

new/usr/src/uts/Makefile.uts

1

```
*****
21739 Mon Apr 15 12:20:21 2019
new/usr/src/uts/Makefile.uts
10805 Fix for 10687 can be improved
*****
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 (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
25 # Copyright (c) 2011 by Delphix. All rights reserved.
26 # Copyright (c) 2013 Andrew Stormont. All rights reserved.
27 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
28 # Copyright (c) 2019, Joyent, Inc.
29 #
31 #
32 # This Makefile contains the common targets and definitions for
33 # all kernels. It is to be included in the Makefiles for specific
34 # implementation architectures and processor architecture dependent
35 # modules: i.e.: all driving kernel Makefiles.
36 #
37 # Include global definitions:
38 #
39 include $(SRC)/Makefile.master
41 #
42 # No text domain in the kernel.
43 #
44 DTEXTDOM =
46 #
47 # Keep references to $(SRC)/common relative.
48 COMMONBASE= $(UTSBASE)/../common
50 #
51 # Setup build-specific vars
52 # To add a build type:
53 # add name to ALL_BUILDS32 & ALL_BUILDS64
54 # set CLASS_name and OBJ_DIR_name
55 # add targets to Makefile.targ
56 #
58 #
59 # DEF_BUILDS is for def, lint, sischeck, and install
60 # ALL_BUILDS is for everything else (all, clean, ...)
61 #
```

new/usr/src/uts/Makefile.uts

2

```
62 # The NOT_RELEASE_BUILD noise is to maintain compatibility with the
63 # gatekeeper's nightly build script.
64 #
65 DEF_BUILDS32 = obj32
66 DEF_BUILDS64 = obj64
67 DEF_BUILDSONLY64 = obj64
68 $(NOT_RELEASE_BUILD)DEF_BUILDS32 = debug32
69 $(NOT_RELEASE_BUILD)DEF_BUILDS64 = debug64
70 $(NOT_RELEASE_BUILD)DEF_BUILDSONLY64 = debug64
71 ALL_BUILDS32 = obj32 debug32
72 ALL_BUILDS64 = obj64 debug64
73 ALL_BUILDSONLY64 = obj64 debug64
75 #
76 # For modules in 64b dirs that aren't built 64b
77 # or modules in 64b dirs that aren't built 32b we
78 # need to create empty modlintlib files so global lint works
79 #
80 LINT32_BUILDS = debug32
81 LINT64_BUILDS = debug64
83 #
84 # Build class (32b or 64b)
85 #
86 CLASS_OBJ32 = 32
87 CLASS_DBG32 = 32
88 CLASS_OBJ64 = 64
89 CLASS_DBG64 = 64
90 CLASS = $(CLASS_$(BUILD_TYPE))
92 #
93 # Build subdirectory
94 #
95 OBJ32_DIR_OBJ32 = obj32
96 OBJ32_DIR_DBG32 = debug32
97 OBJ64_DIR_OBJ64 = obj64
98 OBJ64_DIR_DBG64 = debug64
99 OBJ32_DIR = $(OBJ32_DIR_$(BUILD_TYPE))
101 #
102 # Create defaults so empty rules don't
103 # confuse make
104 #
105 CLASS_ = 64
106 OBJ64_DIR_ = debug64
108 #
109 # Build tools
110 #
111 CC_sparc_32 = $(sparc_CC)
112 CC_sparc_64 = $(sparcv9_CC)
114 CC_i386_32 = $(i386_CC)
115 CC_i386_64 = $(amd64_CC)
116 CC_amd64_64 = $(amd64_CC)
118 CC = $(CC_$(MACH)_$(CLASS))
120 AS_sparc_32 = $(sparc_AS)
121 AS_sparc_64 = $(sparcv9_AS)
123 AS_i386_32 = $(i386_AS)
124 AS_i386_64 = $(amd64_AS)
125 AS_amd64_64 = $(amd64_AS)
127 AS = $(AS_$(MACH)_$(CLASS))
```

```

129 LD_sparc_32      = $(sparc_LD)
130 LD_sparc_64      = $(sparcv9_LD)

132 LD_i386_32      = $(i386_LD)
133 LD_i386_64      = $(amd64_LD)
134 LD_amd64_64     = $(amd64_LD)

136 LD              = $(LD_$(MACH)_$(CLASS))

138 LINT_sparc_32    = $(sparc_LINT)
139 LINT_sparc_64    = $(sparcv9_LINT)

141 LINT_i386_32     = $(i386_LINT)
142 LINT_i386_64     = $(amd64_LINT)
143 LINT_amd64_64    = $(amd64_LINT)

145 LINT             = $(LINT_$(MACH)_$(CLASS))

147 MODEL_32        = ilp32
148 MODEL_64        = lp64
149 MODEL            = $(MODEL_$(CLASS))

151 #
152 #           Build rules for linting the kernel.
153 #
154 LHEAD = $(ECHO) "\n$@";

156 # Note: egrep returns "failure" if there are no matches, which is
157 # exactly the opposite of what we need.
158 LGREP.2 =          if egrep -v ' (_init|_fini|_info) ' ; then false; else true; fi

160 LTAIL =

162 LINT.c =          $(LINT) -c -dirout=$(LINTS_DIR) $(LINTFLAGS) $(LINT_DEFS) $(CPPF

164 # Please do not add new erroff directives here.  If you need to disable
165 # lint warnings in your module for things that cannot be fixed in any
166 # reasonable manner, please augment LINTTAGS in your module Makefile
167 # instead.
168 LINTTAGS          = -erroff=E_INCONS_ARG_DECL2
169 LINTTAGS          += -erroff=E_INCONS_VAL_TYPE_DECL2

171 LINTFLAGS_sparc_32 = $(LINTCCMODE) -nsxmuF -errtags=yes
172 LINTFLAGS_sparc_64 = $(LINTFLAGS_sparc_32) -m64
173 LINTFLAGS_i386_32  = $(LINTCCMODE) -nsxmuF -errtags=yes
174 LINTFLAGS_i386_64  = $(LINTFLAGS_i386_32) -m64

176 LINTFLAGS        = $(LINTFLAGS_$(MACH)_$(CLASS)) $(LINTTAGS)
177 LINTFLAGS        += $(C99LMODE)

179 #
180 #           Override this variable to modify the name of the lint target.
181 #
182 LINT_MODULE=      $(MODULE)

184 #
185 #           Build the compile/assemble lines:
186 #
187 EXTRA_OPTIONS     =
188 AS_DEFS           = -D_ASM -D__STDC__=0

190 ALWAYS_DEFS_32    = -D_KERNEL -D_SYSCALL32 -D_DDI_STRICT
191 ALWAYS_DEFS_64    = -D_KERNEL -D_SYSCALL32 -D_SYSCALL32_IMPL -D_ELF64 \
192                  -D_DDI_STRICT
193 #

```

```

194 # XX64 This should be defined by the compiler!
195 #
196 ALWAYS_DEFS_64    += -Dsun -D__sun -D__SVR4
197 ALWAYS_DEFS      = $(ALWAYS_DEFS_$(CLASS))

199 #
200 #           CPPFLAGS is deliberately set with a "=" and not a "+=".  For the kernel
201 #           the header include path should not look for header files outside of
202 #           the kernel code.  This "=" removes the search path built in
203 #           Makefile.master inside CPPFLAGS.  Ditto for AS_CPPFLAGS.
204 #
205 CPPFLAGS          = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) \
206                  $(INCLUDE_PATH) $(EXTRA_OPTIONS)
207 ASFLAGS           += -P
208 AS_CPPFLAGS       = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) $(AS_DEFS) \
209                  $(AS_INC_PATH) $(EXTRA_OPTIONS)

211 #
212 #           Make it (relatively) easy to share compilation options between
213 #           all kernel implementations.
214 #

216 # Override the default, the kernel is squeaky clean
217 CERRWARN = -errtags=yes -errwarn=%all

219 CERRWARN += -_gcc=-Wno-missing-braces
220 CERRWARN += -_gcc=-Wno-sign-compare
221 CERRWARN += -_gcc=-Wno-unknown-pragmas
222 CERRWARN += -_gcc=-Wno-unused-parameter
223 CERRWARN += -_gcc=-Wno-missing-field-initializers

225 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
226 # -nd builds
227 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
228 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

230 CERRWARN += -_smatch=-p=illumos_kernel
231 include $(SRC)/Makefile.smatch

233 #
234 # Unfortunately, _IOWR() is regularly used with a third argument of 0,
235 # so we have to disable all these smatch checks.
236 #
237 SMOFF += sizeof

239 #
240 # DEBUG-only macros that define away to nothing confuse this check,
241 # unfortunately.
242 #
243 $(RELEASE_BUILD)SMOFF += indenting

239 CSTD = $(CSTD_GNU99)

241 CFLAGS_uts        =
242 CFLAGS_uts        += $(STAND_FLAGS_$(CLASS))
243 CFLAGS_uts        += $(CCVERBOSE)
244 CFLAGS_uts        += $(ILD OFF)
245 CFLAGS_uts        += $(XAOPT)
246 CFLAGS_uts        += $(CTF_FLAGS_$(CLASS))
247 CFLAGS_uts        += $(CERRWARN)
248 CFLAGS_uts        += $(CCNOAUTOINLINE)
249 CFLAGS_uts        += $(CCNOREORDER)
250 CFLAGS_uts        += $(CCNOAGGRESSIVELOOPS)
251 CFLAGS_uts        += $(CGLOBALSTATIC)
252 CFLAGS_uts        += $(EXTRA_CFLAGS)
253 CFLAGS_uts        += $(CSOURCEDEBUGFLAGS)

```

```

254 CFLAGS_uts          += $(CUSERFLAGS)

256 #
257 #   Declare that $(OBJECTS) and $(LINTS) can be compiled in parallel.
258 #   The DUMMY target is for those instances where OBJECTS and LINTS
259 #   are empty (to avoid an unconditional .PARALLEL).
260 .PARALLEL:          $(OBJECTS) $(LINTS) DUMMY

262 #
263 #   Expanded dependencies
264 #
265 DEF_DEPS             = $(DEF_BUILDS:%=def.%)
266 ALL_DEPS             = $(ALL_BUILDS:%=all.%)
267 CLEAN_DEPS          = $(ALL_BUILDS:%=clean.%)
268 CLOBBER_DEPS        = $(ALL_BUILDS:%=clobber.%)
269 LINT_DEPS           = $(DEF_BUILDS:%=lint.%)
270 MODLINTLIB_DEPS     = $(DEF_BUILDS:%=modlintlib.%)
271 MODLIST_DEPS        = $(DEF_BUILDS:%=modlist.%)
272 CLEAN_LINT_DEPS     = $(ALL_BUILDS:%=clean.lint.%)
273 INSTALL_DEPS        = $(DEF_BUILDS:%=install.%)
274 SYM_DEPS            = $(SYM_BUILDS:%=symcheck.%)
275 SISCHECK_DEPS       = $(DEF_BUILDS:%=sischeck.%)
276 SISCLEAN_DEPS       = $(ALL_BUILDS:%=sisclean.%)

278 #
279 #   Default module name
280 #
281 BINARY               = $(OBJS_DIR)/$(MODULE)

283 #
284 #   Default cleanup definitions
285 #
286 CLEANLINTFILES      = $(LINTS) $(MOD_LINT_LIB)
287 CLEANFILES           = $(OBJECTS) $(CLEANLINTFILES)
288 CLOBBERFILES         = $(BINARY) $(CLEANFILES)

290 #
291 #   Installation constants:
292 #
293 #   FILEMODE is the mode given to the kernel modules
294 #   CFILEMODE is the mode given to the '.conf' files
295 #
296 FILEMODE             = 755
297 DIRMODE              = 755
298 CFILEMODE            = 644

300 #
301 #   Special Installation Macros for the installation of '.conf' files.
302 #
303 #   These are unique because they are not installed from the current
304 #   working directory.
305 #
306 #   Sigh. Apparently at some time in the past there was a confusion on
307 #   whether the name is SRC_CONFFILE or SRC_CONFFILE. Consistency with the
308 #   other names would indicate SRC_CONFFILE, but the voting is >180 Makefiles
309 #   with SRC_CONFFILE and about 11 with SRC_CONFFILE. Software development
310 #   isn't a popularity contest, though, and so my inclination is to define
311 #   both names for now and incrementally convert to SRC_CONFFILE to be consistent
312 #   with the other names.
313 #
314 CONFFILE              = $(MODULE).conf
315 SRC_CONFFILE          = $(CONF_SRCDIR)/$(CONFFILE)
316 SRC_CONFFILE          = $(SRC_CONFFILE)
317 ROOT_CONFFILE_32      = $(ROOTMODULE).conf
318 ROOT_CONFFILE_64      = $(ROOTMODULE:%/$(SUBDIR64)/$(MODULE)=%/$(MODULE)).conf
319 ROOT_CONFFILE         = $(ROOT_CONFFILE_$(CLASS))

```

```

322 INS.conf= \
323     $(RM) $@; $(INS) -s -m $(CFILEMODE) -f $(@D) $(SRC_CONFFILE)

325 #
326 #   The CTF merge of child kernel modules is performed against one of the genunix
327 #   modules. For Intel builds, all modules will be used with a single genunix:
328 #   the one built in intel/genunix. For SPARC builds, a given
329 #   module may be
330 #   used with one of a number of genunix files, depending on what platform the
331 #   module is deployed on. We merge against the sun4u genunix to optimize for
332 #   the common case. We also merge against the ip driver since networking is
333 #   typically loaded and types defined therein are shared between many modules.
334 #
335 CTFMERGE_GUDIR_sparc = sun4u
336 CTFMERGE_GUDIR_i386 = intel
337 CTFMERGE_GUDIR       = $(CTFMERGE_GUDIR_$(MACH))

339 CTFMERGE_GENUNIX     = \
340     $(UTSBASE)/$(CTFMERGE_GUDIR)/genunix/$(OBJS_DIR)/genunix

342 #
343 #   Used to uniquify a non-genunix module against genunix. $VERSION is used
344 #   for the label.
345 #
346 #   For the ease of developers dropping modules onto possibly unrelated systems,
347 #   you can set NO_GENUNIX_UNIQUIFY= in the environment to skip uniquifying
348 #   against genunix.
349 #
350 NO_GENUNIX_UNIQUIFY=$(POUND_SIGN)
351 CTFMERGE_GENUNIX_DFLAG=-d $(CTFMERGE_GENUNIX)
352 $(NO_GENUNIX_UNIQUIFY)CTF_GENUNIX_DFLAG=

354 CTFMERGE_UNIQUIFY_AGAINST_GENUNIX = \
355     $(CTFMERGE) $(CTFMRGFLAGS) -L VERSION \
356     $(CTFMERGE_GENUNIX_DFLAG) -o $@ $(OBJECTS) $(CTFEXTRAOBJS)

358 #
359 #   Used to merge the genunix module.
360 #
361 CTFMERGE_GENUNIX_MERGE = \
362     $(CTFMERGE) $(CTFMRGFLAGS) -L VERSION -o $@ \
363     $(OBJECTS) $(CTFEXTRAOBJS) $(IPCTF_TARGET)

365 #
366 #   We ctfmerge the ip objects into genunix to maximize the number of common types
367 #   found there, thus maximizing the effectiveness of uniquification. We don't
368 #   want the genunix build to have to know about the individual ip objects, so we
369 #   put them in an archive. The genunix ctfmerge then includes this archive.
370 #
371 IPCTF                    = $(IPDRV_DIR)/$(OBJS_DIR)/ipctf.a

373 #
374 #   Rule for building fake shared libraries used for symbol resolution
375 #   when building other modules. -znoreloc is needed here to avoid
376 #   tripping over code that isn't really suitable for shared libraries.
377 #
378 BUILD.SO                 = \
379     $(LD) -o $@ $(GSHARED) $(ZNORELOC) -h $(SONAME)

381 #
382 #   SONAME defaults for common fake shared libraries.
383 #
384 $(LIBGEN)                := SONAME = $(MODULE)
385 $(PLATLIB)               := SONAME = misc/platmod

```

```

386 $(CPULIB) := SONAME = 'cpu/$$CPU'
387 $(DTRACESTUBS) := SONAME = dtracestubs

389 #
390 # Installation directories
391 #

393 #
394 # For now, 64b modules install into a subdirectory
395 # of their 32b brethren.
396 #
397 SUBDIR64_sparc = sparcv9
398 SUBDIR64_i386 = amd64
399 SUBDIR64 = $(SUBDIR64_$(MACH))

401 ROOT_MOD_DIR = $(ROOT)/kernel

403 ROOT_KERN_DIR_32 = $(ROOT_MOD_DIR)
404 ROOT_BRAND_DIR_32 = $(ROOT_MOD_DIR)/brand
405 ROOT_DRV_DIR_32 = $(ROOT_MOD_DIR)/drv
406 ROOT_DTRACE_DIR_32 = $(ROOT_MOD_DIR)/dtrace
407 ROOT_EXEC_DIR_32 = $(ROOT_MOD_DIR)/exec
408 ROOT_FS_DIR_32 = $(ROOT_MOD_DIR)/fs
409 ROOT_SCHED_DIR_32 = $(ROOT_MOD_DIR)/sched
410 ROOT_SOCKET_DIR_32 = $(ROOT_MOD_DIR)/socketmod
411 ROOT_STRMOD_DIR_32 = $(ROOT_MOD_DIR)/strmod
412 ROOT_IPP_DIR_32 = $(ROOT_MOD_DIR)/ipp
413 ROOT_SYS_DIR_32 = $(ROOT_MOD_DIR)/sys
414 ROOT_MISC_DIR_32 = $(ROOT_MOD_DIR)/misc
415 ROOT_KGSS_DIR_32 = $(ROOT_MOD_DIR)/misc/kgss
416 ROOT_SCSI_VHCI_DIR_32 = $(ROOT_MOD_DIR)/misc/scsi_vhci
417 ROOT_PMCS_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/pmcs
418 ROOT_QLC_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/qlc
419 ROOT_EMLXS_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/emlxs
420 ROOT_NLMISC_DIR_32 = $(ROOT_MOD_DIR)/misc
421 ROOT_MACH_DIR_32 = $(ROOT_MOD_DIR)/mach
422 ROOT_CPU_DIR_32 = $(ROOT_MOD_DIR)/cpu
423 ROOT_TOD_DIR_32 = $(ROOT_MOD_DIR)/tod
424 ROOT_FONT_DIR_32 = $(ROOT_MOD_DIR)/fonts
425 ROOT_DACF_DIR_32 = $(ROOT_MOD_DIR)/dacf
426 ROOT_CRYPTODIR_32 = $(ROOT_MOD_DIR)/crypto
427 ROOT_MAC_DIR_32 = $(ROOT_MOD_DIR)/mac
428 ROOT_KICONV_DIR_32 = $(ROOT_MOD_DIR)/kiconv

430 ROOT_KERN_DIR_64 = $(ROOT_MOD_DIR)/$(SUBDIR64)
431 ROOT_BRAND_DIR_64 = $(ROOT_MOD_DIR)/brand/$(SUBDIR64)
432 ROOT_DRV_DIR_64 = $(ROOT_MOD_DIR)/drv/$(SUBDIR64)
433 ROOT_DTRACE_DIR_64 = $(ROOT_MOD_DIR)/dtrace/$(SUBDIR64)
434 ROOT_EXEC_DIR_64 = $(ROOT_MOD_DIR)/exec/$(SUBDIR64)
435 ROOT_FS_DIR_64 = $(ROOT_MOD_DIR)/fs/$(SUBDIR64)
436 ROOT_SCHED_DIR_64 = $(ROOT_MOD_DIR)/sched/$(SUBDIR64)
437 ROOT_SOCKET_DIR_64 = $(ROOT_MOD_DIR)/socketmod/$(SUBDIR64)
438 ROOT_STRMOD_DIR_64 = $(ROOT_MOD_DIR)/strmod/$(SUBDIR64)
439 ROOT_IPP_DIR_64 = $(ROOT_MOD_DIR)/ipp/$(SUBDIR64)
440 ROOT_SYS_DIR_64 = $(ROOT_MOD_DIR)/sys/$(SUBDIR64)
441 ROOT_MISC_DIR_64 = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
442 ROOT_KGSS_DIR_64 = $(ROOT_MOD_DIR)/misc/kgss/$(SUBDIR64)
443 ROOT_SCSI_VHCI_DIR_64 = $(ROOT_MOD_DIR)/misc/scsi_vhci/$(SUBDIR64)
444 ROOT_PMCS_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/pmcs/$(SUBDIR64)
445 ROOT_QLC_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/qlc/$(SUBDIR64)
446 ROOT_EMLXS_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/emlxs/$(SUBDIR64)
447 ROOT_NLMISC_DIR_64 = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
448 ROOT_MACH_DIR_64 = $(ROOT_MOD_DIR)/mach/$(SUBDIR64)
449 ROOT_CPU_DIR_64 = $(ROOT_MOD_DIR)/cpu/$(SUBDIR64)
450 ROOT_TOD_DIR_64 = $(ROOT_MOD_DIR)/tod/$(SUBDIR64)
451 ROOT_FONT_DIR_64 = $(ROOT_MOD_DIR)/fonts/$(SUBDIR64)

```

```

452 ROOT_DACF_DIR_64 = $(ROOT_MOD_DIR)/dacf/$(SUBDIR64)
453 ROOT_CRYPTODIR_64 = $(ROOT_MOD_DIR)/crypto/$(SUBDIR64)
454 ROOT_MAC_DIR_64 = $(ROOT_MOD_DIR)/mac/$(SUBDIR64)
455 ROOT_KICONV_DIR_64 = $(ROOT_MOD_DIR)/kiconv/$(SUBDIR64)

457 ROOT_KERN_DIR = $(ROOT_KERN_DIR_$(CLASS))
458 ROOT_BRAND_DIR = $(ROOT_BRAND_DIR_$(CLASS))
459 ROOT_DRV_DIR = $(ROOT_DRV_DIR_$(CLASS))
460 ROOT_DTRACE_DIR = $(ROOT_DTRACE_DIR_$(CLASS))
461 ROOT_EXEC_DIR = $(ROOT_EXEC_DIR_$(CLASS))
462 ROOT_FS_DIR = $(ROOT_FS_DIR_$(CLASS))
463 ROOT_SCHED_DIR = $(ROOT_SCHED_DIR_$(CLASS))
464 ROOT_SOCKET_DIR = $(ROOT_SOCKET_DIR_$(CLASS))
465 ROOT_STRMOD_DIR = $(ROOT_STRMOD_DIR_$(CLASS))
466 ROOT_IPP_DIR = $(ROOT_IPP_DIR_$(CLASS))
467 ROOT_SYS_DIR = $(ROOT_SYS_DIR_$(CLASS))
468 ROOT_MISC_DIR = $(ROOT_MISC_DIR_$(CLASS))
469 ROOT_KGSS_DIR = $(ROOT_KGSS_DIR_$(CLASS))
470 ROOT_SCSI_VHCI_DIR = $(ROOT_SCSI_VHCI_DIR_$(CLASS))
471 ROOT_PMCS_FW_DIR = $(ROOT_PMCS_FW_DIR_$(CLASS))
472 ROOT_QLC_FW_DIR = $(ROOT_QLC_FW_DIR_$(CLASS))
473 ROOT_EMLXS_FW_DIR = $(ROOT_EMLXS_FW_DIR_$(CLASS))
474 ROOT_NLMISC_DIR = $(ROOT_NLMISC_DIR_$(CLASS))
475 ROOT_MACH_DIR = $(ROOT_MACH_DIR_$(CLASS))
476 ROOT_CPU_DIR = $(ROOT_CPU_DIR_$(CLASS))
477 ROOT_TOD_DIR = $(ROOT_TOD_DIR_$(CLASS))
478 ROOT_FONT_DIR = $(ROOT_FONT_DIR_$(CLASS))
479 ROOT_DACF_DIR = $(ROOT_DACF_DIR_$(CLASS))
480 ROOT_CRYPTODIR = $(ROOT_CRYPTODIR_$(CLASS))
481 ROOT_MAC_DIR = $(ROOT_MAC_DIR_$(CLASS))
482 ROOT_KICONV_DIR = $(ROOT_KICONV_DIR_$(CLASS))
483 ROOT_FIRMWARE_DIR = $(ROOT_MOD_DIR)/firmware

485 ROOT_MOD_DIRS_32 = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
486 ROOT_MOD_DIRS_32 = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
487 ROOT_MOD_DIRS_32 += $(ROOT_EXEC_DIR_32) $(ROOT_DTRACE_DIR_32)
488 ROOT_MOD_DIRS_32 += $(ROOT_FS_DIR_32) $(ROOT_SCHED_DIR_32)
489 ROOT_MOD_DIRS_32 += $(ROOT_STRMOD_DIR_32) $(ROOT_SYS_DIR_32)
490 ROOT_MOD_DIRS_32 += $(ROOT_IPP_DIR_32) $(ROOT_SOCKET_DIR_32)
491 ROOT_MOD_DIRS_32 += $(ROOT_MISC_DIR_32) $(ROOT_MACH_DIR_32)
492 ROOT_MOD_DIRS_32 += $(ROOT_KGSS_DIR_32)
493 ROOT_MOD_DIRS_32 += $(ROOT_SCSI_VHCI_DIR_32)
494 ROOT_MOD_DIRS_32 += $(ROOT_PMCS_FW_DIR_32)
495 ROOT_MOD_DIRS_32 += $(ROOT_QLC_FW_DIR_32)
496 ROOT_MOD_DIRS_32 += $(ROOT_EMLXS_FW_DIR_32)
497 ROOT_MOD_DIRS_32 += $(ROOT_CPU_DIR_32) $(ROOT_FONT_DIR_32)
498 ROOT_MOD_DIRS_32 += $(ROOT_TOD_DIR_32) $(ROOT_DACF_DIR_32)
499 ROOT_MOD_DIRS_32 += $(ROOT_CRYPTODIR_32) $(ROOT_MAC_DIR_32)
500 ROOT_MOD_DIRS_32 += $(ROOT_KICONV_DIR_32)
501 ROOT_MOD_DIRS_32 += $(ROOT_FIRMWARE_DIR)

503 USR_MOD_DIR = $(ROOT)/usr/kernel

505 USR_DRV_DIR_32 = $(USR_MOD_DIR)/drv
506 USR_EXEC_DIR_32 = $(USR_MOD_DIR)/exec
507 USR_FS_DIR_32 = $(USR_MOD_DIR)/fs
508 USR_SCHED_DIR_32 = $(USR_MOD_DIR)/sched
509 USR_SOCKET_DIR_32 = $(USR_MOD_DIR)/socketmod
510 USR_STRMOD_DIR_32 = $(USR_MOD_DIR)/strmod
511 USR_SYS_DIR_32 = $(USR_MOD_DIR)/sys
512 USR_MISC_DIR_32 = $(USR_MOD_DIR)/misc
513 USR_DACF_DIR_32 = $(USR_MOD_DIR)/dacf
514 USR_PCBE_DIR_32 = $(USR_MOD_DIR)/pcbe
515 USR_DTRACE_DIR_32 = $(USR_MOD_DIR)/dtrace
516 USR_BRAND_DIR_32 = $(USR_MOD_DIR)/brand

```

```

518 USR_DRV_DIR_64      = $(USR_MOD_DIR)/drv/$(SUBDIR64)
519 USR_EXEC_DIR_64     = $(USR_MOD_DIR)/exec/$(SUBDIR64)
520 USR_FS_DIR_64       = $(USR_MOD_DIR)/fs/$(SUBDIR64)
521 USR_SCHED_DIR_64    = $(USR_MOD_DIR)/sched/$(SUBDIR64)
522 USR SOCK_DIR_64     = $(USR_MOD_DIR)/socketmod/$(SUBDIR64)
523 USR_STRMOD_DIR_64   = $(USR_MOD_DIR)/strmod/$(SUBDIR64)
524 USR_SYS_DIR_64      = $(USR_MOD_DIR)/sys/$(SUBDIR64)
525 USR_MISC_DIR_64     = $(USR_MOD_DIR)/misc/$(SUBDIR64)
526 USR_DACF_DIR_64     = $(USR_MOD_DIR)/dacf/$(SUBDIR64)
527 USR_PCBE_DIR_64     = $(USR_MOD_DIR)/pcbe/$(SUBDIR64)
528 USR_DTRACE_DIR_64  = $(USR_MOD_DIR)/dtrace/$(SUBDIR64)
529 USR_BRAND_DIR_64    = $(USR_MOD_DIR)/brand/$(SUBDIR64)

531 USR_DRV_DIR          = $(USR_DRV_DIR_$(CLASS))
532 USR_EXEC_DIR         = $(USR_EXEC_DIR_$(CLASS))
533 USR_FS_DIR           = $(USR_FS_DIR_$(CLASS))
534 USR_SCHED_DIR        = $(USR_SCHED_DIR_$(CLASS))
535 USR SOCK_DIR         = $(USR SOCK_DIR_$(CLASS))
536 USR_STRMOD_DIR       = $(USR_STRMOD_DIR_$(CLASS))
537 USR_SYS_DIR          = $(USR_SYS_DIR_$(CLASS))
538 USR_MISC_DIR         = $(USR_MISC_DIR_$(CLASS))
539 USR_DACF_DIR         = $(USR_DACF_DIR_$(CLASS))
540 USR_PCBE_DIR         = $(USR_PCBE_DIR_$(CLASS))
541 USR_DTRACE_DIR       = $(USR_DTRACE_DIR_$(CLASS))
542 USR_BRAND_DIR        = $(USR_BRAND_DIR_$(CLASS))

544 USR_MOD_DIRS_32     = $(USR_DRV_DIR_32) $(USR_EXEC_DIR_32)
545 USR_MOD_DIRS_32     += $(USR_FS_DIR_32) $(USR_SCHED_DIR_32)
546 USR_MOD_DIRS_32     += $(USR_STRMOD_DIR_32) $(USR_SYS_DIR_32)
547 USR_MOD_DIRS_32     += $(USR_MISC_DIR_32) $(USR_DACF_DIR_32)
548 USR_MOD_DIRS_32     += $(USR_PCBE_DIR_32)
549 USR_MOD_DIRS_32     += $(USR_DTRACE_DIR_32) $(USR_BRAND_DIR_32)
550 USR_MOD_DIRS_32     += $(USR SOCK_DIR_32)

552 #
553 #
554 #
555 include $(SRC)/Makefile.psm

557 #
558 #   The "-r" on the remove may be considered temporary, but is required
559 #   while the replacement of the SUNW,SPARCstation-10,SX directory by
560 #   a symbolic link is being propagated.
561 #
562 INS.slink1= $(RM) -r $@; $(SYMLINK) $(PLATFORM) $@
563 INS.slink2= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/$(@F) $@
564 INS.slink3= $(RM) -r $@; $(SYMLINK) $(IMPLEMENTED_PLATFORM) $@
565 INS.slink4= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/include $@
566 INS.slink5= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/sbin $@
567 INS.slink6= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/lib/$(MODULE) $@
568 INS.slink7= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/sbin/$(@F) $@

570 ROOT_PLAT_LINKS      = $(PLAT_LINKS:%=$(ROOT_PLAT_DIR)/%)
571 ROOT_PLAT_LINKS_2    = $(PLAT_LINKS_2:%=$(ROOT_PLAT_DIR)/%)
572 USR_PLAT_LINKS        = $(PLAT_LINKS:%=$(USR_PLAT_DIR)/%)
573 USR_PLAT_LINKS_2      = $(PLAT_LINKS_2:%=$(USR_PLAT_DIR)/%)

575 #
576 # Collection of all relevant, delivered kernel modules.
577 #
578 # Note that we insist on building genunix first, because everything else
579 # unifies against it. When doing a 'make' from usr/src/uts/, we'll enter
580 # the platform directories first. These will cd into the corresponding genunix
581 # directory and build it. So genunix /shouldn't/ get rebuilt when we get to
582 # building all the kernel modules. However, due to an as-yet-unexplained
583 # problem with dependencies, sometimes it does get rebuilt, which then messes

```

```

584 # up the other modules. So we always force the issue here rather than try to
585 # build genunix in parallel with everything else.
586 #
587 PARALLEL_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
588                 $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
589                 $(NLMISC_KMODS) $(MACH_KMODS) $(CPU_KMODS) $(GSS_KMODS) \
590                 $(MMU_KMODS) $(DACF_KMODS) $(EXPORT_KMODS) $(IPP_KMODS) \
591                 $(CRYPTO_KMODS) $(PCBE_KMODS) \
592                 $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
593                 $(BRAND_KMODS) $(KICONV_KMODS) \
594                 $(SOCKET_KMODS)

596 KMODS = $(GENUNIX_KMODS) $(PARALLEL_KMODS)

598 $(PARALLEL_KMODS): $(GENUNIX_KMODS)

600 LINT_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
601             $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
602             $(MACH_KMODS) $(GSS_KMODS) $(DACF_KMODS) $(IPP_KMODS) \
603             $(CRYPTO_KMODS) $(PCBE_KMODS) \
604             $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
605             $(BRAND_KMODS) $(KICONV_KMODS) $(SOCKET_KMODS)

607 #
608 #   Files to be compiled with -xa, to generate basic block execution
609 #   count data.
610 #
611 #   There are several ways to compile parts of the kernel for kcov:
612 #       1) Add targets to BB_FILES here or in other Makefiles
613 #           (they must in the form of $(OBJSDIR)/target.o)
614 #       2) setenv BB_FILES '$(XXX_OBJS:%=$(OBJSDIR)/%)'
615 #       3) setenv BB_FILES '$(OBJECTS)'
616 #
617 #   Do NOT setenv CFLAGS -xa, as that will cause infinite recursion
618 #   in unix_bb.o
619 #
620 BB_FILES =
621 $(BB_FILES)      := XAOPT = -xa

623 #
624 #   The idea here is for unix_bb.o to be in all kernels except the
625 #   kernel which actually gets shipped to customers. In practice,
626 #   $(RELEASE_BUILD) is on for a number of the late beta and fcs builds.
627 #
628 $(NOT_RELEASE_BUILD)$(OBJSDIR)/unix_bb.o := CPPFLAGS += -DKCOV
629 $(NOT_RELEASE_BUILD)$(OBJSDIR)/unix_bb.ln := CPPFLAGS += -DKCOV

631 #
632 #   Do not let unix_bb.o get compiled with -xa!
633 #
634 $(OBJSDIR)/unix_bb.o := XAOPT =

636 #
637 # Privilege files
638 #
639 PRIVS_AWK = $(SRC)/uts/common/os/privs.awk
640 PRIVS_DEF = $(SRC)/uts/common/os/priv_defs

642 #
643 # USB device data
644 #
645 USBDEVS_AWK = $(SRC)/uts/common/io/usb/usbdevs2h.awk
646 USBDEVS_DATA = $(SRC)/uts/common/io/usb/usbdevs

649 #

```

new/usr/src/uts/Makefile.uts

11

```
650 # If we're using the newer CTF tools, then we need to make sure that we
651 # are building with the private -X option to ctfconvert which allows us
652 # to fixup the struct cpu to account for machcpu.
653 #
654 ${BUILD_NEW_CTF_TOOLS}CTFCVTFLAGS += -X
```

```

new/usr/src/uts/common/io/comstar/port/srpt/srpt_impl.h 1
*****
13960 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/io/comstar/port/srpt/srpt_impl.h
10805 Fix for 10687 can be improved
*****
_____unchanged_portion_omitted_____

470 extern srpt_ctxt_t *srpt_ctxt;

472 /*
473  * For Non recoverable or Major Errors
474  */
475 #define SRPT_LOG_L0    0

477 /*
478  * For additional information on Non recoverable errors and
479  * warnings/informational message for sys-admin types.
480  */
481 #define SRPT_LOG_L1    1

483 /*
484  * debug only
485  * for more verbose trace than L1, for e.g. recoverable errors,
486  * or intersting trace
487  */
488 #define SRPT_LOG_L2    2

490 /*
491  * debug only
492  * for more verbose trace than L2, for e.g. printing function entries....
493  */
494 #define SRPT_LOG_L3    3

496 /*
497  * debug only
498  * for more verbose trace than L3, for e.g. printing minor function entries...
499  */
500 #define SRPT_LOG_L4    4

502 /*
503  * srpt_errlevel can be set in the debugger to enable additional logging.
504  * You can also add set srpt:srpt_errlevel={0,1,2,3,4} in /etc/system.
505  * The default log level is L1.
506  */
507 #define SRPT_LOG_DEFAULT_LEVEL SRPT_LOG_L1

509 extern uint_t srpt_errlevel;

512 #define SRPT_DPRINTF_L0(...) cmn_err(CE_WARN, __VA_ARGS__)
513 #define SRPT_DPRINTF_L1(...) cmn_err(CE_NOTE, __VA_ARGS__)
514 #define SRPT_DPRINTF_L2(...) if (srpt_errlevel >= SRPT_LOG_L2) { \
515     cmn_err(CE_NOTE, __VA_ARGS__); \
516 }
517 #ifdef DEBUG
518 #define SRPT_DPRINTF_L3(...) if (srpt_errlevel >= SRPT_LOG_L3) { \
519     cmn_err(CE_NOTE, __VA_ARGS__); \
520 }
521 #define SRPT_DPRINTF_L4(...) if (srpt_errlevel >= SRPT_LOG_L4) { \
522     cmn_err(CE_NOTE, __VA_ARGS__); \
523 }
524 #else
525 #define SRPT_DPRINTF_L3(...) (void)(0)
526 #define SRPT_DPRINTF_L4(...) (void)(0)
525 #define SRPT_DPRINTF_L3(...)
526 #define SRPT_DPRINTF_L4(...)

```

```

new/usr/src/uts/common/io/comstar/port/srpt/srpt_impl.h 2
527 #endif
529 #ifdef __cplusplus
530 }
_____unchanged_portion_omitted_____

```

new/usr/src/uts/common/io/pciex/pcieb.h

1

6526 Mon Apr 15 12:20:22 2019

new/usr/src/uts/common/io/pciex/pcieb.h

10805 Fix for 10687 can be improved

```
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 (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
23 */

25 /*
26 * Copyright 2019, Joyent, Inc.
27 */

29 #ifndef _SYS_PCIEB_H
30 #define _SYS_PCIEB_H

32 #ifdef __cplusplus
33 extern "C" {
34 #endif

36 #if defined(DEBUG)
37 #define PCIEB_DEBUG pcieb_dbg
38 extern void pcieb_dbg(uint_t bit, dev_info_t *dip, char *fmt, ...);
39 #else /* DEBUG */
40 #define PCIEB_DEBUG(...) (void)(0)
40 #define PCIEB_DEBUG(...)
41 #endif /* DEBUG */

43 typedef enum { /* same sequence as pcieb_debug_sym[] */
44     /* 0 */ DBG_ATTACH,
45     /* 1 */ DBG_PWR,
46     /* 2 */ DBG_INTR
47 } pcieb_debug_bit_t;
unchanged_portion_omitted
```


new/usr/src/uts/common/sys/ib/clients/eoib/eib_impl.h

1

```
*****
29303 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/sys/ib/clients/eoib/eib_impl.h
10805 Fix for 10687 can be improved
*****
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 (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 */
26 /*
27  * Copyright 2019, Joyent, Inc.
28 */
30 #ifndef _SYS_IB_EOIB_EIB_IMPL_H
31 #define _SYS_IB_EOIB_EIB_IMPL_H
33 #ifdef __cplusplus
34 extern "C" {
35 #endif
37 #include <sys/ddi.h>
38 #include <sys/mac.h>
39 #include <sys/sunddi.h>
40 #include <sys/varargs.h>
41 #include <sys/vlan.h>
42 #include <sys/ib/ibt1/ibt1.h>
43 #include <sys/ib/ibt1/ibvt1.h>
44 #include <sys/ib/ib_pkt_hdrs.h>
46 #include <sys/ib/clients/eoib/fip.h>
47 #include <sys/ib/clients/eoib/eib.h>
49 /*
50  * Driver specific constants
51  */
52 #define EIB_E_SUCCESS 0
53 #define EIB_E_FAILURE -1
54 #define EIB_MAX_LINE 128
55 #define EIB_MAX_SGL 59
56 #define EIB_MAX_POST_MULTIPLE 4
57 #define EIB_MAX_PAYLOAD_HDR_SZ 160
58 #define EIB_TX_COPY_THRESH 4096 /* greater than mtu */
59 #define EIB_MAX_VNIC_S 64 /* do not change this */
60 #define EIB_LOGIN_TIMEOUT_USEC 8000000
61 #define EIB_RWR_CHUNK_SZ 8
```

new/usr/src/uts/common/sys/ib/clients/eoib/eib_impl.h

2

```
62 #define EIB_IPHDR_ALIGN_ROOM 32
63 #define EIB_IP_HDR_ALIGN 2
64 #define EIB_MAX_RX_PKTS_ONINTR 0x800
65 #define EIB_MAX_LOGIN_ATTEMPTS 3
66 #define EIB_MAX_VHUB_TBL_ATTEMPTS 3
67 #define EIB_MAX_KA_ATTEMPTS 3
68 #define EIB_MAX_ATTEMPTS 10
69 #define EIB_DELAY_HALF_SECOND 500000
70 #define EIB_GRH_SZ (sizeof (ib_grh_t))
72 /*
73  * Debug messages
74  */
75 #define EIB_MSGS_CRIT 0x01
76 #define EIB_MSGS_ERR 0x02
77 #define EIB_MSGS_WARN 0x04
78 #define EIB_MSGS_DEBUG 0x08
79 #define EIB_MSGS_ARGS 0x10
80 #define EIB_MSGS_PKT 0x20
81 #define EIB_MSGS_VERBOSE 0x40
82 #define EIB_MSGS_DEFAULT (EIB_MSGS_CRIT | EIB_MSGS_ERR | EIB_MSGS_WARN)
84 #define EIB_LOGSZ_DEFAULT 0x20000
86 #define EIB_DPRINTF_CRIT eib_dprintf_crit
87 #define EIB_DPRINTF_ERR eib_dprintf_err
88 #define EIB_DPRINTF_WARN eib_dprintf_warn
89 #ifdef EIB_DEBUG
90 #define EIB_DPRINTF_DEBUG eib_dprintf_debug
91 #define EIB_DPRINTF_ARGS eib_dprintf_args
92 #define EIB_DPRINTF_PKT eib_dprintf_pkt
93 #define EIB_DPRINTF_VERBOSE eib_dprintf_verbose
94 #else
95 #define EIB_DPRINTF_DEBUG(...) (void)(0)
96 #define EIB_DPRINTF_ARGS(...) (void)(0)
97 #define EIB_DPRINTF_PKT(...) (void)(0)
98 #define EIB_DPRINTF_VERBOSE(...) (void)(0)
99 #endif
101 /*
102  * EoIB threads to provide various services
103  */
104 #define EIB_EVENTS_HDLR "eib_events_handler"
105 #define EIB_RWQES_REFILLER "eib_rwqes_refiller"
106 #define EIB_VNIC_CREATOR "eib_vnic_creator"
107 #define EIB_TXWQES_MONITOR "eib_txwqes_monitor"
108 #define EIB_LSOBUFS_MONITOR "eib_lsobufs_monitor"
110 /*
111  * Macro for finding the least significant bit set in a 64-bit unsigned int
112  */
113 #define EIB_FIND_LSB_SET(val64) eib_setbit_mod67[(((val64) & (val64)) % 67)]
115 /*
116  * LSO buffers
117  *
118  * Under normal circumstances we should never need to use any buffer
119  * that's larger than MTU. Unfortunately, IB HCA has limitations
120  * on the length of SGL that are much smaller than those for regular
121  * ethernet NICs. Since the network layer doesn't care to limit the
122  * number of mblk fragments in any send up chain, we end up having to
123  * use these larger buffers occasionally.
```

```
124 */
125 #define EIB_LSO_MAXLEN          65536
126 #define EIB_LSO_BUFSZ          8192
127 #define EIB_LSO_NUM_BUFS       1024
128 #define EIB_LSO_FREE_BUFS_THRESH (EIB_LSO_NUM_BUFS >> 5)

130 typedef struct eib_lsobuf_s {
131     struct eib_lsobuf_s *lb_next;
132     uint8_t *lb_buf;
133     int lb_isfree;
134 } eib_lsobuf_t;
_____ unchanged_portion_omitted
```

new/usr/src/uts/common/sys/ib/clients/eoib/enx_impl.h

1

```
*****
14148 Mon Apr 15 12:20:22 2019
new/usr/src/uts/common/sys/ib/clients/eoib/enx_impl.h
10805 Fix for 10687 can be improved
*****
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 (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 */

26 /*
27  * Copyright 2019, Joyent, Inc.
28 */

30 #ifndef _SYS_IB_EOIB_ENX_IMPL_H
31 #define _SYS_IB_EOIB_ENX_IMPL_H

33 #ifdef __cplusplus
34 extern "C" {
35 #endif

37 #include <sys/ddi.h>
38 #include <sys/sunddi.h>
39 #include <sys/varargs.h>
40 #include <sys/ib/ibt1/ibti.h>
41 #include <sys/ib/ibt1/ibvti.h>
42 #include <sys/ib/ib_pkt_hdrs.h>
43 #include <sys/ib/ibt1/impl/ibt1_ibnex.h>
44 #include <sys/ib/mgt/sm_attr.h>

46 #include <sys/ib/clients/eoib/fip.h>
47 #include <sys/ib/clients/eoib/eib.h>

49 /*
50  * Driver specific constants
51  */
52 #define ENX_E_SUCCESS          0
53 #define ENX_E_FAILURE         -1
54 #define ENX_MAX_LINE          128
55 #define ENX_GRH_SZ             (sizeof (ib_grh_t))

57 /*
58  * Debug messages
59  */
60 #define ENX_MSGS_CRIT          0x01
61 #define ENX_MSGS_ERR           0x02
```

new/usr/src/uts/common/sys/ib/clients/eoib/enx_impl.h

2

```
62 #define ENX_MSGS_WARN          0x04
63 #define ENX_MSGS_DEBUG         0x08
64 #define ENX_MSGS_ARGS          0x10
65 #define ENX_MSGS_VERBOSE       0x20
66 #define ENX_MSGS_DEFAULT       (ENX_MSGS_CRIT | ENX_MSGS_ERR | ENX_MSGS_WARN)

68 #define ENX_LOGSZ_DEFAULT       0x20000

70 #define ENX_DPRINTF_CRIT       eibnx_dprintf_crit
71 #define ENX_DPRINTF_ERR        eibnx_dprintf_err
72 #define ENX_DPRINTF_WARN       eibnx_dprintf_warn
73 #ifdef ENX_DEBUG
74 #define ENX_DPRINTF_DEBUG       eibnx_dprintf_debug
75 #define ENX_DPRINTF_ARGS       eibnx_dprintf_args
76 #define ENX_DPRINTF_VERBOSE    eibnx_dprintf_verbose
77 #else
78 #define ENX_DPRINTF_DEBUG(...) (void)(0)
79 #define ENX_DPRINTF_ARGS(...) (void)(0)
80 #define ENX_DPRINTF_VERBOSE(...) (void)(0)
81 #endif

83 /*
84  * EoIB Nexus service threads
85  */
86 #define ENX_PORT_MONITOR        "eibnx_port_%d_monitor"
87 #define ENX_NODE_CREATOR       "eibnx_node_creator"

89 /*
90  * Default period (us) for unicast solicitations to discovered gateways.
91  * EoIB specification requires that hosts send solicitation atleast every
92  * 4 * GW_ADV_PERIOD.
93  */
94 #define ENX_DFL_SOLICIT_PERIOD_USEC    3200000

96 /*
97  * Portinfo list per HCA
98  */
99 typedef struct eibnx_port_s {
100     struct eibnx_port_s    *po_next;
101     ibt_hca_portinfo_t     *po_pi;
102     uint_t                  po_pi_size;
103 } eibnx_port_t;
_____ unchanged portion omitted
```

new/usr/src/uts/common/sys/ib/clients/rds/rdsib_debug.h

1

```
*****
2872 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/clients/rds/rdsib_debug.h
10805 Fix for 10687 can be improved
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 */

26 /*
27 * Copyright 2019, Joyent, Inc.
28 */

30 #ifndef _RDSIB_DEBUG_H
31 #define _RDSIB_DEBUG_H

33 #ifdef __cplusplus
34 extern "C" {
35 #endif

37 #define LABEL "RDS"

39 /*
40 * warnings, console & syslog buffer.
41 * For Non recoverable or Major Errors
42 */
43 #define RDS_LOG_L0 0

45 /*
46 * syslog buffer or RDS trace buffer (console if booted /w debug)
47 * For additional information on Non recoverable errors and
48 * warnings/informational message for sys-admin types.
49 */
50 #define RDS_LOG_L1 1

52 /*
53 * debug only
54 * for more verbose trace than L1, for e.g. recoverable errors,
55 * or interesting trace
56 */
57 #define RDS_LOG_L2 2

59 /*
60 * debug only
61 * for more verbose trace than L2, for e.g. informational messages
```

new/usr/src/uts/common/sys/ib/clients/rds/rdsib_debug.h

2

```
62 */
63 #define RDS_LOG_L3 3

65 /*
66 * debug only
67 * for more verbose trace than L3, for e.g. printing function entries...
68 */
69 #define RDS_LOG_L4 4

71 /*
72 * debug only
73 * most verbose level. Used only for excessive trace, for e.g.
74 * printing structures etc.
75 */
76 #define RDS_LOG_L5 5

78 /*
79 * debug only
80 * for messages from softints, taskqs, intr handlers, timeout handlers etc.
81 */
82 #define RDS_LOG_LINTR 6

85 #ifdef DEBUG
86 #define RDS_DPRINTF_INTR rds_dprintf_intr
87 #define RDS_DPRINTF5 rds_dprintf5
88 #define RDS_DPRINTF4 rds_dprintf4
89 #define RDS_DPRINTF3 rds_dprintf3

91 void rds_dprintf_intr(
92     char *name,
93     char *fmt, ...);
94 void rds_dprintf5(
95     char *name,
96     char *fmt, ...);
97 void rds_dprintf4(
98     char *name,
99     char *fmt, ...);
100 void rds_dprintf3(
101     char *name,
102     char *fmt, ...);
103 #else
104 #define RDS_DPRINTF_INTR(...) (void)(0)
105 #define RDS_DPRINTF5(...) (void)(0)
106 #define RDS_DPRINTF4(...) (void)(0)
107 #define RDS_DPRINTF3(...) (void)(0)
108 #endif

110 #define RDS_DPRINTF2 rds_dprintf2
111 #define RDS_DPRINTF1 rds_dprintf1
112 #define RDS_DPRINTF0 rds_dprintf0

114 void rds_dprintf2(
115     char *name,
116     char *fmt, ...);
117 void rds_dprintf1(
118     char *name,
119     char *fmt, ...);
120 void rds_dprintf0(
121     char *name,
122     char *fmt, ...);
```

new/usr/src/uts/common/sys/ib/clients/rds/rdsib_debug.h

3

```
124 #ifdef __cplusplus
125 }
_____unchanged_portion_omitted_
```

new/usr/src/uts/common/sys/ib/clients/rdsv3/rdsv3_debug.h 1

```
*****
3204 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/clients/rdsv3/rdsv3_debug.h
10805 Fix for 10687 can be improved
*****
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 (c) 2010, Oracle and/or its affiliates. All rights reserved.
23 */

25 /*
26 * Copyright 2019, Joyent, Inc.
27 */

29 #ifndef _RDSV3_DEBUG_H
30 #define _RDSV3_DEBUG_H

32 #ifdef __cplusplus
33 extern "C" {
34 #endif

36 #define LABEL "RDSV3"

38 /*
39 * warnings, console & syslog buffer.
40 * For Non recoverable or Major Errors
41 */
42 #define RDSV3_LOG_L0 0

44 /*
45 * syslog buffer or RDS trace buffer (console if booted /w debug)
46 * For additional information on Non recoverable errors and
47 * warnings/informational message for sys-admin types.
48 */
49 #define RDSV3_LOG_L1 1

51 /*
52 * debug only
53 * for more verbose trace than L1, for e.g. recoverable errors,
54 * or interesting trace
55 */
56 #define RDSV3_LOG_L2 2

58 /*
59 * debug only
60 * for more verbose trace than L2, for e.g. informational messages
61 */
```

new/usr/src/uts/common/sys/ib/clients/rdsv3/rdsv3_debug.h 2

```
62 #define RDSV3_LOG_L3 3

64 /*
65 * debug only
66 * for more verbose trace than L3, for e.g. printing function entries...
67 */
68 #define RDSV3_LOG_L4 4

70 /*
71 * debug only
72 * most verbose level. Used only for excessive trace, for e.g.
73 * printing structures etc.
74 */
75 #define RDSV3_LOG_L5 5

77 /*
78 * debug only
79 * for messages from softints, taskqs, intr handlers, timeout handlers etc.
80 */
81 #define RDSV3_LOG_LINTR 6

84 #ifdef DEBUG
85 #define RDSV3_DPRINTF_INTR rdsv3_dprintf_intr
86 #define RDSV3_DPRINTF5 rdsv3_dprintf5
87 #define RDSV3_DPRINTF4 rdsv3_dprintf4
88 #define RDSV3_DPRINTF3 rdsv3_dprintf3

90 void rdsv3_dprintf_intr(
91     char *name,
92     char *fmt, ...);
93 void rdsv3_dprintf5(
94     char *name,
95     char *fmt, ...);
96 void rdsv3_dprintf4(
97     char *name,
98     char *fmt, ...);
99 void rdsv3_dprintf3(
100    char *name,
101    char *fmt, ...);
102 #else
103 #define RDSV3_DPRINTF_INTR(...) (void)(0)
104 #define RDSV3_DPRINTF5(...) (void)(0)
105 #define RDSV3_DPRINTF4(...) (void)(0)
106 #define RDSV3_DPRINTF3(...) (void)(0)
107 #endif

109 #define RDSV3_DPRINTF2 rdsv3_dprintf2
110 #define RDSV3_DPRINTF1 rdsv3_dprintf1
111 #define RDSV3_DPRINTF0 rdsv3_dprintf0

113 void rdsv3_dprintf2(
114     char *name,
115     char *fmt, ...);
116 void rdsv3_dprintf1(
117     char *name,
118     char *fmt, ...);
119 void rdsv3_dprintf0(
120     char *name,
121     char *fmt, ...);

123 void rdsv3_trace(
```

```
124         char          *name,  
125         uint8_t       lvl,  
126         char          *fmt, ...);
```

```
128 void rdsv3_vprintk(  
129     char          *name,  
130     uint8_t       lvl,  
131     const char    *fmt,  
132     va_list       ap);
```

```
134 /* defined in rds_debug.c */  
135 void rdsv3_logging_initialization();  
136 void rdsv3_logging_destroy();  
137 int rdsv3_printk_ratelimit(void);
```

```
139 #ifdef __cplusplus  
140 }
```

unchanged_portion_omitted

```
new/usr/src/uts/common/sys/ib/ibtl/impl/ibtl_util.h
```

1

```
*****
3745 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/ib/ibtl/impl/ibtl_util.h
10805 Fix for 10687 can be improved
*****
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, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23  * Copyright 2004 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */

27 /*
28  * Copyright 2019, Joyent, Inc.
29 */

31 #ifndef _SYS_IB_IBTL_IMPL_IBTL_UTIL_H
32 #define _SYS_IB_IBTL_IMPL_IBTL_UTIL_H

34 /*
35  * ibtl_util.h
36  *
37  * All data structures and function prototypes that serve as helper
38  * routines for IBTF implementation.
39  */

41 #ifdef __cplusplus
42 extern "C" {
43 #endif

45 #include <sys/ib/ib_types.h>
46 #include <sys/varargs.h>

48 /*
49  * Time Related Functions
50  *
51  *   ibt_usec2ib
52  *   This function converts the standard input time in microseconds to
53  *   IB's 6 bits of timeout exponent, calculated based on
54  *   time = 4.096us * 2 ^ exp.
55  *
56  *   ibt_ib2usec
57  *   This function converts the input IB timeout exponent (6 bits) to
58  *   standard time in microseconds, calculated based on
59  *   time = 4.096us * 2 ^ exp.
60  */
61 ib_time_t      ibt_usec2ib(clock_t microseconds);
```

```
new/usr/src/uts/common/sys/ib/ibtl/impl/ibtl_util.h
```

2

```
62 clock_t      ibt_ib2usec(ib_time_t ib_time);

65 /*
66  * IB logging, debug and console message handling
67  */

70 /*
71  * warnings, console & syslog buffer.
72  * For Non recoverable or Major Errors
73  */
74 #define IBTF_LOG_L0      0

76 /*
77  * syslog buffer or IBTF trace buffer (console if booted /w debug)
78  * For additional information on Non recoverable errors and
79  * warnings/informational message for sys-admin types.
80  */
81 #define IBTF_LOG_L1      1

83 /*
84  * debug only
85  * for more verbose trace than L1, for e.g. recoverable errors,
86  * or interesting trace
87  */
88 #define IBTF_LOG_L2      2

90 /*
91  * debug only
92  * for more verbose trace than L2, for e.g. printing function entries....
93  */
94 #define IBTF_LOG_L3      3

96 /*
97  * debug only
98  * for more verbose trace than L3, for e.g. printing minor function entries...
99  */
100 #define IBTF_LOG_L4      4

102 /*
103  * debug only
104  * most verbose level. Used only for excessive trace, for e.g.
105  * printing structures etc.
106  */
107 #define IBTF_LOG_L5      5

109 /*
110  * debug only
111  * for messages from softints, taskqs, intr handlers, timeout handlers etc.
112  * Only gets printed if "ibtl_allow_intr_msgs" is set
113  */
114 #define IBTF_LOG_LINTR  6

117 #ifdef DEBUG
118 #define IBTF_DPRINTF_LINTR      ibtl_dprintf_intr
119 #define IBTF_DPRINTF_L5        ibtl_dprintf5
120 #define IBTF_DPRINTF_L4        ibtl_dprintf4
121 #define IBTF_DPRINTF_L3        ibtl_dprintf3

123 void ibtl_dprintf_intr(
124     char      *name,
125     char      *fmt, ...);
126 void ibtl_dprintf5(
127     char      *name,
```



```
128         char          *fmt, ...);
129 void ibtl_dprintf4(
130     char          *name,
131     char          *fmt, ...);
132 void ibtl_dprintf3(
133     char          *name,
134     char          *fmt, ...);
135 #else
136 #define IBTF_DPRINTF_LINTR(...) (void)(0)
137 #define IBTF_DPRINTF_L5(...) (void)(0)
138 #define IBTF_DPRINTF_L4(...) (void)(0)
139 #define IBTF_DPRINTF_L3(...) (void)(0)
136 #define IBTF_DPRINTF_LINTR(...)
137 #define IBTF_DPRINTF_L5(...)
138 #define IBTF_DPRINTF_L4(...)
139 #define IBTF_DPRINTF_L3(...)
140 #endif

142 #define IBTF_DPRINTF_L2 ibtl_dprintf2
143 #define IBTF_DPRINTF_L1 ibtl_dprintf1
144 #define IBTF_DPRINTF_L0 ibtl_dprintf0

146 void ibtl_dprintf2(
147     char          *name,
148     char          *fmt, ...);
149 void ibtl_dprintf1(
150     char          *name,
151     char          *fmt, ...);
152 void ibtl_dprintf0(
153     char          *name,
154     char          *fmt, ...);

156 #ifdef __cplusplus
157 }
    unchanged_portion_omitted
```

```

*****
24339 Mon Apr 15 12:20:23 2019
new/usr/src/uts/common/sys/pcie_impl.h
10805 Fix for 10687 can be improved
*****
_____unchanged_portion_omitted_____

452 /*
453  * Default interrupt priority for all PCI and PCIe nexus drivers including
454  * hotplug interrupts.
455  */
456 #define PCIE_INTR_PRI          (LOCK_LEVEL - 1)

458 /*
459  * XXX - PCIE_IS_PCIE check is required in order not to invoke these macros
460  * for non-standard PCI or PCI Express Hotplug Controllers.
461  */
462 #define PCIE_ENABLE_ERRORS(dip) \
463     if (PCIE_IS_PCIE(PCIE_DIP2BUS(dip))) { \
464         pcie_enable_errors(dip); \
465         (void) pcie_enable_ce(dip); \
466     }

468 #define PCIE_DISABLE_ERRORS(dip) \
469     if (PCIE_IS_PCIE(PCIE_DIP2BUS(dip))) { \
470         pcie_disable_errors(dip); \
471     }

473 /*
474  * pcie_init_busp_cie_fini_bus specific flags
475  */
476 #define PCIE_BUS_INITIAL        0x0001
477 #define PCIE_BUS_FINAL         0x0002
478 #define PCIE_BUS_ALL           (PCIE_BUS_INITIAL | PCIE_BUS_FINAL)

480 #ifdef DEBUG
481 #define PCIE_DBG pcie_dbg
482 /* Common Debugging shortcuts */
483 #define PCIE_DBG_CFG(dip, bus_p, name, sz, off, org) \
484     PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
485         ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
486         PCIE_GET(sz, bus_p, off))
487 #define PCIE_DBG_CAP(dip, bus_p, name, sz, off, org) \
488     PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
489         ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
490         PCIE_CAP_GET(sz, bus_p, off))
491 #define PCIE_DBG_AER(dip, bus_p, name, sz, off, org) \
492     PCIE_DBG("%s:%d:(0x%x) %s(0x%x) 0x%x -> 0x%x\n", ddi_node_name(dip), \
493         ddi_get_instance(dip), bus_p->bus_bdf, name, off, org, \
494         PCIE_AER_GET(sz, bus_p, off))

496 #else /* DEBUG */

498 #define PCIE_DBG_CFG(...)      (void)(0)
499 #define PCIE_DBG(...)         (void)(0)
500 #define PCIE_ARI_DBG(...)     (void)(0)
501 #define PCIE_DBG_CAP(...)    (void)(0)
502 #define PCIE_DBG_AER(...)    (void)(0)
498 #define PCIE_DBG_CFG(...)
499 #define PCIE_DBG(...)
500 #define PCIE_ARI_DBG(...)
501 #define PCIE_DBG_CAP(...)
502 #define PCIE_DBG_AER(...)

504 #endif /* DEBUG */

```

```

506 /* PCIe Friendly Functions */
507 extern int pcie_init(dev_info_t *dip, caddr_t arg);
508 extern int pcie_uninit(dev_info_t *dip);
509 extern int pcie_hpintr_enable(dev_info_t *dip);
510 extern int pcie_hpintr_disable(dev_info_t *dip);
511 extern int pcie_intr(dev_info_t *dip);
512 extern int pcie_open(dev_info_t *dip, dev_t *devp, int flags, int otyp,
513     cred_t *credp);
514 extern int pcie_close(dev_info_t *dip, dev_t dev, int flags, int otyp,
515     cred_t *credp);
516 extern int pcie_ioctl(dev_info_t *dip, dev_t dev, int cmd, intptr_t arg,
517     int mode, cred_t *credp, int *rvalp);
518 extern int pcie_prop_op(dev_t dev, dev_info_t *dip, ddi_prop_op_t prop_op,
519     int flags, char *name, caddr_t valuep, int *lengthp);

521 extern void pcie_init_root_port_mps(dev_info_t *dip);
522 extern int pcie_initchild(dev_info_t *dip);
523 extern void pcie_unitchild(dev_info_t *dip);
524 extern int pcie_init_cfghdl(dev_info_t *dip);
525 extern void pcie_fini_cfghdl(dev_info_t *dip);
526 extern void pcie_clear_errors(dev_info_t *dip);
527 extern int pcie_postattach_child(dev_info_t *dip);
528 extern void pcie_enable_errors(dev_info_t *dip);
529 extern void pcie_disable_errors(dev_info_t *dip);
530 extern int pcie_enable_ce(dev_info_t *dip);
531 extern boolean_t pcie_bridge_is_link_disabled(dev_info_t *);

533 extern pcie_bus_t *pcie_init_bus(dev_info_t *dip, pcie_req_id_t bdf,
534     uint8_t flags);
535 extern void pcie_fini_bus(dev_info_t *dip, uint8_t flags);
536 extern void pcie_fab_init_bus(dev_info_t *dip, uint8_t flags);
537 extern void pcie_fab_fini_bus(dev_info_t *dip, uint8_t flags);
538 extern void pcie_rc_init_bus(dev_info_t *dip);
539 extern void pcie_rc_fini_bus(dev_info_t *dip);
540 extern void pcie_rc_init_pfd(dev_info_t *dip, pf_data_t *pfd);
541 extern void pcie_rc_fini_pfd(pf_data_t *pfd);
542 extern boolean_t pcie_is_child(dev_info_t *dip, dev_info_t *rdip);
543 extern int pcie_get_bdf_from_dip(dev_info_t *dip, pcie_req_id_t *bdf);
544 extern dev_info_t *pcie_get_my_childs_dip(dev_info_t *dip, dev_info_t *rdip);
545 extern uint32_t pcie_get_bdf_for_dma_xfer(dev_info_t *dip, dev_info_t *rdip);
546 extern int pcie_dev(dev_info_t *dip);
547 extern void pcie_get_fabric_mps(dev_info_t *rc_dip, dev_info_t *dip,
548     int *max_supported);
549 extern int pcie_root_port(dev_info_t *dip);
550 extern int pcie_initchild_mps(dev_info_t *dip);
551 extern void pcie_set_rber_fatal(dev_info_t *dip, boolean_t val);
552 extern boolean_t pcie_get_rber_fatal(dev_info_t *dip);

554 extern uint32_t pcie_get_aer_uce_mask();
555 extern uint32_t pcie_get_aer_ce_mask();
556 extern uint32_t pcie_get_aer_suce_mask();
557 extern uint32_t pcie_get_serr_mask();
558 extern void pcie_set_aer_uce_mask(uint32_t mask);
559 extern void pcie_set_aer_ce_mask(uint32_t mask);
560 extern void pcie_set_aer_suce_mask(uint32_t mask);
561 extern void pcie_set_serr_mask(uint32_t mask);
562 extern void pcie_init_plat(dev_info_t *dip);
563 extern void pcie_fini_plat(dev_info_t *dip);
564 extern int pcie_read_only_probe(dev_info_t *, char *, dev_info_t **);
565 extern dev_info_t *pcie_func_to_dip(dev_info_t *dip, pcie_req_id_t function);
566 extern int pcie_ari_disable(dev_info_t *dip);
567 extern int pcie_ari_enable(dev_info_t *dip);

569 #define PCIE_ARI_FORW_NOT_SUPPORTED    0
570 #define PCIE_ARI_FORW_SUPPORTED      1

```

```
572 extern int pcie_ari_supported(dev_info_t *dip);

574 #define PCIE_ARI_FORW_DISABLED 0
575 #define PCIE_ARI_FORW_ENABLED 1

577 extern int pcie_ari_is_enabled(dev_info_t *dip);

579 #define PCIE_NOT_ARI_DEVICE 0
580 #define PCIE_ARI_DEVICE 1

582 extern int pcie_ari_device(dev_info_t *dip);
583 extern int pcie_ari_get_next_function(dev_info_t *dip, int *func);

585 /* PCIe error handling functions */
586 extern void pf_eh_enter(pcie_bus_t *bus_p);
587 extern void pf_eh_exit(pcie_bus_t *bus_p);
588 extern int pf_scan_fabric(dev_info_t *rpdip, ddi_fm_error_t *derr,
589     pf_data_t *root_pfd_p);
590 extern void pf_init(dev_info_t *, ddi_iblock_cookie_t, ddi_attach_cmd_t);
591 extern void pf_fini(dev_info_t *, ddi_detach_cmd_t);
592 extern int pf_hdl_lookup(dev_info_t *, uint64_t, uint32_t, uint64_t,
593     pcie_req_id_t);
594 extern int pf_tlp_decode(pcie_bus_t *, pf_pcie_adv_err_regs_t *);
595 extern void pcie_force_fullscan();

597 #ifdef DEBUG
598 extern uint_t pcie_debug_flags;
599 extern void pcie_dbg(char *fmt, ...);
600 #endif /* DEBUG */

602 /* PCIe IOV functions */
603 extern dev_info_t *pcie_find_dip_by_bdf(dev_info_t *rootp, pcie_req_id_t bdf);

605 extern boolean_t pf_in_bus_range(pcie_bus_t *, pcie_req_id_t);
606 extern boolean_t pf_in_assigned_addr(pcie_bus_t *, uint64_t);
607 extern int pf_pci_decode(pf_data_t *, uint16_t *);
608 extern pcie_bus_t *pf_find_busp_by_bdf(pf_impl_t *, pcie_req_id_t);
609 extern pcie_bus_t *pf_find_busp_by_addr(pf_impl_t *, uint64_t);
610 extern pcie_bus_t *pf_find_busp_by_aer(pf_impl_t *, pf_data_t *);
611 extern pcie_bus_t *pf_find_busp_by_saer(pf_impl_t *, pf_data_t *);

613 extern int pciev_eh(pf_data_t *, pf_impl_t *);
614 extern pcie_bus_t *pciev_get_affected_dev(pf_impl_t *, pf_data_t *,
615     uint16_t, uint16_t);
616 extern void pciev_eh_exit(pf_data_t *, uint_t);
617 extern boolean_t pcie_in_domain(pcie_bus_t *, uint_t);

619 #define PCIE_ZALLOC(data) kmem_zalloc(sizeof (data), KM_SLEEP)

622 #ifdef __cplusplus
623 }
    unchanged_portion_omitted

```

new/usr/src/uts/common/sys/usb/scsa2usb/scsa2usb.h

1

25821 Mon Apr 15 12:20:23 2019

new/usr/src/uts/common/sys/usb/scsa2usb/scsa2usb.h

10805 Fix for 10687 can be improved

_____ unchanged portion omitted _____

```
624 #define SCSA2USB_MK_32BIT(a, b, c, d) \
625     (((a) << 24) | ((b) << 16) | ((c) << 8) | (d))

627 /* position of fields for SCMD_READ_CD CDB */
628 #define SCSA2USB_READ_CD_LEN_0 6 /* LEN[0] of SCMD_READ_CD */
629 #define SCSA2USB_READ_CD_LEN_1 7 /* LEN[1] of SCMD_READ_CD */
630 #define SCSA2USB_READ_CD_LEN_2 8 /* LEN[2] of SCMD_READ_CD */

632 /* macro to calculate LEN for SCMD_READ_CD command */
633 #define SCSA2USB_LEN_READ_CD(pkt) \
634     (((pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_0] << 16) + \
635      ((pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_1] << 8) + \
636      (pkt)->pkt_cdbp[SCSA2USB_READ_CD_LEN_2])

638 /* Figure out Block Size before issuing a WRITE to CD-RW device */
639 #define SCSA2USB_CDRW_BKLSZ(bcount, len) ((bcount) / (len));
640 #define SCSA2USB_VALID_CDRW_BKLSZ(blksz) \
641     (((blksz) == CDROM_BLK_2048) || ((blksz) == CDROM_BLK_2352) || \
642      ((blksz) == CDROM_BLK_2336) || ((blksz) == CDROM_BLK_2324) || \
643      ((blksz) == 0))

645 /* debug and error msg logging */
646 #define DPRINT_MASK_SCSA 0x0001 /* for SCSA */
647 #define DPRINT_MASK_ATT A 0x0002 /* for ATTA */
648 #define DPRINT_MASK_EVENTS 0x0004 /* for event handling */
649 #define DPRINT_MASK_CALLBACKS 0x0008 /* for callbacks */
650 #define DPRINT_MASK_TIMEOUT 0x0010 /* for timeouts */
651 #define DPRINT_MASK_DUMPING 0x0020 /* for dumping */
652 #define DPRINT_MASK_PM 0x0040 /* for pwr mgmt */
653 #define DPRINT_MASK_ALL 0xffffffff /* for everything */

655 #ifdef DEBUG
656 #define SCSA2USB_PRINT_CDB scsa2usb_print_cdb
657 #else
658 #define SCSA2USB_PRINT_CDB(...) (void)(0)
659 #define SCSA2USB_PRINT_CDB(...)
660 #endif

661 /* ugen support */
662 #define SCSA2USB_MINOR_UGEN_BITS_MASK 0xff
663 #define SCSA2USB_MINOR_INSTANCE_MASK ~SCSA2USB_MINOR_UGEN_BITS_MASK
664 #define SCSA2USB_MINOR_INSTANCE_SHIFT 8

666 #define SCSA2USB_MINOR_TO_INSTANCE(minor) \
667     (((minor) & SCSA2USB_MINOR_INSTANCE_MASK) >> \
668      SCSA2USB_MINOR_INSTANCE_SHIFT)

670 #ifdef __cplusplus
671 }
```

_____ unchanged portion omitted _____