;
77 #endif

79     return (xx.d);
80 }
_____unchanged_portion_omitted_____
```

```

*****
2386 Mon Oct 27 12:37:05 2014
new/usr/src/lib/libm/common/m9x/fminf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fminf = __fminf
32 #endif

32 #include "libm.h" /* for islessequal macro */

34 float
35 __fminf(float x, float y) {
36     /*
37      * On SPARC v8plus/v9, this could be implemented as follows
38      * (assuming %f0 = x, %f1 = y, return value left in %f0):
39      *
40      * fcmps    %fcc0,%f1,%f1
41      * fmovsu   %fcc0,%f0,%f1
42      * fcmps    %fcc0,%f0,%f1
43      * fmovsug  %fcc0,%f1,%f0
44      * st       %f0,[x]
45      * st       %f1,[y]
46      * ld       [x],%l0
47      * ld       [y],%l1
48      * or       %l0,%l1,%l2
49      * sethi    %hi(0x80000000),%l3
50      * and      %l3,%l2,%l2
51      * or       %l0,%l2,%l0
52      * st       %l0,[x]
53      * ld       [x],%f0
54      *
55      * If VIS instructions are available, use this code instead:
56      *
57      * fcmps    %fcc0,%f1,%f1
58      * fmovsu   %fcc0,%f0,%f1

```

```

59      * fcmps    %fcc0,%f0,%f1
60      * fmovsug  %fcc0,%f1,%f0
61      * fors     %f0,%f1,%f2
62      * fzeros   %f3
63      * fnegs    %f3,%f3
64      * fands    %f3,%f2,%f2
65      * fors     %f0,%f2,%f0
66      *
67      * If VIS 3.0 instructions are available, use this:
68      *
69      * flcmps    %fcc0,%f0,%f1
70      * fmovsge   %fcc0,%f1,%f0 ! move if %fcc0 is 0 or 2
71      */

73     union {
74         unsigned i;
75         float f;
76     } xx, yy;
77     unsigned s;

79     /* if y is nan, replace it by x */
80     if (y != y)
81         y = x;

83     /* if x is nan, replace it by y */
84     if (x != x)
85         x = y;

87     /* At this point, x and y are either both numeric, or both NaN */
88     if (!isnan(x) && !islessequal(x, y))
89         x = y;

91     /*
92      * set the sign of the result if either x or y has its sign set
93      */
94     xx.f = x;
95     yy.f = y;
96     s = (xx.i | yy.i) & 0x80000000;
97     xx.i |= s;

99     return (xx.f);
100 }
    unchanged_portion_omitted

```

new/usr/src/lib/libm/common/m9x/fminl.c

1

```
*****
1833 Mon Oct 27 12:37:06 2014
new/usr/src/lib/libm/common/m9x/fminl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak fminl = __fminl
32 #endif

32 #include "libm.h"      /* for islessequal macro */

34 long double
35 __fminl(long double x, long double y) {
36     union {
37 #if defined(__sparc)
38         unsigned i[4];
39 #elif defined(__x86)
40         unsigned i[3];
41 #else
42 #error Unknown architecture
43 #endif
44         long double ld;
45     } xx, yy;
46     unsigned s;

48     /* if y is nan, replace it by x */
49     if (y != y)
50         y = x;

52     /* if x is nan, replace it by y */
53     if (x != x)
54         x = y;

56     /* At this point, x and y are either both numeric, or both NaN */
57     if (!isnan(x) && !islessequal(x, y))
58         x = y;
```

new/usr/src/lib/libm/common/m9x/fminl.c

2

```
60     /*
61      * set the sign of the result if either x or y has its sign set
62      */
63     xx.ld = x;
64     yy.ld = y;
65 #if defined(__sparc)
66     s = (xx.i[0] | yy.i[0]) & 0x80000000;
67     xx.i[0] |= s;
68 #elif defined(__x86)
69     s = (xx.i[2] | yy.i[2]) & 0x8000;
70     xx.i[2] |= s;
71 #else
72 #error Unknown architecture
73 #endif

75     return (xx.ld);
76 }
unchanged_portion_omitted
```

```

*****
2799 Mon Oct 27 12:37:06 2014
new/usr/src/lib/libm/common/m9x/frexp.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak frexp = __frexp
32 #endif

32 /*
33 * frexp(x, exp) returns the normalized significand of x and sets
34 * *exp so that x = r*2^(*exp) where r is the return value. If x
35 * is finite and nonzero, 1/2 <= |r| < 1.
36 *
37 * If x is zero, infinite or NaN, frexp returns x and sets *exp = 0.
38 * (The relevant standards do not specify *exp when x is infinite or
39 * NaN, but this code sets it anyway.)
40 *
41 * If x is a signaling NaN, this code returns x without attempting
42 * to raise the invalid operation exception. If x is subnormal,
43 * this code treats it as nonzero regardless of nonstandard mode.
44 */

46 #include "libm.h"

48 double
49 __frexp(double x, int *exp) {
50     union {
51         unsigned i[2];
52         double d;
53     } xx, yy;
54     double t;
55     unsigned hx;
56     int e;

58     xx.d = x;

```

```

59     hx = xx.i[HIWORD] & ~0x80000000;

61     if (hx >= 0x7ff00000) { /* x is infinite or NaN */
62         *exp = 0;
63         return (x);
64     }

66     e = 0;
67     if (hx < 0x00100000) { /* x is subnormal or zero */
68         if ((hx | xx.i[LOWORD]) == 0) {
69             *exp = 0;
70             return (x);
71         }

73         /*
74          * normalize x by regarding it as an integer
75          *
76          * Here we use 32-bit integer arithmetic to avoid trapping
77          * or emulating 64-bit arithmetic. If 64-bit arithmetic is
78          * available (e.g., in SPARC V9), do this instead:
79          *
80          * long lx = ((long) hx << 32) | xx.i[LOWORD];
81          * xx.d = (xx.i[HIWORD] < 0)? -lx : lx;
82          *
83          * If subnormal arithmetic doesn't trap, just multiply x by
84          * a power of two.
85          */
86         yy.i[HIWORD] = 0x43300000 | hx;
87         yy.i[LOWORD] = xx.i[LOWORD];
88         t = yy.d;
89         yy.i[HIWORD] = 0x43300000;
90         yy.i[LOWORD] = 0;
91         t -= yy.d; /* t = |x| scaled */
92         xx.d = ((int)xx.i[HIWORD] < 0)? -t : t;
93         hx = xx.i[HIWORD] & ~0x80000000;
94         e = -1074;
95     }

97     /* now xx.d is normal */
98     xx.i[HIWORD] = (xx.i[HIWORD] & ~0x7ff00000) | 0x3fe00000;
99     *exp = e + (hx >> 20) - 0x3fe;
100     return (xx.d);
101 }

_____unchanged_portion_omitted_____

```



new/usr/src/lib/libm/common/m9x/frexp.c

1

\*\*\*\*\*

1638 Mon Oct 27 12:37:07 2014

new/usr/src/lib/libm/common/m9x/frexp.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 #if defined(ELFOBJ)
30 #pragma weak frexp = __frexp
32 #endif
```

```
32 #include "libm.h"
```

```
34 float
35 __frexp(float x, int *exp) {
36     union {
37         unsigned i;
38         float f;
39     } xx;
40     unsigned hx;
41     int e;
42
43     xx.f = x;
44     hx = xx.i & ~0x80000000;
45
46     if (hx >= 0x7f800000) { /* x is infinite or NaN */
47         *exp = 0;
48         return (x);
49     }
50
51     e = 0;
52     if (hx < 0x00800000) { /* x is subnormal or zero */
53         if (hx == 0) {
54             *exp = 0;
55             return (x);
56         }
57     }
58     /* normalize x by regarding it as an integer */
```

new/usr/src/lib/libm/common/m9x/frexp.c

2

```
59         xx.f = (int) xx.i < 0 ? -(int) hx : (int) hx;
60         hx = xx.i & ~0x80000000;
61         e = -149;
62     }
63
64     /* now xx.f is normal */
65     xx.i = (xx.i & ~0x7f800000) | 0x3f000000;
66     *exp = e + (hx >> 23) - 0x7e;
67     return (xx.f);
68 }
```

unchanged\_portion\_omitted

```

*****
2623 Mon Oct 27 12:37:07 2014
new/usr/src/lib/libm/common/m9x/frexpl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak frexpl = __frexpl
32 #endif

32 #include "libm.h"

34 #if defined(__sparc)

36 long double
37 __frexpl(long double x, int *exp) {
38     union {
39         unsigned i[4];
40         long double q;
41     } xx;
42     unsigned hx;
43     int e, s;

45     xx.q = x;
46     hx = xx.i[0] & ~0x80000000;

48     if (hx >= 0x7fff0000) { /* x is infinite or NaN */
49         *exp = 0;
50         return (x);
51     }

53     e = 0;
54     if (hx < 0x00010000) { /* x is subnormal or zero */
55         if ((hx | xx.i[1] | xx.i[2] | xx.i[3]) == 0) {
56             *exp = 0;
57             return (x);
58         }

```

```

60         /* normalize x */
61         s = xx.i[0] & 0x80000000;
62         while ((hx | (xx.i[1] & 0xffff0000)) == 0) {
63             hx = xx.i[1];
64             xx.i[1] = xx.i[2];
65             xx.i[2] = xx.i[3];
66             xx.i[3] = 0;
67             e -= 32;
68         }
69         while (hx < 0x10000) {
70             hx = (hx << 1) | (xx.i[1] >> 31);
71             xx.i[1] = (xx.i[1] << 1) | (xx.i[2] >> 31);
72             xx.i[2] = (xx.i[2] << 1) | (xx.i[3] >> 31);
73             xx.i[3] <<= 1;
74             e--;
75         }
76         xx.i[0] = s | hx;
77     }

79     /* now xx.q is normal */
80     xx.i[0] = (xx.i[0] & ~0x7fff0000) | 0x3ffe0000;
81     *exp = e + (hx >> 16) - 0x3ffe;
82     return (xx.q);
83 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/m9x/ldexp.c

1

```
*****
1614 Mon Oct 27 12:37:08 2014
new/usr/src/lib/libm/common/m9x/ldexp.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak ldexp = __ldexp
32 #endif

32 #include "libm.h"
33 #include <errno.h>

35 double
36 ldexp(double x, int n) {
37     int *px = (int *) &x, ix = px[HIWORD] & ~0x80000000;

39     if (ix >= 0x7ff00000 || (px[LOWORD] | ix) == 0)
40 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
41         return (ix >= 0x7ff80000 ? x : x + x);
42         /* assumes sparc-like QNaN */
43 #else
44         return (x + x);
45 #endif
46     x = scalbn(x, n);
47     ix = px[HIWORD] & ~0x80000000;
48     /*
49      * SVID3 requires both overflow and underflow cases to set errno
50      * XPG3/XPG4/XPG4.2/SUSv2 requires overflow to set errno
51      */
52     if (ix >= 0x7ff00000 || (px[LOWORD] | ix) == 0)
53         errno = ERANGE;
54     return (x);
55 }
    unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/m9x/ldexpf.c

1

\*\*\*\*\*

1106 Mon Oct 27 12:37:08 2014

new/usr/src/lib/libm/common/m9x/ldexpf.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak ldexpf = __ldexpf
32 #endif
33
34 #include "libm.h"
35
36 float
37 ldexpf(float x, int n) {
38     return (scalbnf(x, n));
39 }
40
41 unchanged portion omitted
```

new/usr/src/lib/libm/common/m9x/ldexpl.c

1

\*\*\*\*\*

1118 Mon Oct 27 12:37:08 2014

new/usr/src/lib/libm/common/m9x/ldexpl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak ldexpl = __ldexpl
32 #endif
33
34 #include "libm.h"
35
36 long double
37 ldexpl(long double x, int n) {
38     return (scalbnl(x, n));
39 }
40
41 unchanged portion omitted
```

new/usr/src/lib/libm/common/m9x/llrint.c

1

\*\*\*\*\*

2284 Mon Oct 27 12:37:09 2014

new/usr/src/lib/libm/common/m9x/llrint.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak llrint = __llrint
32 #pragma weak lrint = __lrint
33 #pragma weak __lrint = __llrint
34 #endif
35 #endif
36
37 /*
38 * llrint(x) rounds its argument to the nearest integer according
39 * to the current rounding direction and converts the result to a
40 * 64 bit signed integer.
41 *
42 * If x is NaN, infinite, or so large that the nearest integer would
43 * exceed 64 bits, the invalid operation exception is raised. If x
44 * is not an integer, the inexact exception is raised.
45 */
46
47 #include "libm.h"
48
49 long long
50 llrint(double x) {
51     /*
52      * Note: The following code works on x86 (in the default rounding
53      * precision mode), but one should just use the fistpll instruction
54      * instead.
55      */
56     union {
57         unsigned i[2];
58         double d;
59     } xx, yy;
```

new/usr/src/lib/libm/common/m9x/llrint.c

2

```
59     unsigned hx;
60
61     xx.d = x;
62     hx = xx.i[HIWORD] & ~0x80000000;
63
64     if (hx < 0x43300000) { /* |x| < 2^52 */
65         /* add and subtract a power of two to round x to an integer */
66         #if defined(__sparc) || defined(__amd64)
67             yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43300000;
68         #elif defined(__i386) /* !defined(__amd64) */
69             yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43e00000;
70         #else
71             #error Unknown architecture
72         #endif
73         yy.i[LOWORD] = 0;
74         x = (x + yy.d) - yy.d;
75     }
76
77     /* now x is nan, inf, or integral */
78     return ((long long) x);
79 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/llrintf.c

1

```
*****
2000 Mon Oct 27 12:37:09 2014
new/usr/src/lib/libm/common/m9x/llrintf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak llrintf = __llrintf
31 #if defined(__sparcv9) || defined(__amd64)
32 #pragma weak lrintf = __llrintf
33 #pragma weak __lrintf = __llrintf
34 #endif
36 #endif

36 #include "libm.h"

38 long long
39 llrintf(float x) {
40     /*
41      * Note: The following code works on x86 (in the default rounding
42      * precision mode), but one should just use the fistpll instruction
43      * instead.
44      */
45     union {
46         unsigned i;
47         float f;
48     } xx, yy;
49     unsigned hx;

51     xx.f = x;
52     hx = xx.i & ~0x80000000;

54     if (hx < 0x4b000000) { /* |x| < 2^23 */
55         /* add and subtract a power of two to round x to an integer */
56 #if defined(__sparc) || defined(__amd64)
57         yy.i = (xx.i & 0x80000000) | 0x4b000000;
58 #elif defined(__i386)
```

new/usr/src/lib/libm/common/m9x/llrintf.c

2

```
59         /* assume 64-bit precision */
60         yy.i = (xx.i & 0x80000000) | 0x5f000000;
61 #else
62 #error Unknown architecture
63 #endif
64         x = (x + yy.f) - yy.f;

66     /*
67      * on LP32 architectures, we can just convert x to a 32-bit
68      * integer and sign-extend it
69      */
70     return ((long) x);
71 }

73     /* now x is nan, inf, or integral */
74     return ((long long) x);
75 }
_____ unchanged_portion_omitted
```

```

*****
4317 Mon Oct 27 12:37:10 2014
new/usr/src/lib/libm/common/m9x/llrintl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak llrintl = __llrintl
31 #if defined(__sparcv9) || defined(__amd64)
32 #pragma weak lrintl = __llrintl
33 #pragma weak __lrintl = __llrintl
34 #endif
36 #endif
36 #include "libm.h"
38 #if defined(__sparc)
40 #include "fma.h"
41 #include "fenv_inlines.h"
43 long long
44 llrintl(long double x) {
45     union {
46         unsigned i[4];
47         long double q;
48     } xx;
49     union {
50         unsigned i[2];
51         long long l;
52     } zz;
53     union {
54         unsigned i;
55         float f;
56     } tt;
57     unsigned int hx, sx, frac, fsr;
58     int rm, j;

```

```

59     volatile float dummy;
61     xx.q = x;
62     sx = xx.i[0] & 0x80000000;
63     hx = xx.i[0] & ~0x80000000;
65     /* handle trivial cases */
66     if (hx > 0x403e0000) { /* |x| > 2^63 + ... or x is nan */
67         /* convert an out-of-range float */
68         tt.i = sx | 0x7f000000;
69         return ((long long) tt.f);
70     } else if ((hx | xx.i[1] | xx.i[2] | xx.i[3]) == 0) /* x is zero */
71         return (0LL);
73     /* get the rounding mode */
74     __fenv_getfsr32(&fsr);
75     rm = fsr >> 30;
77     /* flip the sense of directed roundings if x is negative */
78     if (sx)
79         rm ^= rm >> 1;
81     /* handle |x| < 1 */
82     if (hx < 0x3fff0000) {
83         dummy = 1.0e30f; /* x is nonzero, so raise inexact */
84         dummy += 1.0e-30f;
85         if (rm == FSR_RP || (rm == FSR_RN && (hx >= 0x3ffe0000 &&
86             ((hx & 0xffff) | xx.i[1] | xx.i[2] | xx.i[3]))))
87             return (sx ? -1LL : 1LL);
88         return (0LL);
89     }
91     /* extract the integer and fractional parts of x */
92     j = 0x406f - (hx >> 16);
93     xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
94     if (j >= 96) {
95         zz.i[0] = 0;
96         zz.i[1] = xx.i[0] >> (j - 96);
97         frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
98         if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
99             frac |= 1;
100     } else if (j >= 64) {
101         zz.i[0] = xx.i[0] >> (j - 64);
102         zz.i[1] = ((xx.i[0] << 1) << (95 - j)) | (xx.i[1] >> (j - 64));
103         frac = ((xx.i[1] << 1) << (95 - j)) | (xx.i[2] >> (j - 64));
104         if (((xx.i[2] << 1) << (95 - j)) | xx.i[3])
105             frac |= 1;
106     } else {
107         zz.i[0] = ((xx.i[0] << 1) << (63 - j)) | (xx.i[1] >> (j - 32));
108         zz.i[1] = ((xx.i[1] << 1) << (63 - j)) | (xx.i[2] >> (j - 32));
109         frac = ((xx.i[2] << 1) << (63 - j)) | (xx.i[3] >> (j - 32));
110         if ((xx.i[3] << 1) << (63 - j))
111             frac |= 1;
112     }
114     /* round */
115     if (frac && (rm == FSR_RP || (rm == FSR_RN && (frac > 0x80000000u ||
116         (frac == 0x80000000u && (zz.i[1] & 1))))) {
117         if (++zz.i[1] == 0)
118             zz.i[0]++;
119     }
121     /* check for result out of range (note that z is |x| at this point) */
122     if (zz.i[0] > 0x80000000u || (zz.i[0] == 0x80000000u && (zz.i[1] |
123         !sx))) {
124         tt.i = sx | 0x7f000000;

```



```
125         return ((long long) tt.f);
126     }
127
128     /* raise inexact if need be */
129     if (frac) {
130         dummy = 1.0e30F;
131         dummy += 1.0e-30F;
132     }
133
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

```

*****
2189 Mon Oct 27 12:37:10 2014
new/usr/src/lib/libm/common/m9x/llround.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELF_OBJ)
30 #pragma weak llround = __llround
31 #if defined(__sparcv9) || defined(__amd64)
32 #pragma weak lround = __llround
33 #pragma weak __lround = __llround
34 #endif
36 #endif

36 /*
37 * llround(x) rounds its argument to the nearest integer, rounding
38 * ties away from zero, and converts the result to a 64 bit signed
39 * integer.
40 *
41 * If x is NaN, infinite, or so large that the nearest integer
42 * would exceed 64 bits, the invalid operation exception is raised.
43 */

45 #include "libm.h"

47 long long
48 llround(double x) {
49     union {
50         unsigned i[2];
51         double d;
52     } xx;
53     unsigned hx, sx, i;

55     xx.d = x;
56     hx = xx.i[HIWORD] & ~0x80000000;
57     sx = xx.i[HIWORD] & 0x80000000;

```

```

59     if (hx < 0x43300000) { /* |x| < 2^52 */
60         /* handle |x| < 1 */
61         if (hx < 0x3ff00000) {
62             if (hx >= 0x3fe00000)
63                 return (sx ? -1LL : 1LL);
64             return (0LL);
65         }

67         /* round x at the integer bit */
68         if (hx < 0x41300000) {
69             i = 1 << (0x412 - (hx >> 20));
70             xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
71             xx.i[LOWORD] = 0;
72         } else {
73             i = 1 << (0x432 - (hx >> 20));
74             xx.i[LOWORD] += i;
75             if (xx.i[LOWORD] < i)
76                 xx.i[HIWORD]++;
77             xx.i[LOWORD] &= ~(i | (i - 1));
78         }
79     }

81     /* now x is nan, inf, or integral */
82     return ((long long) xx.d);
83 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/m9x/llroundf.c

1

\*\*\*\*\*

1799 Mon Oct 27 12:37:11 2014  
new/usr/src/lib/libm/common/m9x/llroundf.c  
5262 libm needs to be carefully undef'd  
5268 libm doesn't need to hide symbols which are already local  
\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak llroundf = __llroundf
32 #pragma weak lroundf = __lroundf
33 #pragma weak __lroundf = __llroundf
34 #endif
35 #endif
36
37 #include "libm.h"
38
39 long long
40 llroundf(float x) {
41     union {
42         unsigned i;
43         float f;
44     } xx;
45     unsigned hx, sx, i;
46
47     xx.f = x;
48     hx = xx.i & ~0x80000000;
49     sx = xx.i & 0x80000000;
50
51     if (hx < 0x4b000000) { /* |x| < 2^23 */
52         /* handle |x| < 1 */
53         if (hx < 0x3f800000) {
54             if (hx >= 0x3f000000)
55                 return (sx ? -1LL : 1LL);
56             return (0LL);
57         }
58     }
59     /* round x at the integer bit */
```

new/usr/src/lib/libm/common/m9x/llroundf.c

2

```
59         i = 1 << (0x95 - (hx >> 23));
60         xx.i = (xx.i + i) & ~(i << 1) - 1);
61
62         /*
63          * on LP32 architectures, we can just convert x to a 32-bit
64          * integer and sign-extend it
65          */
66         return ((long) xx.f);
67     }
68
69     /* now x is nan, inf, or integral */
70     return ((long long) x);
71 }
72
73 unchanged_portion_omitted
```

```

*****
3823 Mon Oct 27 12:37:11 2014
new/usr/src/lib/libm/common/m9x/llroundl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak llroundl = __llroundl
31 #if defined(__sparcv9) || defined(__amd64)
32 #pragma weak lroundl = __llroundl
33 #pragma weak __lroundl = __llroundl
34 #endif
36 #endif
36 #include "libm.h"
38 #if defined(__sparc)
39 long long
40 llroundl(long double x) {
41     union {
42         unsigned i[4];
43         long double q;
44     } xx;
45     union {
46         unsigned i[2];
47         long long l;
48     } zz;
49     union {
50         unsigned i;
51         float f;
52     } tt;
53     unsigned hx, sx, frac;
54     int j;
56     xx.q = x;
57     sx = xx.i[0] & 0x80000000;
58     hx = xx.i[0] & ~0x80000000;

```

```

60     /* handle trivial cases */
61     if (hx > 0x403e0000) { /* |x| > 2^63 + ... or x is nan */
62         /* convert an out-of-range float */
63         tt.i = sx | 0x7f000000;
64         return ((long long) tt.f);
65     }
67     /* handle |x| < 1 */
68     if (hx < 0x3fff0000) {
69         if (hx >= 0x3ffe0000)
70             return (sx ? -1LL : 1LL);
71         return (0LL);
72     }
74     /* extract the integer and fractional parts of x */
75     j = 0x406f - (hx >> 16);
76     xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
77     if (j >= 96) {
78         zz.i[0] = 0;
79         zz.i[1] = xx.i[0] >> (j - 96);
80         frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
81         if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
82             frac |= 1;
83     } else if (j >= 64) {
84         zz.i[0] = xx.i[0] >> (j - 64);
85         zz.i[1] = ((xx.i[0] << 1) << (95 - j)) | (xx.i[1] >> (j - 64));
86         frac = ((xx.i[1] << 1) << (95 - j)) | (xx.i[2] >> (j - 64));
87         if (((xx.i[2] << 1) << (95 - j)) | xx.i[3])
88             frac |= 1;
89     } else {
90         zz.i[0] = ((xx.i[0] << 1) << (63 - j)) | (xx.i[1] >> (j - 32));
91         zz.i[1] = ((xx.i[1] << 1) << (63 - j)) | (xx.i[2] >> (j - 32));
92         frac = ((xx.i[2] << 1) << (63 - j)) | (xx.i[3] >> (j - 32));
93         if ((xx.i[3] << 1) << (63 - j))
94             frac |= 1;
95     }
97     /* round */
98     if (frac >= 0x80000000u) {
99         if (++zz.i[1] == 0)
100             zz.i[0]++;
101     }
103     /* check for result out of range (note that z is |x| at this point) */
104     if (zz.i[0] > 0x80000000u || (zz.i[0] == 0x80000000 && (zz.i[1] ||
105         !sx))) {
106         tt.i = sx | 0x7f000000;
107         return ((long long) tt.f);
108     }
110     /* negate result if need be */
111     if (sx) {
112         zz.i[0] = ~zz.i[0];
113         zz.i[1] = -zz.i[1];
114         if (zz.i[1] == 0)
115             zz.i[0]++;
116     }
118     return (zz.l);
119 }
_____unchanged_portion_omitted_____

```

new/usr/src/lib/libm/common/m9x/lrint.c

1

```
*****
2236 Mon Oct 27 12:37:12 2014
new/usr/src/lib/libm/common/m9x/lrint.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak lrint = __lrint
32 #endif

32 /*
33 * lrint(x) rounds its argument to the nearest integer according
34 * to the current rounding direction and converts the result to
35 * a 32 bit signed integer.
36 *
37 * If x is NaN, infinite, or so large that the nearest integer
38 * would exceed 32 bits, the invalid operation exception is raised.
39 * If x is not an integer, the inexact exception is raised.
40 */

42 #include <sys/isa_defs.h>      /* _ILP32 */
43 #include "libm.h"

45 #if defined(_ILP32)
46 long
47 lrint(double x) {
48     /*
49      * Note: The following code works on x86 (in the default rounding
50      * precision mode), but one should just use the fistpl instruction
51      * instead.
52      */
53     union {
54         unsigned i[2];
55         double d;
56     } xx, yy;
57     unsigned hx;
```

new/usr/src/lib/libm/common/m9x/lrint.c

2

```
59     xx.d = x;
60     hx = xx.i[HIWORD] & ~0x80000000;
61     if (hx < 0x43300000) { /* |x| < 2^52 */
62         /* add and subtract a power of two to round x to an integer */
63 #if defined(__sparc)
64         yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43300000;
65 #elif defined(__x86)
66         yy.i[HIWORD] = (xx.i[HIWORD] & 0x80000000) | 0x43e00000;
67 #else
68 #error Unknown architecture
69 #endif
70         yy.i[LOWORD] = 0;
71         x = (x + yy.d) - yy.d;
72     }

74     /* now x is nan, inf, or integral */
75     return ((long) x);
76 }
unchanged_portion_omitted
```

new/usr/src/lib/libm/common/m9x/lrintf.c

1

```
*****
1868 Mon Oct 27 12:37:12 2014
new/usr/src/lib/libm/common/m9x/lrintf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak lrintf = __lrintf
32 #endif

32 #include <sys/isa_defs.h> /* _ILP32 */
33 #include "libm.h"

35 #if defined(_ILP32)
36 long
37 lrintf(float x) {
38     /*
39      * Note: The following code works on x86 (in the default rounding
40      * precision mode), but one should just use the fistpl instruction
41      * instead.
42      */
43     union {
44         unsigned i;
45         float f;
46     } xx, yy;
47     unsigned hx;

49     xx.f = x;
50     hx = xx.i & ~0x80000000;
51     if (hx < 0x4b000000) { /* |x| < 2^23 */
52         /* add and subtract a power of two to round x to an integer */
53 #if defined(__sparc)
54         yy.i = (xx.i & 0x80000000) | 0x4b000000;
55 #elif defined(__x86)
56         /* assume 64-bit precision */
57         yy.i = (xx.i & 0x80000000) | 0x5f000000;
58 #else
```

new/usr/src/lib/libm/common/m9x/lrintf.c

2

```
59 #error Unknown architecture
60 #endif
61         x = (x + yy.f) - yy.f;
62         return ((long) x);
63     }

65     /* now x is nan, inf, or integral */
66     return ((long) x);
67 }

_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/lrintl.c

1

```
*****
3883 Mon Oct 27 12:37:12 2014
new/usr/src/lib/libm/common/m9x/lrintl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak lrintl = __lrintl
32 #endif
33
34 #include <sys/isa_defs.h> /* _ILP32 */
35 #include "libm.h"
36
37 #if defined(_ILP32)
38 #if defined(__sparc)
39 #include "fma.h"
40 #include "fenv_inlines.h"
41
42 long
43 lrintl(long double x) {
44     union {
45         unsigned int i[4];
46         long double q;
47     } xx;
48     union {
49         unsigned int i;
50         float f;
51     } tt;
52     unsigned int hx, sx, frac, l, fsr;
53     int rm, j;
54     volatile float dummy;
55
56     xx.q = x;
57     sx = xx.i[0] & 0x80000000;
58     hx = xx.i[0] & ~0x80000000;
```

new/usr/src/lib/libm/common/m9x/lrintl.c

2

```
59 /* handle trivial cases */
60 if (hx > 0x401e0000) { /* |x| > 2^31 + ... or x is nan */
61     /* convert an out-of-range float */
62     tt.i = sx | 0x7f000000;
63     return ((long) tt.f);
64 } else if ((hx | xx.i[1] | xx.i[2] | xx.i[3]) == 0) /* x is zero */
65     return (0L);
66
67 /* get the rounding mode */
68 __fenv_getfsr32(&fsr);
69 rm = fsr >> 30;
70
71 /* flip the sense of directed roundings if x is negative */
72 if (sx)
73     rm ^= rm >> 1;
74
75 /* handle |x| < 1 */
76 if (hx < 0x3fff0000) {
77     dummy = 1.0e30F; /* x is nonzero, so raise inexact */
78     dummy += 1.0e-30F;
79     if (rm == FSR_RP || (rm == FSR_RN && (hx >= 0x3ffe0000 &&
80         ((hx & 0xffff) | xx.i[1] | xx.i[2] | xx.i[3])))
81         return (sx ? -1L : 1L);
82     return (0L);
83 }
84
85 /* extract the integer and fractional parts of x */
86 j = 0x406f - (hx >> 16); /* 91 <= j <= 112 */
87 xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
88 if (j >= 96) { /* 96 <= j <= 112 */
89     l = xx.i[0] >> (j - 96);
90     frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
91     if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
92         frac |= 1;
93 } else { /* 91 <= j <= 95 */
94     l = (xx.i[0] << (96 - j)) | (xx.i[1] >> (j - 64));
95     frac = (xx.i[1] << (96 - j)) | (xx.i[2] >> (j - 64));
96     if ((xx.i[2] << (96 - j)) | xx.i[3])
97         frac |= 1;
98 }
99
100 /* round */
101 if (frac && (rm == FSR_RP || (rm == FSR_RN && (frac > 0x80000000U ||
102     (frac == 0x80000000U && (1 & 1)))))
103     l++;
104
105 /* check for result out of range (note that z is |x| at this point) */
106 if (l > 0x80000000U || (l == 0x80000000U && !sx)) {
107     tt.i = sx | 0x7f000000;
108     return ((long) tt.f);
109 }
110
111 /* raise inexact if need be */
112 if (frac) {
113     dummy = 1.0e30F;
114     dummy += 1.0e-30F;
115 }
116
117 /* negate result if need be */
118 if (sx)
119     l = -l;
120 return ((long) l);
121 }
122
123 unchanged_portion_omitted
```

```

*****
2169 Mon Oct 27 12:37:13 2014
new/usr/src/lib/libm/common/m9x/lround.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak lround = __lround
32 #endif

32 /*
33 * lround(x) rounds its argument to the nearest integer, rounding ties
34 * away from zero, and converts the result to a 32 bit signed integer.
35 *
36 * If x is NaN, infinite, or so large that the nearest integer
37 * would exceed 32 bits, the invalid operation exception is raised.
38 */

40 #include <sys/isa_defs.h> /* _ILP32 */
41 #include "libm.h"

43 #if defined(_ILP32)
44 long
45 lround(double x) {
46     union {
47         unsigned i[2];
48         double d;
49     } xx;
50     unsigned hx, sx, i;

52     xx.d = x;
53     hx = xx.i[HIWORD] & ~0x80000000;
54     sx = xx.i[HIWORD] & 0x80000000;
55     if (hx < 0x43300000) { /* |x| < 2^52 */
56         if (hx < 0x3ff00000) { /* |x| < 1 */
57             if (hx >= 0x3fe00000)
58                 return (sx ? -1L : 1L);

```

```

59         return (0L);
60     }

62     /* round x at the integer bit */
63     if (hx < 0x41300000) {
64         i = 1 << (0x412 - (hx >> 20));
65         xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
66         xx.i[LOWORD] = 0;
67     } else {
68         i = 1 << (0x432 - (hx >> 20));
69         xx.i[LOWORD] += i;
70         if (xx.i[LOWORD] < i)
71             xx.i[HIWORD]++;
72         xx.i[LOWORD] &= ~(i | (i - 1));
73     }
74 }

76 /* now x is nan, inf, or integral */
77 return ((long) xx.d);
78 }
_____unchanged_portion_omitted_____

```



new/usr/src/lib/libm/common/m9x/lroundf.c

1

```
*****
1675 Mon Oct 27 12:37:13 2014
new/usr/src/lib/libm/common/m9x/lroundf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak lroundf = __lroundf
32 #endif

32 #include <sys/isa_defs.h> /* _ILP32 */
33 #include "libm.h"

35 #if defined(_ILP32)
36 long
37 lroundf(float x) {
38     union {
39         unsigned i;
40         float f;
41     } xx;
42     unsigned hx, sx, i;

44     xx.f = x;
45     hx = xx.i & ~0x80000000;
46     sx = xx.i & 0x80000000;
47     if (hx < 0x4b000000) { /* |x| < 2^23 */
48         if (hx < 0x3f800000) { /* |x| < 1 */
49             if (hx >= 0x3f000000)
50                 return (sx ? -1L : 1L);
51             return (0L);
52         }

54         /* round x at the integer bit */
55         i = 1 << (0x95 - (hx >> 23));
56         xx.i = (xx.i + i) & ~((i << 1) - 1);
57         return ((long) xx.f);
58     }
}
```

new/usr/src/lib/libm/common/m9x/lroundf.c

2

```
60     /* now x is nan, inf, or integral */
61     return ((long) x);
62 }
_____unchanged_portion_omitted_____
```

\*\*\*\*\*

3372 Mon Oct 27 12:37:14 2014

new/usr/src/lib/libm/common/m9x/lroundl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```

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 #if defined(ELFOBJ)
30 #pragma weak lroundl = __lroundl
32 #endif

32 #include <sys/isa_defs.h> /* _ILP32 */
33 #include "libm.h"

35 #if defined(_ILP32)
36 #if defined(__sparc)
37 long
38 lroundl(long double x) {
39     union {
40         unsigned i[4];
41         long double q;
42     } xx;
43     union {
44         unsigned i;
45         float f;
46     } tt;
47     unsigned hx, sx, frac, l;
48     int j;

50     xx.q = x;
51     sx = xx.i[0] & 0x80000000;
52     hx = xx.i[0] & ~0x80000000;

54     /* handle trivial cases */
55     if (hx > 0x401e0000) { /* |x| > 2^31 + ... or x is nan */
56         /* convert an out-of-range float */
57         tt.i = sx | 0x7f000000;
58         return ((long) tt.f);

```

```

59     }

61     /* handle |x| < 1 */
62     if (hx < 0x3fff0000) {
63         if (hx >= 0x3ffe0000)
64             return (sx ? -1L : 1L);
65         return (0L);
66     }

68     /* extract the integer and fractional parts of x */
69     j = 0x406f - (hx >> 16); /* 91 <= j <= 112 */
70     xx.i[0] = 0x10000 | (xx.i[0] & 0xffff);
71     if (j >= 96) { /* 96 <= j <= 112 */
72         l = xx.i[0] >> (j - 96);
73         frac = ((xx.i[0] << 1) << (127 - j)) | (xx.i[1] >> (j - 96));
74         if (((xx.i[1] << 1) << (127 - j)) | xx.i[2] | xx.i[3])
75             frac |= 1;
76     } else { /* 91 <= j <= 95 */
77         l = (xx.i[0] << (96 - j)) | (xx.i[1] >> (j - 64));
78         frac = (xx.i[1] << (96 - j)) | (xx.i[2] >> (j - 64));
79         if ((xx.i[2] << (96 - j)) | xx.i[3])
80             frac |= 1;
81     }

83     /* round */
84     if (frac >= 0x80000000U)
85         l++;

87     /* check for result out of range (note that z is |x| at this point) */
88     if (l > 0x80000000U || (l == 0x80000000U && !sx)) {
89         tt.i = sx | 0x7f000000;
90         return ((long) tt.f);
91     }

93     /* negate result if need be */
94     if (sx)
95         l = -l;
96     return ((long) l);
97 }
_____unchanged_portion_omitted_____

```

```

*****
2414 Mon Oct 27 12:37:14 2014
new/usr/src/lib/libm/common/m9x/modf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak modf = __modf
31 #pragma weak _modf = __modf
33 #endif

33 /*
34 * modf(x, iptr) decomposes x into an integral part and a fractional
35 * part both having the same sign as x. It stores the integral part
36 * in *iptr and returns the fractional part.
37 *
38 * If x is infinite, modf sets *iptr to x and returns copysign(0.0,x).
39 * If x is NaN, modf sets *iptr to x and returns x.
40 *
41 * If x is a signaling NaN, this code does not attempt to raise the
42 * invalid operation exception.
43 */

45 #include "libm.h"

47 double
48 __modf(double x, double *iptr) {
49     union {
50         unsigned i[2];
51         double d;
52     } xx, yy;
53     unsigned hx, s;

55     xx.d = x;
56     hx = xx.i[HIWORD] & ~0x80000000;

58     if (hx >= 0x43300000) { /* x is NaN, infinite, or integral */

```

```

59         *iptr = x;
60         if (hx < 0x7ff00000 || (hx == 0x7ff00000 &&
61             xx.i[LOWORD] == 0)) {
62             xx.i[HIWORD] &= 0x80000000;
63             xx.i[LOWORD] = 0;
64         }
65         return (xx.d);
66     }

68     if (hx < 0x3ff00000) { /* |x| < 1 */
69         xx.i[HIWORD] &= 0x80000000;
70         xx.i[LOWORD] = 0;
71         *iptr = xx.d;
72         return (x);
73     }

75     /* split x at the binary point */
76     s = xx.i[HIWORD] & 0x80000000;
77     if (hx < 0x41400000) {
78         yy.i[HIWORD] = xx.i[HIWORD] & ~((1 << (0x413 - (hx >> 20))) -
79             1);
80         yy.i[LOWORD] = 0;
81     } else {
82         yy.i[HIWORD] = xx.i[HIWORD];
83         yy.i[LOWORD] = xx.i[LOWORD] & ~((1 << (0x433 - (hx >> 20))) -
84             1);
85     }
86     *iptr = yy.d;
87     xx.d -= yy.d;
88     xx.i[HIWORD] = (xx.i[HIWORD] & ~0x80000000) | s;
89
90     return (xx.d);
91 }
_____unchanged_portion_omitted_____

```

new/usr/src/lib/libm/common/m9x/modff.c

1

```
*****
1691 Mon Oct 27 12:37:14 2014
new/usr/src/lib/libm/common/m9x/modff.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak modff = __modff
31 #pragma weak _modff = __modff
33 #endif

33 #include "libm.h"

35 float
36 __modff(float x, float *iptr) {
37     union {
38         unsigned i;
39         float f;
40     } xx, yy;
41     unsigned hx, s;

43     xx.f = x;
44     hx = xx.i & ~0x80000000;

46     if (hx >= 0x4b000000) { /* x is NaN, infinite, or integral */
47         *iptr = x;
48         if (hx <= 0x7f800000)
49             xx.i &= 0x80000000;
50         return (xx.f);
51     }

53     if (hx < 0x3f800000) { /* |x| < 1 */
54         xx.i &= 0x80000000;
55         *iptr = xx.f;
56         return (x);
57     }
}
```

new/usr/src/lib/libm/common/m9x/modff.c

2

```
59     /* split x at the binary point */
60     s = xx.i & 0x80000000;
61     yy.i = xx.i & ~((1 << (0x96 - (hx >> 23))) - 1);
62     *iptr = yy.f;
63     xx.f -= yy.f;
64     xx.i = (xx.i & ~0x80000000) | s;
65     /* restore sign in case difference is 0 */
66     return (xx.f);
67 }
_____unchanged_portion_omitted_____
```

```

*****
3532 Mon Oct 27 12:37:15 2014
new/usr/src/lib/libm/common/m9x/modfl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak modfl = __modfl
32 #endif

32 #include "libm.h"

34 #if defined(__sparc)

36 long double
37 __modfl(long double x, long double *iptr) {
38     union {
39         unsigned i[4];
40         long double q;
41     } xx, yy;
42     unsigned hx, s;

44     xx.q = x;
45     hx = xx.i[0] & ~0x80000000;

47     if (hx >= 0x406f0000) { /* x is NaN, infinite, or integral */
48         *iptr = x;
49         if (hx < 0x7fff0000 || (hx == 0x7fff0000 &&
50             (xx.i[1] | xx.i[2] | xx.i[3]) == 0)) {
51             xx.i[0] &= 0x80000000;
52             xx.i[1] = xx.i[2] = xx.i[3] = 0;
53         }
54         return (xx.q);
55     }

57     if (hx < 0x3fff0000) { /* |x| < 1 */
58         xx.i[0] &= 0x80000000;

```

```

59         xx.i[1] = xx.i[2] = xx.i[3] = 0;
60         *iptr = xx.q;
61         return (x);
62     }

64     /* split x at the binary point */
65     s = xx.i[0] & 0x80000000;
66     if (hx < 0x40100000) {
67         yy.i[0] = xx.i[0] & ~((1 << (0x400f - (hx >> 16))) - 1);
68         yy.i[1] = yy.i[2] = yy.i[3] = 0;
69     } else if (hx < 0x40300000) {
70         yy.i[0] = xx.i[0];
71         yy.i[1] = xx.i[1] & ~((1 << (0x402f - (hx >> 16))) - 1);
72         yy.i[2] = yy.i[3] = 0;
73     } else if (hx < 0x40500000) {
74         yy.i[0] = xx.i[0];
75         yy.i[1] = xx.i[1];
76         yy.i[2] = xx.i[2] & ~((1 << (0x404f - (hx >> 16))) - 1);
77         yy.i[3] = 0;
78     } else {
79         yy.i[0] = xx.i[0];
80         yy.i[1] = xx.i[1];
81         yy.i[2] = xx.i[2];
82         yy.i[3] = xx.i[3] & ~((1 << (0x406f - (hx >> 16))) - 1);
83     }
84     *iptr = yy.q;

86     /*
87     * we could implement the following more efficiently than by using
88     * software emulation of fsubq, but we'll do it this way for now
89     * (and hope hardware support becomes commonplace)
90     */
91     xx.q -= yy.q;
92     xx.i[0] = (xx.i[0] & ~0x80000000) | s; /* keep sign of x */
93     return (xx.q);
94 }

unchanged_portion_omitted

```

new/usr/src/lib/libm/common/m9x/nan.c

1

```
*****
1500 Mon Oct 27 12:37:15 2014
new/usr/src/lib/libm/common/m9x/nan.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nan = __nan
32 #endif

32 /*
33 * nan(c) returns a NaN. This implementation ignores c.
34 */

36 #include "libm.h"
37 #include <sys/isa_defs.h>

39 #if defined(__sparc)

41 static const union {
42     unsigned i[2];
43     double d;
44 } __nan_union = { 0x7fffffff, 0xffffffff };
unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/m9x/nanf.c

1

```
*****  
1196 Mon Oct 27 12:37:16 2014  
new/usr/src/lib/libm/common/m9x/nanf.c  
5262 libm needs to be carefully undef'd  
5268 libm doesn't need to hide symbols which are already local  
*****
```

```
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 #if defined(ELFOBJ)  
30 #pragma weak nanf = __nanf  
32 #endif  
  
32 #include "libm.h"  
  
34 static const union {  
35     unsigned i;  
36     float f;  
37 } __nanf_union = { 0x7fffffff };  
_____  
unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/m9x/nanl.c

1

\*\*\*\*\*

1443 Mon Oct 27 12:37:17 2014

new/usr/src/lib/libm/common/m9x/nanl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak nanl = __nanl
32 #endif
33
34 #include "libm.h"
35
36 #if defined(__sparc)
37
38 static const union {
39     unsigned i[4];
40     long double ld;
41 } __nanl_union = { 0x7fffffff, 0xffffffff, 0xffffffff, 0xffffffff };
42
43 unchanged_portion_omitted

```



```

*****
3189 Mon Oct 27 12:37:17 2014
new/usr/src/lib/libm/common/m9x/nearbyint.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nearbyint = __nearbyint
32 #endif

32 /*
33 * nearbyint(x) returns the nearest fp integer to x in the direction
34 * corresponding to the current rounding direction without raising
35 * the inexact exception.
36 *
37 * nearbyint(x) is x unchanged if x is +/-0 or +/-inf. If x is NaN,
38 * nearbyint(x) is also NaN.
39 */

41 #include "libm.h"
42 #include "fenv_synonyms.h"
43 #include <fenv.h>

45 double
46 __nearbyint(double x) {
47     union {
48         unsigned i[2];
49         double d;
50     } xx;
51     unsigned hx, sx, i, frac;
52     int rm, j;

54     xx.d = x;
55     sx = xx.i[HIWORD] & 0x80000000;
56     hx = xx.i[HIWORD] & ~0x80000000;

58     /* handle trivial cases */

```

```

59     if (hx >= 0x43300000) { /* x is nan, inf, or already integral */
60         if (hx >= 0x7ff00000) /* x is inf or nan */
61             #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
62                 return (hx >= 0x7ff80000 ? x : x + x);
63                 /* assumes sparc-like QNaN */
64             #else
65                 return (x + x);
66             #endif
67         return (x);
68     } else if ((hx | xx.i[LOWORD]) == 0) /* x is zero */
69         return (x);

71     /* get the rounding mode */
72     rm = fegetround();

74     /* flip the sense of directed roundings if x is negative */
75     if (sx && (rm == FE_UPWARD || rm == FE_DOWNWARD))
76         rm = (FE_UPWARD + FE_DOWNWARD) - rm;

78     /* handle |x| < 1 */
79     if (hx < 0x3ff00000) {
80         if (rm == FE_UPWARD || (rm == FE_TONEAREST &&
81             (hx >= 0x3fe00000 && ((hx & 0xffff) | xx.i[LOWORD])))
82             xx.i[HIWORD] = sx | 0x3ff00000;
83         else
84             xx.i[HIWORD] = sx;
85         xx.i[LOWORD] = 0;
86         return (xx.d);
87     }

89     /* round x at the integer bit */
90     j = 0x433 - (hx >> 20);
91     if (j >= 32) {
92         i = 1 << (j - 32);
93         frac = ((xx.i[HIWORD] << 1) << (63 - j)) |
94             (xx.i[LOWORD] >> (j - 32));
95         if (xx.i[LOWORD] & (i - 1))
96             frac |= 1;
97         if (!frac)
98             return (x);
99         xx.i[LOWORD] = 0;
100        xx.i[HIWORD] &= ~(i - 1);
101        if ((rm == FE_UPWARD) || ((rm == FE_TONEAREST) &&
102            ((frac > 0x80000000u) || ((frac == 0x80000000) &&
103                (xx.i[HIWORD] & i))))
104            xx.i[HIWORD] += i;
105    } else {
106        i = 1 << j;
107        frac = (xx.i[LOWORD] << 1) << (31 - j);
108        if (!frac)
109            return (x);
110        xx.i[LOWORD] &= ~(i - 1);
111        if ((rm == FE_UPWARD) || ((rm == FE_TONEAREST) &&
112            (frac > 0x80000000u) || ((frac == 0x80000000) &&
113                (xx.i[LOWORD] & i)))) {
114            xx.i[LOWORD] += i;
115            if (xx.i[LOWORD] == 0)
116                xx.i[HIWORD]++;
117        }
118    }
119    return (xx.d);
120 }

```

\_\_\_\_\_unchanged\_portion\_omitted\_\_\_\_\_

```

*****
3997 Mon Oct 27 12:37:17 2014
new/usr/src/lib/libm/common/m9x/nearbyintf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nearbyintf = __nearbyintf
32 #endif

32 #include "libm.h"
33 #include "fenv_synonyms.h"
34 #include <fenv.h>

36 float
37 __nearbyintf(float x) {
38     union {
39         unsigned i;
40         float f;
41     } xx;
42     unsigned hx, sx, i, frac;
43     int rm;

45     xx.f = x;
46     sx = xx.i & 0x80000000;
47     hx = xx.i & ~0x80000000;

49     /* handle trivial cases */
50     if (hx >= 0x4b000000) { /* x is nan, inf, or already integral */
51         if (hx > 0x7f800000) /* x is nan */
52             return (x * x); /* + -> * for Cheetah */
53         return (x);
54     } else if (hx == 0) /* x is zero */
55         return (x);

57     /* get the rounding mode */
58     rm = fegetround();

```

```

60     /* flip the sense of directed roundings if x is negative */
61     if (sx && (rm == FE_UPWARD || rm == FE_DOWNWARD))
62         rm = (FE_UPWARD + FE_DOWNWARD) - rm;

64     /* handle |x| < 1 */
65     if (hx < 0x3f800000) {
66         if (rm == FE_UPWARD || (rm == FE_TONEAREST && hx > 0x3f000000))
67             xx.i = sx | 0x3f800000;
68         else
69             xx.i = sx;
70         return (xx.f);
71     }

73     /* round x at the integer bit */
74     i = 1 << (0x96 - (hx >> 23));
75     frac = hx & (i - 1);
76     if (!frac)
77         return (x);

79     hx &= ~(i - 1);
80     if (rm == FE_UPWARD || (rm == FE_TONEAREST && (frac > (i >> 1) ||
81         ((frac == (i >> 1)) && (hx & i)))))
82         xx.i = sx | (hx + i);
83     else
84         xx.i = sx | hx;
85     return (xx.f);
86 }
_____unchanged_portion_omitted_____

```

new/usr/src/lib/libm/common/m9x/nearbyintl.c

1

\*\*\*\*\*

4408 Mon Oct 27 12:37:18 2014

new/usr/src/lib/libm/common/m9x/nearbyintl.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak nearbyintl = __nearbyintl
32 #endif
33
32 #include "libm.h"
33 #include "fma.h"
34 #include "fenv_inlines.h"
35
36 #if defined(__sparc)
37
38 static union {
39     unsigned i;
40     float f;
41 } snan = { 0x7f800001 };
42
43 _____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/nexttoward.c

1

```
*****
4646 Mon Oct 27 12:37:18 2014
new/usr/src/lib/libm/common/m9x/nexttoward.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak nexttoward = __nexttoward
32 #endif

32 /*
33 * nexttoward(x, y) delivers the next representable number after x
34 * in the direction of y. If x and y are both zero, the result is
35 * zero with the same sign as y. If either x or y is NaN, the result
36 * is NaN.
37 *
38 * If x != y and the result is infinite, overflow is raised; if
39 * x != y and the result is subnormal or zero, underflow is raised.
40 * (This is wrong, but it's what C99 apparently wants.)
41 */

43 #include "libm.h"

45 #if defined(__sparc)

47 static union {
48     unsigned i[2];
49     double d;
50 } C[] = {
    unchanged_portion_omitted

```

new/usr/src/lib/libm/common/m9x/nexttowardf.c

1

\*\*\*\*\*

3656 Mon Oct 27 12:37:19 2014

new/usr/src/lib/libm/common/m9x/nexttowardf.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
31 #pragma weak nexttowardf = __nexttowardf
32 #endif
33
34 #include "libm.h"
35
36 static union {
37     unsigned i;
38     float f;
39 } C[] = {
40     unchanged_portion_omitted
41 }
```

```

*****
2735 Mon Oct 27 12:37:19 2014
new/usr/src/lib/libm/common/m9x/nexttowardl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
31 #pragma weak nexttowardl = __nexttowardl
32 #endif

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

35 #if defined(__sparc)
36 #define n0      0
37 #define n1      1
38 #define n2      2
39 #define n3      3
40 #define X86PDNRM1(x)
41 #define INC(px) { \
42     if (++px[n3] == 0) \
43         if (++px[n2] == 0) \
44             if (++px[n1] == 0) \
45                 ++px[n0]; \
46 }
47 #define DEC(px) { \
48     if (--px[n3] == 0xffffffff) \
49         if (--px[n2] == 0xffffffff) \
50             if (--px[n1] == 0xffffffff) \
51                 --px[n0]; \
52 }
53 #elif defined(__x86)
54 #define n0      2
55 #define n1      1
56 #define n2      0
57 #define n3      0
58 /*

```

```

59 * if pseudo-denormal, replace by the equivalent normal
60 */
61 #define X86PDNRM1(x)    if (XBIASED_EXP(x) == 0 && (((int *) &x)[1] & \
62     0x80000000) != 0) \
63     ((int *) &x)[2] |= 1
64 #define INC(px) { \
65     if (++px[n2] == 0) \
66         if (++px[n1] & ~0x80000000) == 0) \
67             px[n1] = 0x80000000, ++px[n0]; \
68 }
69 #define DEC(px) { \
70     if (--px[n2] == 0xffffffff) \
71         if (--px[n1] == 0x7fffffff) \
72             if (--px[n0] & 0x7fff) != 0) \
73                 px[n1] |= 0x80000000; \
74 }
75 #endif

77 long double
78 nexttowardl(long double x, long double y) {
79     int *px = (int *) &x;
80     int *py = (int *) &y;

82     if (x == y)
83         return (y);          /* C99 requirement */
84     if (x != x || y != y)
85         return (x * y);

87     if (ISZEROL(x)) {        /* x == 0.0 */
88         px[n0] = py[n0] & XSGNMSK;
89         px[n1] = px[n2] = 0;
90         px[n3] = 1;
91     } else {
92         X86PDNRM1(x);
93         if ((px[n0] & XSGNMSK) == 0) { /* x > 0.0 */
94             if (x > y) /* x > y */
95                 DEC(px)
96             else
97                 INC(px)
98         } else {
99             if (x < y) /* x < y */
100                 DEC(px)
101             else
102                 INC(px)
103         }
104     }
105 #ifndef lint
106     {
107         volatile long double dummy;
108         int k = XBIASED_EXP(x);

110         if (k == 0)
111             dummy = LDBL_MIN * copysign(LDBL_MIN, x);
112         else if (k == 0x7fff)
113             dummy = LDBL_MAX * copysign(LDBL_MAX, x);
114     }
115 #endif
116     return (x);
117 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/m9x/round.c

1

```
*****
1921 Mon Oct 27 12:37:19 2014
new/usr/src/lib/libm/common/m9x/round.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak round = __round
32 #endif

32 #include "libm.h"

34 double
35 round(double x) {
36     union {
37         unsigned i[2];
38         double d;
39     } xx;
40     unsigned hx, sx, i;

42     xx.d = x;
43     hx = xx.i[HIWORD] & ~0x80000000;
44     sx = xx.i[HIWORD] & 0x80000000;
45     if (hx < 0x43300000) { /* |x| < 2^52 */
46         if (hx < 0x3ff00000) { /* |x| < 1 */
47             if (hx >= 0x3fe00000)
48                 return (sx ? -1.0 : 1.0);
49             return (sx ? -0.0 : 0.0);
50         }

52         /* round x at the integer bit */
53         if (hx < 0x41300000) {
54             i = 1 << (0x412 - (hx >> 20));
55             xx.i[HIWORD] = (xx.i[HIWORD] + i) & ~(i | (i - 1));
56             xx.i[LOWORD] = 0;
57         } else {
58             i = 1 << (0x432 - (hx >> 20));
```

new/usr/src/lib/libm/common/m9x/round.c

2

```
59         xx.i[LOWORD] += i;
60         if (xx.i[LOWORD] < i)
61             xx.i[HIWORD]++;
62         xx.i[LOWORD] &= ~(i | (i - 1));
63     }
64     return (xx.d);
65 } else if (hx < 0x7ff00000)
66     return (x);
67 else
68 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
69     return (hx >= 0x7ff80000 ? x : x + x);
70 /* assumes sparc-like QNaN */
71 #else
72     return (x + x);
73 #endif
74 }
unchanged_portion_omitted_
```

new/usr/src/lib/libm/common/m9x/roundf.c

1

```
*****
1686 Mon Oct 27 12:37:20 2014
new/usr/src/lib/libm/common/m9x/roundf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak roundf = __roundf
32 #endif
33
32 #include "libm.h"
33
34 float
35 roundf(float x) {
36     union {
37         unsigned i;
38         float f;
39     } xx;
40     unsigned hx, sx, i;
41
42     xx.f = x;
43     hx = xx.i & ~0x80000000;
44     sx = xx.i & 0x80000000;
45     if (hx < 0x4b000000) { /* |x| < 2^23 */
46         if (hx < 0x3f800000) { /* |x| < 1 */
47             if (hx >= 0x3f000000)
48                 return (sx ? -1.0F : 1.0F);
49             return (sx ? -0.0F : 0.0F);
50         }
51
52         /* round x at the integer bit */
53         i = 1 << (0x95 - (hx >> 23));
54         xx.i = (xx.i + i) & ~(i << 1) - 1;
55         return (xx.f);
56     } else if (hx < 0x7f800000) /* |x| is integral */
57         return (x);
58     else
```

new/usr/src/lib/libm/common/m9x/roundf.c

2

```
59 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
60     return (hx > 0x7f800000 ? x * x : x + x);
61 #else
62     return (x + x);
63 #endif
64 }
unchanged_portion_omitted
```



```

*****
3658 Mon Oct 27 12:37:20 2014
new/usr/src/lib/libm/common/m9x/roundl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak roundl = __roundl
32 #endif

32 #include "libm.h"

34 #if defined(__sparc)
35 long double
36 roundl(long double x) {
37     union {
38         unsigned i[4];
39         long double q;
40     } xx;
41     unsigned hx, sx, v;
42     int j;

44     xx.q = x;
45     sx = xx.i[0] & 0x80000000;
46     hx = xx.i[0] & ~0x80000000;

48     /* handle trivial cases */
49     if (hx >= 0x406f0000) /* |x| >= 2^112 + ... or x is nan */
50         return (hx >= 0x7fff0000 ? x + x : x);

52     /* handle |x| < 1 */
53     if (hx < 0x3fff0000) {
54         if (hx >= 0x3ffe0000)
55             return (sx ? -1.0L : 1.0L);
56         return (sx ? -0.0L : 0.0L);
57     }

```

```

59     xx.i[0] = hx;
60     j = 0x406f - (hx >> 16); /* 1 <= j <= 112 */
61     if (j >= 96) { /* 96 <= j <= 112 */
62         v = (1U << (j - 96)) >> 1;
63         if (v) {
64             if (xx.i[0] & v)
65                 xx.i[0] += v;
66             xx.i[0] &= ~(v - 1);
67         } else if (xx.i[1] & 0x80000000)
68             ++xx.i[0];
69         xx.i[1] = xx.i[2] = xx.i[3] = 0;
70     } else if (j >= 64) { /* 64 <= j <= 95 */
71         v = (1U << (j - 64)) >> 1;
72         if (v) {
73             if (xx.i[1] & v) {
74                 xx.i[1] += v;
75                 if (xx.i[1] < v)
76                     ++xx.i[0];
77             }
78             xx.i[1] &= ~(v - 1);
79         } else if (xx.i[2] & 0x80000000) {
80             if (++xx.i[1] == 0)
81                 ++xx.i[0];
82         }
83         xx.i[2] = xx.i[3] = 0;
84     } else if (j >= 32) { /* 32 <= j <= 63 */
85         v = (1U << (j - 32)) >> 1;
86         if (v) {
87             if (xx.i[2] & v) {
88                 xx.i[2] += v;
89                 if (xx.i[2] < v) {
90                     if (++xx.i[1] == 0)
91                         ++xx.i[0];
92                 }
93             }
94             xx.i[2] &= ~(v - 1);
95         } else if (xx.i[3] & 0x80000000) {
96             if (++xx.i[2] == 0) {
97                 if (++xx.i[1] == 0)
98                     ++xx.i[0];
99             }
100         }
101         xx.i[3] = 0;
102     } else { /* 1 <= j <= 31 */
103         v = 1U << (j - 1);
104         if (xx.i[3] & v) {
105             xx.i[3] += v;
106             if (xx.i[3] < v) {
107                 if (++xx.i[2] == 0) {
108                     if (++xx.i[1] == 0)
109                         ++xx.i[0];
110                 }
111             }
112         }
113         xx.i[3] &= ~(v - 1);
114     }

116     /* negate result if need be */
117     if (sx)
118         xx.i[0] |= 0x80000000;
119     return (xx.q);
120 }

unchanged_portion_omitted
162 #else
163 #error Unknown architecture
164 #endif /* defined(__sparc) || defined(__x86) */

```

```

*****
2659 Mon Oct 27 12:37:21 2014
new/usr/src/lib/libm/common/m9x/scalbln.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak scalbln = __scalbln
32 #endif

32 #include "libm.h"
33 #include <float.h>          /* DBL_MAX, DBL_MIN */

35 static const double twom54 = 5.5511151231257827021181583404541015625e-17;
36 #if defined(__x86)
38 #if defined(USE_FPSCALE) || defined(__x86)
37 static const double two52 = 4503599627370496.0;
38 #else
39 /*
40  * Normalize non-zero subnormal x and return biased exponent of x in [-51,0]
41  */
42 static int
43 ilogb_biased(unsigned *px) {
44     int s = 52;
45     unsigned v = px[HIWORD] & ~0x80000000, w = px[LOWORD], t = v;

47     if (t)
48         s -= 32;
49     else
50         t = w;
51     if (t & 0xffff0000)
52         s -= 16, t >>= 16;
53     if (t & 0xff00)
54         s -= 8, t >>= 8;
55     if (t & 0xf0)
56         s -= 4, t >>= 4;
57     t <<= 1;

```

```

58     s -= (0xffffaa50 >> t) & 0x3;
59     if (s < 32) {
60         v = (v << s) | w >> (32 - s);
61         w <<= s;
62     } else {
63         v = w << (s - 32);
64         w = 0;
65     }
66     px[HIWORD] = (px[HIWORD] & 0x80000000) | v;
67     px[LOWORD] = w;
68     return (1 - s);
69 }
70 #endif /* defined(__x86) */
72 #endif /* defined(USE_FPSCALE) */

72 double
73 scalbln(double x, long n) {
74     int *px = (int *) &x, ix, k;

76     ix = px[HIWORD] & ~0x80000000;
77     k = ix >> 20;
78     if (k == 0x7ff)
79 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
80         return ((px[HIWORD] & 0x80000) != 0 ? x : x + x);
81         /* assumes sparc-like QNaN */
82 #else
83         return (x + x);
84 #endif
85     if ((px[LOWORD] | ix) == 0 || n == 0)
86         return (x);
87     if (k == 0) {
88 #if defined(__x86)
90 #if defined(USE_FPSCALE) || defined(__x86)
89         x *= two52;
90         k = ((px[HIWORD] & ~0x80000000) >> 20) - 52;
91 #else
92         k = ilogb_biased((unsigned *) px);
93 #endif
94     }
95     k += (int) n;
96     if (n > 5000 || k > 0x7fe)
97         return (DBL_MAX * copysign(DBL_MAX, x));
98     if (n < -5000 || k <= -54)
99         return (DBL_MIN * copysign(DBL_MIN, x));
100     if (k > 0) {
101         px[HIWORD] = (px[HIWORD] & ~0x7ff00000) | (k << 20);
102         return (x);
103     }
104     k += 54;
105     px[HIWORD] = (px[HIWORD] & ~0x7ff00000) | (k << 20);
106     return (x * twom54);
107 }

```

unchanged\_portion\_omitted

new/usr/src/lib/libm/common/m9x/scalblnf.c

1

```
*****
2273 Mon Oct 27 12:37:22 2014
new/usr/src/lib/libm/common/m9x/scalblnf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak scalblnf = __scalblnf
32 #endif
33
34 #include "libm.h"
35 #include <float.h>          /* FLT_MAX, FLT_MIN */
36
37 static const float twom25f = 2.98023223876953125e-8F;
38 #if defined(__x86)
38 #if defined(USE_FPSCALE) || defined(__x86)
37 static const float two23f = 8388608.0F;
38 #else
39 /*
40  * v: a non-zero subnormal |x|; returns [-22, 0]
41  */
42 static int
43 ilogbf_biased(unsigned v) {
44     int r = -22;
45
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 }
55 #endif /* defined(__x86) */
57 #endif /* defined(USE_FPSCALE) */
```

new/usr/src/lib/libm/common/m9x/scalblnf.c

2

```
57 float
58 scalblnf(float x, long n) {
59     int *px = (int *) &x, ix, k;
60
61     ix = *px & ~0x80000000;
62     k = ix >> 23;
63     if (k == 0xff)
64 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
65         return (ix > 0x7f800000 ? x * x : x);
66 #else
67         return (x + x);
68 #endif
69     if (ix == 0 || n == 0)
70         return (x);
71     if (k == 0) {
72 #if defined(__x86)
74 #if defined(USE_FPSCALE) || defined(__x86)
73         x *= two23f;
74         k = ((*px & ~0x80000000) >> 23) - 23;
75 #else
76         k = ilogbf_biased(ix);
77         *px = (*px & 0x80000000) | (ix << (-k + 1));
78 #endif
79     }
80     k += (int) n;
81     if (n > 5000 || k > 0xfe)
82         return (FLT_MAX * copysignf(FLT_MAX, x));
83     if (n < -5000 || k <= -25)
84         return (FLT_MIN * copysignf(FLT_MIN, x));
85     if (k > 0) {
86         *px = (*px & ~0x7f800000) | (k << 23);
87         return (x);
88     }
89     k += 25;
90     *px = (*px & ~0x7f800000) | (k << 23);
91     return (x * twom25f);
92 }
93
94 _____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/scalblnl.c

1

```
*****
2404 Mon Oct 27 12:37:22 2014
new/usr/src/lib/libm/common/m9x/scalblnl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak scalblnl = __scalblnl
32 #endif

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

35 #if defined(__sparc)
36 #define XSET_EXP(k, x) (((int *) &x)[0] = (((int *) &x)[0] & ~0x7fff0000) | \
37                      (k << 16))
38 #define ISINFNANL(k, x) (k == 0x7fff)
39 #define XTWOT_OFFSET 113
40 static const long double xtwtot = 10384593717069655257060992658440192.0L,
41                      /* 2^113 */
42                      twomt1 = 4.814824860968089632639944856462318296E-35L; /* 2^-114 */
43 #elif defined(__x86)
44 #define XSET_EXP(k, x) (((int *) &x)[2] = (((int *) &x)[2] & ~0x7fff) | k
45 #if defined(HANDLE_UNSUPPORTED)
46 #define ISINFNANL(k, x) (k == 0x7fff || \
47                      (k != 0 && (((int *) &x)[1] & 0x80000000) == 0))
48 #else
49 #define ISINFNANL(k, x) (k == 0x7fff)
50 #endif
51 #define XTWOT_OFFSET 64
52 static const long double xtwtot = 18446744073709551616.0L, /* 2^64 */
53                      twomt1 = 2.7105054312137610850186E-20L; /* 2^-65 */
54 #endif

56 long double
57 scalblnl(long double x, long n) {
58     int k = XBIASED_EXP(x);
```

new/usr/src/lib/libm/common/m9x/scalblnl.c

2

```
60     if (ISINFNANL(k, x))
61         return (x + x);
62     if (ISZEROL(x) || n == 0)
63         return (x);
64     if (k == 0) {
65         x *= xtwtot;
66         k = XBIASED_EXP(x) - XTWOT_OFFSET;
67     }
68     k += (int) n;
69     if (n > 50000 || k > 0x7ffe)
70         return (LDBL_MAX * copysignl(LDBL_MAX, x));
71     if (n < -50000 || k <= -XTWOT_OFFSET - 1)
72         return (LDBL_MIN * copysignl(LDBL_MIN, x));
73     if (k > 0) {
74         XSET_EXP(k, x);
75         return (x);
76     }
77     k += XTWOT_OFFSET + 1;
78     XSET_EXP(k, x);
79     return (x * twomt1);
80 }
_____unchanged_portion_omitted_____
```

```

*****
68964 Mon Oct 27 12:37:22 2014
new/usr/src/lib/libm/common/m9x/tgamma.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak tgamma = __tgamma
32 #endif

32 /* INDENT OFF */
33 /*
34  * True gamma function
35  * double tgamma(double x)
36  *
37  * Error:
38  * -----
39  * Less than one ulp for both positive and negative arguments.
40  *
41  * Algorithm:
42  * -----
43  * A: For negative argument
44  * (1) gamma(-n or -inf) is NaN
45  * (2) Underflow Threshold
46  * (3) Reduction to gamma(1+x)
47  *
48  * B: For x between 1 and 2
49  * C: For x between 0 and 1
50  * D: For x between 2 and 8
51  * E: Overflow threshold {see over.c}
52  * F: For overflow_threshold >= x >= 8
53  *
54  * Implementation details
55  * -----
56  * (A) For negative argument, use gamma(-x) = -----
57  * (sin(pi*x)*gamma(1+x))
58  *

```

```

59 * (1) gamma(-n or -inf) is NaN with invalid signal by SUSv3 spec.
60 * (Ideally, gamma(-n) = 1/sinpi(n) = (-1)**(n+1) * inf.)
61 *
62 * (2) Underflow Threshold. For each precision, there is a value T
63 * such that when x>T and when x is not an integer, gamma(-x) will
64 * always underflow. A table of the underflow threshold value is given
65 * below. For proof, see file "under.c".
66 *
67 * Precision underflow threshold T =
68 * -----
69 * single 41.000041962 = 41 + 11 ULP
70 * (machine format) 4224000B
71 * double 183.000000000000312639 = 183 + 11 ULP
72 * (machine format) 4066E000 0000000B
73 * quad 1774.000000000000000000000000000017749370 = 1774 + 9 ULP
74 * (machine format) 4009BB80000000000000000000000000
75 * -----
76 *
77 * (3) Reduction to gamma(1+x).
78 * Because of (1) and (2), we need only consider non-integral x
79 * such that 0<x<T. Let k = [x] and z = x-[x]. Define
80 * sin(x*pi) cos(x*pi)
81 * kpsin(x) = ----- and kpcos(x) = ----- . Then
82 * pi pi
83 *
84 * gamma(-x) = -----
85 * -kpsin(x)*gamma(1+x)
86 *
87 * Since x = k+z,
88 * -sin(x*pi) = -sin(k*pi+z*pi) = (-1)^(k+1) * sin(z*pi),
89 * k+1
90 * we have -kpsin(x) = (-1) * kpsin(z). We can further
91 * reduce z to t by
92 * (I) t = z when 0.00000 <= z < 0.31830...
93 * (II) t = 0.5-z when 0.31830... <= z < 0.681690...
94 * (III) t = 1-z when 0.681690... <= z < 1.00000
95 * and correspondingly
96 * (I) kpsin(z) = kpsin(t) ... 0<= z < 0.3184
97 * (II) kpsin(z) = kpcos(t) ... |t| < 0.182
98 * (III) kpsin(z) = kpsin(t) ... 0<= t < 0.3184
99 *
100 * Using a special Remez algorithm, we obtain the following polynomial
101 * approximation for kpsin(t) for 0<=t<0.3184:
102 *
103 * Computation note: in simulating higher precision arithmetic, kcpsin
104 * return head = t and tail = ks[0]*t^3 + (...) to maintain extra bits.
105 *
106 * Quad precision, remez error <= 2**(-129.74)
107 * 3 5
108 * kpsin(t) = t + ks[0] * t + ks[1] * t + ... + ks[12] * t 27
109 *
110 * ks[ 0] = -1.64493406684822643647241516664602518705158902870e+0000
111 * ks[ 1] = 8.11742425283353643637002772405874238094995726160e-0001
112 * ks[ 2] = -1.90751824122084213696472111835337366232282723933e-0001
113 * ks[ 3] = 2.61478478176548005046532613563241288115395517084e-0002
114 * ks[ 4] = -2.34608103545582363750893072647117829448016479971e-0003
115 * ks[ 5] = 1.48428793031071003684606647212534027556262040158e-0004
116 * ks[ 6] = -6.97587366165638046518462722252768122615952898698e-0006
117 * ks[ 7] = 2.53121740413702536928659271747187500934840057929e-0007
118 * ks[ 8] = -7.30471182221385990397683641695766121301933621956e-0009
119 * ks[ 9] = 1.71653847451163495739958249695549313987973589884e-0010
120 * ks[10] = -3.34813314714560776122245796929054813458341420565e-0012
121 * ks[11] = 5.50724992262622033449487808306969135431411753047e-0014
122 * ks[12] = -7.67678132753577998601234393215802221104236979928e-0016
123 *
124 * Double precision, Remez error <= 2**(-62.9)

```

```

125 *
126 *      kpsin(t) = t + ks[0] * t3 + ks[1] * t5 + ... + ks[6] * t15
127 *
128 *      ks[0] = -1.644934066848226406065691 (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 *
138 *      kpsin(t) = t + ks[0] * t3 + ks[1] * t5 + ... + ks[3] * t9
139 *
140 *      ks[0] = -1.64493404985645811354476665052005342839447790544e+0000
141 *      ks[1] = 8.117407944583510640927972490694382693673389272270e-0001
142 *      ks[2] = -1.90703144603551216933075809162889536878854055202e-0001
143 *      ks[3] = 2.55742333994264563281155312271481108635575331201e-0002
144 *
145 *      Computation note: in simulating higher precision arithmetic, kpsin
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 *
153 *      kpcos(t) = 1/pi + pi/2 * t2 + kc[2] * t4 + ... + kc[11] * t22
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 *
168 *      kpcos(t) = 1/pi + pi/2 * t2 + kc[2] * t4 + ... + kc[6] * t12
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] = 6.104113568295154145755665647336325323339041115968e-0004
175 *
176 *      Single precision, remez < 2**(-30.13)
177 *
178 *      kpcos(t) = kc[0] + kc[1] * t2 + ... + kc[3] * t6
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, kpcos
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 *
194 *      k+1
195 *      (-1)
196 *      gamma(-x) = ----- ;
197 *      kpsin(z)*gamma(1+x)
198 *
199 *      otherwise, for z in (II),
200 *
201 *      k+1
202 *      (-1)
203 *      gamma(-x) = ----- ;
204 *      kpcos(0.5-z)*gamma(1+x)
205 *
206 *      otherwise, for z in (III),
207 *
208 *      k+1
209 *      (-1)
210 *      gamma(-x) = ----- .
211 *      kpsin(1-z)*gamma(1+x)
212 *
213 *      Thus, the computation of gamma(-x) reduced to the computation of
214 *      gamma(1+x) and kpsin(), kpcos().
215 *
216 *      (B) For x between 1 and 2. We break [1,2] into three parts:
217 *      GT1 = [1.0000, 1.2845]
218 *      GT2 = [1.2844, 1.6374]
219 *      GT3 = [1.6373, 2.0000]
220 *
221 *      For x in GTi, i=1,2,3, let
222 *      z1 = 1.134861805732790769689793935774652917006
223 *      gz1 = gamma(z1) = 0.9382046279096824494097535615803269576988
224 *      tz1 = gamma'(z1) = -0.3517214357852935791015625000000000000000
225 *
226 *      z2 = 1.461632144968362341262659542325721328468e+0000
227 *      gz2 = gamma(z2) = 0.8856031944108887002788159005825887332080
228 *      tz2 = gamma'(z2) = 0.00
229 *
230 *      z3 = 1.819773101100500601787868704921606996312e+0000
231 *      gz3 = gamma(z3) = 0.9367814114636523216188468970808378497426
232 *      tz3 = gamma'(z3) = 0.2805306315422058105468750000000000000000
233 *
234 *      and
235 *      y = x-zi ... for extra precision, write y = y.h + y.l
236 *      Then
237 *      gamma(x) = gzi + tzi*(y.h+y.l) + y*y*Ri(y),
238 *      = gzi.h + (tzi*y.h + ((tzi*y.l+gzi.l) + y*y*Ri(y)))
239 *      = gy.h + gy.l
240 *
241 *      where
242 *      (I) For double precision
243 *
244 *      Ri(y) = Pi(y)/Qi(y), i=1,2,3;
245 *
246 *      P1(y) = p1[0] + p1[1]*y + ... + p1[4]*y^4
247 *      Q1(y) = q1[0] + q1[1]*y + ... + q1[5]*y^5
248 *
249 *      P2(y) = p2[0] + p2[1]*y + ... + p2[3]*y^3
250 *      Q2(y) = q2[0] + q2[1]*y + ... + q2[6]*y^6
251 *
252 *      P3(y) = p3[0] + p3[1]*y + ... + p3[4]*y^4
253 *      Q3(y) = q3[0] + q3[1]*y + ... + q3[5]*y^5
254 *
255 *      Remez precision of Ri(y):
256 *      |gamma(x)-(gzi+tzi*y) - y*y*Ri(y)| <= 2**(-62.3 ... for i = 1
257 *      <= 2**(-59.4 ... for i = 2
258 *      <= 2**(-62.1 ... for i = 3
259 *
260 *      (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 *      Coefficients: Single precision
299 *      i = 1:
300 *      P1[0] = 7.09087253435088360271451613398019280077561279443e-0001
301 *      P1[1] = -5.17229560788652108545141978238701790105241761089e-0001
302 *      P1[2] = 5.23403394528150789405825222323770647162337764327e-0001
303 *      P1[3] = -4.54586308717075010784041566069480411732634814899e-0001
304 *      P1[4] = 4.20596490915239085459964590559256913498190955233e-0001
305 *      P1[5] = -3.57307589712377520978332185838241458642142185789e-0001
306 *
307 *      i = 2:
308 *      p2[0] = 4.28486983980295198166056119223984284434264344578e-0001
309 *      p2[1] = -1.30704539487709138528680121627899735386650103914e-0001
310 *      p2[2] = 1.60856285038051955072861219352655851542955430871e-0001
311 *      p2[3] = -9.22285161346010583774458802067371182158937943507e-0002
312 *      p2[4] = 7.19240511767225260740890292605070595560626179357e-0002
313 *      p2[5] = -4.88158265593355093703112238534484636193260459574e-0002
314 *
315 *      i = 3
316 *      p3[0] = 3.82409531118807759081121479786092134814808872880e-0001
317 *      p3[1] = 2.65309888180188647956400403013495759365167853426e-0002
318 *      p3[2] = 8.06815109775079171923561169415370309376296739835e-0002
319 *      p3[3] = -1.54821591666137613928840890835174351674007764799e-0002
320 *      p3[4] = 1.76308239242717268530498313416899188157165183405e-0002
321 *
322 *      Coefficients: Double precision

```

```

323 *      i = 1:
324 *      p1[0] = 0.70908683619977797008004927192814648151397705078125000
325 *      p1[1] = 1.71987061393048558089579513384356441668351720061e-0001
326 *      p1[2] = -3.19273345791990970293320316122813960527705450671e-0002
327 *      p1[3] = 8.36172645419110036267169600390549973563534476989e-0003
328 *      p1[4] = 1.13745336648572838333152213474277971244629758101e-0003
329 *      q1[0] = 1.0
330 *      q1[1] = 9.71980217826032937526460731778472389791321968082e-0001
331 *      q1[2] = -7.43576743326756176594084137256042653497087666030e-0002
332 *      q1[3] = -1.19345944932265559769719470515102012246995255372e-0001
333 *      q1[4] = 1.5991344575142500262093512047078138221505028472e-0002
334 *      q1[5] = 1.12601136853374984566572691306402321911547550783e-0003
335 *
336 *      i = 2:
337 *      p2[0] = 0.42848681585558601181418225678498856723308563232421875
338 *      p2[1] = 6.53596762668970816023718845105667418483122103629e-0002
339 *      p2[2] = -6.97280829631212931321050770925128264272768936731e-0003
340 *      p2[3] = 6.46342359021981718947208605674813260166116632899e-0003
341 *      q2[0] = 1.0
342 *      q2[1] = 4.57572620560506047062553957454062012327519313936e-0001
343 *      q2[2] = -2.52182594886075452859655003407796103083422572036e-0001
344 *      q2[3] = -1.82970945407778594681348166040103197178711552827e-0002
345 *      q2[4] = 2.43574726993169566475227642128830141304953840502e-0002
346 *      q2[5] = -5.20390406466942525358645957564897411258667085501e-0003
347 *      q2[6] = 4.79520251383279837635552431988023256031951133885e-0004
348 *
349 *      i = 3:
350 *      p3[0] = 0.382409479734567459008331979930517263710498809814453125
351 *      p3[1] = 1.42876048697668161599069814043449301572928034140e-0001
352 *      p3[2] = 3.42157571052250536817923866013561760785748899071e-0003
353 *      p3[3] = -5.01542621710067521405087887856991700987709272937e-0004
354 *      p3[4] = 8.89285814866740910123834688163838287618332122670e-0004
355 *      q3[0] = 1.0
356 *      q3[1] = 3.0425308662944420100221564094895789790629963168e-0001
357 *      q3[2] = -2.2316240737999947728255672834881213873185520006e-0001
358 *      q3[3] = -1.0506086774195206592180981193367013142752903636e-0002
359 *      q3[4] = 1.70511763916186982473301861980856352005926669320e-0002
360 *      q3[5] = -2.12950201683609187927899416700094630764182477464e-0003
361 *
362 *      Note that all pi0 are exact in double, which is obtained by a
363 *      special Remez Algorithm.
364 *
365 *      Coefficients: Quad precision
366 *      i = 1:
367 *      p1[0] = 0.709086836199777919037185741507610124611513720557
368 *      p1[1] = 4.45754781206489035827195969367354835667391606951e-0001
369 *      p1[2] = 3.21049298735832382311662273882632210062918153852e-0002
370 *      p1[3] = -5.71296796342106617651765245858289197369688864350e-0003
371 *      p1[4] = 6.04666892891998977081619174969855831606965352773e-0003
372 *      p1[5] = 8.9910618699688871193967812174765258822658645168e-0004
373 *      p1[6] = -6.9649846144407741431207008527018441810175568949e-0005
374 *      p1[7] = 1.52597046118984020814225409300131445070213882429e-0005
375 *      p1[8] = 5.68521076168495673844711465407432189190681541547e-0007
376 *      p1[9] = 3.30749673519634895220582062520286565610418952979e-0008
377 *      q1[0] = 1.0+0000
378 *      q1[1] = 1.35806511721671070408570853537257079579490650668e+0000
379 *      q1[2] = 2.97567810153429553405327140096063086994072952961e-0001
380 *      q1[3] = -1.52956835982588571502954372821681851681118097870e-0001
381 *      q1[4] = -2.88248519561420109768781615289082053597954521218e-0002
382 *      q1[5] = 1.034753117199374052197899484563139363023748295955e-0002
383 *      q1[6] = 4.123102032438912223689653601243912973782742313e-0004
384 *      q1[7] = -3.1265370815229086724893192512038072918332507388e-0004
385 *      q1[8] = 2.36672170850409745237358105667757760527014332458e-0005
386 *
387 *      i = 2:
388 *      p2[0] = 0.428486815855585429730209907810650616737756697477
389 *      p2[1] = 2.63622124067885222919192651151581541943362617352e-0001
390 *      p2[2] = 3.85520683670028865731877276741390421744971446855e-0002

```

```

389 *      p2[3] = 3.05065978278128549958897133190295325258023525862e-0003
390 *      p2[4] = 2.48232934951723128892080415054084339152450445081e-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] = -1.245383375858993004944444600248687901947684291683e-0001
400 *      q2[4] = 4.49866050763472358547524708431719114204535491412e-0003
401 *      q2[5] = 7.20715455697920560621638325356292640604078591907e-0003
402 *      q2[6] = -8.68513169029126780280798337091982780598228096116e-0004
403 *      q2[7] = -1.251044316294011815250270982227495544809974229874e-0004
404 *      q2[8] = 3.1055834883900003848919130455099804752125347464e-0005
405 *      q2[9] = -1.76829227852852176018537139573609433652506765712e-0006
406 *
407 *      i = 3
408 *      p3[0] = 0.382409479734567504850247661075355640070439388902
409 *      p3[1] = 3.4219809307661849541585490633590842715983337774e-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] = 3.48309693970985612644446415789230015515365291459e-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] = -1.820965538628223466101095220151295856933534348322e-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 *      
$$\frac{1}{x} - \frac{1}{2} \text{ rounded to nearest.}$$

432 *      (1) For  $0 \leq x \leq 2^{-P}$ ,  $\gamma(x)$  is computed by  $\frac{1}{x}$  rounded to nearest.
433 *      The error is bound by  $0.739 \text{ ulp}(\gamma(x))$  in IEEE double precision.
434 *      Proof.
435 *      Since  $\frac{1}{x} \sim \frac{1}{2} + 0.577 \dots x^{-2} - \dots$ , we have, for small  $x$ ,
436 *      
$$\frac{1}{x(1+0.578x)} < \frac{1}{x} < \frac{1}{x(1-0.578x)}$$

437 *      and
438 *      
$$0 < \frac{1}{x} - \frac{1}{x(1+0.578x)} < \frac{1}{x(1+0.578x)} < 0.578$$

439 *      The error is thus bounded by  $\frac{1}{2} \text{ ulp}(\frac{1}{x}) + 0.578$ . Since  $x \leq 2^{-P}$ ,
440 *      
$$\frac{1}{x} \geq 2^P, \text{ ulp}(\frac{1}{x}) \geq \text{ulp}(2^P) \geq 2.$$
 Thus  $0.578 = 0.289 * 2 \leq 0.289 \text{ ulp}(\frac{1}{x})$ .
441 *      Thus
442 *      
$$|\gamma(x) - [\frac{1}{x}] \text{ rounded}| \leq (0.5 + 0.289) * \text{ulp}(\frac{1}{x}).$$


```

```

455 *      
$$\frac{1}{x} - \frac{1}{2} \text{ rounded to nearest.}$$

456 *      Note that for  $x \leq 2^{-P}$ , it is easy to see that  $\text{ulp}(\frac{1}{x}) = \text{ulp}(\gamma(x))$ .
457 *      
$$\frac{1}{x} - \frac{1}{2} \text{ rounded to nearest.}$$

458 *      except only when  $x = 2^{-n}$ , ( $n \leq -53$ ). In such cases,  $\frac{1}{x}$  is exact
459 *      and therefore the error is bounded by
460 *      
$$0.298 * \text{ulp}(\frac{1}{x}) = 0.298 * \text{ulp}(\gamma(x)) = 0.578 \text{ ulp}(\gamma(x)).$$

461 *      Thus we conclude that the error in  $\gamma$  is less than  $0.739 \text{ ulp}$ .
462 *
463 *      (2) Otherwise, for  $x$  in  $GT_{i-1}$  (see B), let  $y = x - (z_i - 1)$ . From (B) we obtain
464 *      
$$\gamma(1+x) = \gamma.y.h + \gamma.y.l$$

465 *      then compute  $\gamma(x)$  by  $\frac{\gamma.y.h}{x}$ .
466 *      Implementaion note. Write  $x = x.h + x.l$ , and Let  $th = \frac{\gamma.y.h}{x}$  chopped to
467 *      20 bits, then
468 *      
$$\gamma(x) = th + \frac{\gamma.y.h + \gamma.y.l}{x} - th$$

469 *      
$$= th + \frac{1}{x} * (\gamma.y.h - th * x.h + \gamma.y.l - th * x.l)$$

470 *      (D) For  $x$  between 2 and 8. Let  $n = 1 + x$  chopped to an integer. Then
471 *      
$$\gamma(x) = (x-1) * (x-2) * \dots * (x-n) * \gamma(x-n)$$

472 *      Since  $x-n$  is between 1 and 2, we can apply (B) to compute  $\gamma(x-n)$ .
473 *      Implementation detail. The computation of  $(x-1)(x-2)\dots(x-n)$  in simulated
474 *      higher precision arithmetic can be somewhat optimized. For example, in
475 *      computing  $(x-1)(x-2)(x-3)(x-4)$ , if we compute  $(x-1)(x-4) = z.h + z.l$ ,
476 *      then  $(x-2)(x-3) = z.h + 2 + z.l$  readily. In below, we list the expression
477 *      of the formula to compute  $\gamma(x)$ .
478 *
479 *      Assume  $x-n$  is in  $GT_i$  ( $i=1,2$ , or  $3$ , see B for detail). Let  $y = x - n - z_i$ .
480 *      By (B) we have  $\gamma(x-n) = \gamma.y.h + \gamma.y.l$ . If  $x = x.h + x.l$ , then we have
481 *      
$$n=1 \text{ (x in [2,3]):}$$

482 *      
$$\gamma(x) = (x-1) * \gamma(x-1) = (x-1) * (\gamma.y.h + \gamma.y.l)$$

483 *      
$$= [(x.h-1) + x.l] * (\gamma.y.h + \gamma.y.l)$$

484 *      
$$n=2 \text{ (x in [3,4]):}$$

485 *      
$$\gamma(x) = (x-1)(x-2) * \gamma(x-2) = (x-1)(x-2) * (\gamma.y.h + \gamma.y.l)$$

486 *      
$$= ((x.h-2) + x.l) * ((x.h-1) + x.l) * (\gamma.y.h + \gamma.y.l)$$

487 *      
$$= [x.h * (x.h-3) + 2 + x.l * (x + (x.h-3))] * (\gamma.y.h + \gamma.y.l)$$

488 *      
$$n=3 \text{ (x in [4,5]):}$$

489 *      
$$\gamma(x) = (x-1)(x-2)(x-3) * \gamma(x-3)$$

490 *      
$$= (x.h * (x.h-3) + 2 + x.l * (x + (x.h-3))) * [(x.h-3) + x.l] * (\gamma.y.h + \gamma.y.l)$$

491 *      
$$n=4 \text{ (x in [5,6]):}$$

492 *      
$$\gamma(x) = [(x-1)(x-4)] * [(x-2)(x-3)] * (\gamma.y.h + \gamma.y.l)$$

493 *      
$$= [(x.h * (x.h-5) + 4 + x.l * (x + (x.h-5)))] * [(x-2)(x-3)] * (\gamma.y.h + \gamma.y.l)$$

494 *      
$$= (y.h + y.l) * (y.h + 1 + y.l) * (\gamma.y.h + \gamma.y.l)$$

495 *      
$$n=5 \text{ (x in [6,7]):}$$

496 *      
$$\gamma(x) = [(x-1)(x-4)] * [(x-2)(x-3)] * [(x-5) * (\gamma.y.h + \gamma.y.l)]$$

497 *      
$$n=6 \text{ (x in [7,8]):}$$

498 *      
$$\gamma(x) = [(x-1)(x-6)] * [(x-2)(x-5)] * [(x-3)(x-4)] * (\gamma.y.h + \gamma.y.l)$$

499 *      
$$= [(y.h + y.l)(y.h + 4 + y.l)] * [(y.h + 6 + y.l)(\gamma.y.h + \gamma.y.l)]$$

500 *
501 *      (E) Overflow Threshold. For  $x > \text{Overflow threshold of } \gamma$ ,
502 *      return huge * huge (overflow).
503 *
504 *      By checking whether  $\lgamma(x) \geq 2 * \{128, 1024, 16384\}$ , one can
505 *      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 = 1.7555483429044629170038892160702032034177e+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(2*pi)-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) approximate log(G(x))-(x-.5)(log(x)-1)-.5(log(2*pi)-1)
571 * to a precision bounded by 2**(-P-5).
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *

```

$$\begin{array}{r}
 \begin{array}{c}
 \text{L1(x):} \\
 \text{L2:} \\
 \text{+ L3(x):}
 \end{array}
 \begin{array}{c}
 \begin{array}{c}
 \overline{\hspace{10em}} \\
 \text{-----} \\
 \text{-----}
 \end{array} \\
 \begin{array}{c}
 \overline{\hspace{10em}} \\
 \text{-----} \\
 \text{-----}
 \end{array} \\
 \begin{array}{c}
 \overline{\hspace{10em}} \\
 \text{-----} \\
 \text{-----}
 \end{array}
 \end{array}
 \end{array}$$

[leading] + [Trailing]

```

583 * For L1(x)=(x-0.5)*(log(x)-1), we need ilogb(L1(x))+5 extra bits for
584 * both multipliers 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 unbiased
586 * binary exponent of y in IEEE format. We can get x-0.5 to the desired

```

```

587 * accuracy easily. It remains to compute log(x)-1 with ilogb(L1(x))+5
588 * extra bits accuracy. 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 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *

```

	single	double	quadruple
ilogb(L1(x))+5 <=	11	14	18

(3) Table Driven Method for log(x)-1:

Let  $x = 2^{**n} * y$ , where  $1 <= y < 2$ . Let  $Z = \{z(i), i=1, \dots, m\}$  be a set of predetermined evenly distributed floating point numbers in  $[1, 2]$ . Let  $z(j)$  be the closest one to  $y$ , then

$$\begin{aligned}
 \log(x)-1 &= n \log(2)-1 + \log(y) \\
 &= n \log(2)-1 + \log(z(j)*y/z(j)) \\
 &= n \log(2)-1 + \log(z(j)) + \log(y/z(j)) \\
 &= T1(n) + T2(j) + T3,
 \end{aligned}$$

where  $T1(n) = n \log(2)-1$  and  $T2(j) = \log(z(j))$ . Both  $T1$  and  $T2$  can be pre-calculated and be looked-up in a table. Note that  $8 <= x < 1756$  implies  $3 <= n <= 10$  implies  $1.079.. < T1(n) < 6.931$ .

For  $T3$ , let  $s = \frac{y-z(i)}{y+z(i)}$ ; then  $\frac{y}{z(i)} = \frac{1+s}{1-s}$  and

$$T3 = \log\left(\frac{1+s}{1-s}\right) = 2s + \frac{2}{3}s^3 + \frac{2}{5}s^5 + \dots$$

Suppose the first term  $2s$  is computed in extra precision. The dominating error in  $T3$  would then be the rounding error of the second term  $2/3*s**3$ . To force the rounding bounded by the required accuracy, we have

single:	$ 2/3*s**3  < 2**(-11)$	== >	$ s  < 0.09014...$
double:	$ 2/3*s**3  < 2**(-14)$	== >	$ s  < 0.04507...$
quad :	$ 2/3*s**3  < 2**(-18)$	== >	$ s  < 0.01788... = 2**(-5.80..)$

Base on this analysis, we choose  $Z = \{z(i) | z(i) = 1+i/64+1/128, 0 <= i <= 63\}$ . For any  $y$  in  $[1, 2]$ , let  $j = [64*y]$  chopped to integer, then  $z(j)$  is the closest to  $y$ , and it is not difficult to see that  $|s| < 2**(-8)$ . Please note that the polynomial approximation of  $T3$  must be accurate

$-24-11$	$-35$	$-53-14$	$-67$	$-113-18$	$-131$
$= 2$	$= 2$	$= 2$	$= 2$	$= 2$	$= 2$

for single, double, and quadruple precision respectively.

Implementation notes.

(1) Table look-up entries for  $T1(n)$  and  $T2(j)$ , as well as the calculation of the leading term  $2s$  in  $T3$ , are broken up into leading and trailing part such that (leading part)\*  $2**24$  will always be an integer. That will guarantee the addition of the leading parts will be exact.

T1(n):	$\overline{\hspace{10em}}$	$\overline{\hspace{10em}}$
T2(j):	$\overline{\hspace{10em}}$	$\overline{\hspace{10em}}$
2s:	$\overline{\hspace{10em}}$	$\overline{\hspace{10em}}$
+ T3(s)-2s:	$\overline{\hspace{10em}}$	$\overline{\hspace{10em}}$

[leading] + [Trailing]

(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 *
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 *
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 = 0.6666717231848518054693623697539230
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 "tgamma_log"
675 *
676 * The computation of 0.5*(ln(2pi)-1):
677 * 0.5*(ln(2pi)-1) = 0.4189385332046727417803297364056176398614...
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 = 0.4189383983612060546875
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)-(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 *
688 * GP(s) = GP0*s+GP1*s+GP2*s+...+GP19*s
689 * GP0 = 0.0833333333333333333333333333333333333333333172839171301
690 *      hex 0x3ffe5555 55555555 55555555 55555555
691 * GP1 = -2.7777777777777777777777777777777777777777777492501211999399424104e-0003
692 * GP2 = 7.93650793650793650793650793650541638236350020883243e-0004
693 * GP3 = -5.95238095238095238057299772679324503339241961704e-0004
694 * GP4 = 8.41750841750841696138422987977683524926142600321e-0004
695 * GP5 = -1.91752691752686682825032547823699662178842123308e-0003
696 * GP6 = 6.41025641022403480921891559356473451161279359322e-0003
697 * GP7 = -2.95506535798414019189819587455577003732808185071e-0002
698 * GP8 = 1.79644367229970031486079180060923073476568732136e-0001
699 * GP9 = -1.3924308648727466217456287256705720025564929064e+0000
700 * GP10 = 1.34025874044417962188677816477842265259608269775e+0001
701 * GP11 = -1.56803713480127469414495545399982508700748274318e+0002
702 * GP12 = 2.18739841656201561694927630335099313968924493891e+0003
703 * GP13 = -3.55249848644100338419187038090925410976237921269e+0004
704 * GP14 = 6.43464880437835286216768959439484376449179576452e+0005
705 * GP15 = -1.20459154385577014992600342782821389605893904624e+0007
706 * GP16 = 2.09263249637351298563934942349749718491071093210e+0008
707 * GP17 = -2.96247483183169219343745316433899599834685703457e+0009
708 * GP18 = 2.88984933605896033154727626086506756972327292981e+0010
709 * GP19 = -1.40960434146030007732838382416230610302678063984e+0011
710 *
711 * double precision
712 * |GP(s) - s*P(s^2)| <= 2**(-63.5), where
713 *
714 * GP(s) = GP0*s+GP1*s+GP2*s+GP3*s+GP4*s+GP5*s+GP6*s+GP7*s
715 *
716 * GP0= 0.0833333333333333287074040640618477 (3FB55555 55555555)
717 * GP1= -2.77777777776649355200565611114627670089130772843e-0003
718 * GP2= 7.93650787486083724805476194170211775784158551509e-0004
```

```
719 * GP3= -5.95236628558314928757811419580281294593903582971e-0004
720 * GP4= 8.41566473999853451983137162780427812781178932540e-0004
721 * GP5= -1.90424776670441373564512942038926168175921303212e-0003
722 * GP6= 5.84933161530949666312333949534482303007354299178e-0003
723 * GP7= -1.59453228931082030262124832506144392496561694550e-0002
724 *
725 * single precision
726 * |GP(s) - s*P(s^2)| <= 2**(-37.78), where
727 *          3      5
728 * GP(s) = GP0*s+GP1*s+GP2*s
729 * GP0 = 8.33333330959694065245736888749042811909994573178e-0002
730 * GP1 = -2.77765545601667179767706600890361535225507762168e-0003
731 * GP2 = 7.77830853479775281781085278324621033523037489883e-0004
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)*expml(r)), where
745 * expml(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 };
760 #endif
```

new/usr/src/lib/libm/common/m9x/tgamma.c

1

\*\*\*\*\*

40060 Mon Oct 27 12:37:23 2014

new/usr/src/lib/libm/common/m9x/tgamma.c

5262 libm needs to be carefully undef'd

5268 libm doesn't need to hide symbols which are already local

\*\*\*\*\*

```
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 /*
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 */
29
30 #if defined(ELFOBJ)
30 #pragma weak tgamma = __tgamma
32 #endif
33
32 #include "libm.h"
33 #include <sys/isa_defs.h>
34
35 #if defined(_BIG_ENDIAN)
36 #define H0_WORD(x) ((unsigned *) &x)[0]
37 #define H3_WORD(x) ((unsigned *) &x)[3]
38 #define CHOPPED(x) (long double) ((double) (x))
39 #else
40 #define H0_WORD(x) (((int *) &x)[2] << 16) | \
41 (0x0000ffff & (((unsigned *) &x)[1] >> 15)))
42 #define H3_WORD(x) ((unsigned *) &x)[0]
43 #define CHOPPED(x) (long double) ((float) (x))
44 #endif
45
46 struct LDouble {
47     long double h, l;
48 };
49
50 _____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/trunc.c

1

```
*****
1772 Mon Oct 27 12:37:23 2014
new/usr/src/lib/libm/common/m9x/trunc.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak trunc = __trunc
32 #endif

32 #include "libm.h"

34 double
35 trunc(double x) {
36     union {
37         unsigned i[2];
38         double d;
39     } xx;
40     unsigned hx, sx, i;

42     xx.d = x;
43     hx = xx.i[HIWORD] & ~0x80000000;
44     sx = xx.i[HIWORD] & 0x80000000;
45     if (hx < 0x43300000) { /* |x| < 2^52 */
46         if (hx < 0x3ff00000) /* |x| < 1 */
47             return (sx ? -0.0 : 0.0);

49         /* chop x at the integer bit */
50         if (hx < 0x41300000) {
51             i = 1 << (0x412 - (hx >> 20));
52             xx.i[HIWORD] &= ~(i | (i - 1));
53             xx.i[LOWORD] = 0;
54         } else {
55             i = 1 << (0x432 - (hx >> 20));
56             xx.i[LOWORD] &= ~(i | (i - 1));
57         }
58         return (xx.d);

```

new/usr/src/lib/libm/common/m9x/trunc.c

2

```
59     } else if (hx < 0x7ff00000)
60         return (x);
61     else
62 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
63         return (hx >= 0x7ff80000 ? x : x + x);
64         /* assumes sparc-like QNaN */
65 #else
66         return (x + x);
67 #endif
68 }
unchanged_portion_omitted

```

new/usr/src/lib/libm/common/m9x/truncf.c

1

```
*****
1610 Mon Oct 27 12:37:24 2014
new/usr/src/lib/libm/common/m9x/truncf.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak truncf = __truncf
32 #endif

32 #include "libm.h"

34 float
35 truncf(float x) {
36     union {
37         unsigned i;
38         float f;
39     } xx;
40     unsigned hx, sx, i;

42     xx.f = x;
43     hx = xx.i & ~0x80000000;
44     sx = xx.i & 0x80000000;
45     if (hx < 0x4b000000) { /* |x| < 2^23 */
46         if (hx < 0x3f800000) /* |x| < 1 */
47             return (sx ? -0.0F : 0.0F);

49         /* chop x at the integer bit */
50         i = 1 << (0x95 - (hx >> 23));
51         xx.i &= ~(i << 1) - 1;
52         return (xx.f);
53     } else if (hx < 0x7f800000) /* |x| is integral */
54         return (x);
55     else
56 #if defined(FPADD_TRAPS_INCOMPLETE_ON_NAN)
57         return (hx > 0x7f800000 ? x * x : x + x);
58 #else
```

new/usr/src/lib/libm/common/m9x/truncf.c

2

```
59         return (x + x);
60 #endif
61 }
_____unchanged_portion_omitted_____
```

new/usr/src/lib/libm/common/m9x/truncl.c

1

```
*****
2668 Mon Oct 27 12:37:24 2014
new/usr/src/lib/libm/common/m9x/truncl.c
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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 #if defined(ELFOBJ)
30 #pragma weak truncl = __truncl
32 #endif

32 #include "libm.h"

34 #if defined(__sparc)
35 long double
36 truncl(long double x) {
37     union {
38         unsigned i[4];
39         long double q;
40     } xx;
41     unsigned hx, sx;
42     int j;

44     xx.q = x;
45     sx = xx.i[0] & 0x80000000;
46     hx = xx.i[0] & ~0x80000000;

48     /* handle trivial cases */
49     if (hx >= 0x406f0000) /* |x| >= 2^112 + ... or x is nan */
50         return (hx >= 0x7fff0000 ? x + x : x);

52     /* handle |x| < 1 */
53     if (hx < 0x3fff0000)
54         return (sx ? -0.0L : 0.0L);

56     j = 0x406f - (hx >> 16);          /* 1 <= j <= 112 */
57     xx.i[0] = hx;
58     if (j >= 96) {                    /* 96 <= j <= 112 */
```

new/usr/src/lib/libm/common/m9x/truncl.c

2

```
59         xx.i[0] &= ~((1 << (j - 96)) - 1);
60         xx.i[1] = xx.i[2] = xx.i[3] = 0;
61     } else if (j >= 64) {              /* 64 <= j <= 95 */
62         xx.i[1] &= ~((1 << (j - 64)) - 1);
63         xx.i[2] = xx.i[3] = 0;
64     } else if (j >= 32) {              /* 32 <= j <= 63 */
65         xx.i[2] &= ~((1 << (j - 32)) - 1);
66         xx.i[3] = 0;
67     } else                              /* 1 <= j <= 31 */
68         xx.i[3] &= ~((1 << j) - 1);

70     /* negate result if need be */
71     if (sx)
72         xx.i[0] |= 0x80000000;
73     return (xx.q);
74 }
_____unchanged_portion_omitted_____
```

```

*****
2572 Mon Oct 27 12:37:25 2014
new/usr/src/lib/libm/sparc/src/nextafter.S
5262 libm needs to be carefully undef'd
5268 libm doesn't need to hide symbols which are already local
*****
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      "nextafter.S"

31 #include "libm.h"
32 LIBM_ANSI_PRAGMA_WEAK(nextafter,function)
33     .weak _nextafter
34     .type _nextafter,#function
35 _nextafter = __nextafter
36 #include "libm_synonyms.h"
37 #include "libm_protos.h"

39 #if defined(LIBM_BUILD) && !defined(ELFOBJ)
40 #define mENTRY(x)      ENTRY(__libm/**/x)
41 #define mName(x)      NAME(__libm/**/x)
42 #else
43 #define mENTRY(x)      ENTRY(x)
44 #define mName(x)      NAME(x)
45 #endif

39      RO_DATA
40      .align      8
41 .Lconstant:
42 two54      = 0x00
43     .word      0x43500000,0x0      ! 2**54
44 twom54     = 0x08
45     .word      0x3c900000,0x0      ! 2**-54
46 tiny      = 0x10
47     .word      0x00100000,0x0      ! tiny

49 ! variable using fp
50 x      = -0x8
51 y      = -0x10

```

```

53      ENTRY(nextafter)
54      save      %sp,-128,%sp
55      PIC_SETUP(17)
56      std      %i0,[%fp+x]
57      or      %g0,%i0,%o0      ! save original arguments
58      or      %g0,%i1,%o1
59      std      %i2,[%fp+y]
60      or      %g0,%i2,%o2
61      or      %g0,%i3,%o3
62      ldd      [%fp+x],%f2      ! x
63      ldd      [%fp+y],%f0      ! y
64      fcmpd   %f2,%f0      ! x:y
65      PIC_SET(17,.Lconstant,10)
66      sethi   %hi(0x80000000),%l1
67      andn   %i0,%l1,%l4
68      fbe    9f      ! next_return
69      nop
70      fbu,a    9f      ! next_return
71      fmuld   %f2,%f0,%f0      ! + -> * for Cheetah
72      orcc   %i1,%l4,%g0      ! see if x is zero
73      bne    1f
74      tst     %i0
75      ! x is zero, return sign(y)*min
76      and    %i2,%l1,%i0
77      ba     4f      ! next_final
78      mov    1,%i1
79 1:      bge    2f
80      nop
81      ! x is negative
82      fbl    1f      ! next_subulp
83      nop
84      fbg    3f      ! next_addulp
85      nop
86 2:
87      fbl    3f      ! next_addulp
88      nop
89 1:
90      subcc   %i1,1,%i1      ! next_subulp
91      ba     4f      ! next_final
92      subx   %i0,0,%i0
93 3:
94      addcc   %i1,1,%i1      ! next_addulp
95      addx   %i0,0,%i0
96 4:
97      sethi   %hi(0x7ff00000),%l3
98      std     %i0,[%fp+x]
99      andcc   %i0,%l3,%i2
100     be,a    1f      ! xflow
101     ldd     [%l0+tiny],%f2
102     cmp     %i2,%l3
103     bne,a   9f      ! next_return
104     ldd     [%fp+x],%f0
105     call    NAME(_SVID_libm_err) ! overflow
106     call    mName(_SVID_libm_err) ! overflow
107     or     %g0,46,%o4
108     ba     9f
109 1:      nop
110     fmuld   %f2,%f2,%f2      ! xflow
111     ldd     [%fp+x],%f0
112 9:      ! next_return
113     ret
114     restore

116     SET_SIZE(nextafter)
      unchanged_portion_omitted

```