```
**********************************************************
    3456 Mon Jun 16 21:18:03 2014
new/usr/src/man/man7d/mpt_sas.7d
NEX-1889 upstream
**********************************************************
```

```
'\" te
.\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved
.\" Copyright 2014, Nexenta Systems, Inc. All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" or http://www.opensolaris.org/os/licensing.  See the License for the specifi
.\" the following below this CDDL HEADER, with the fields enclosed by brackets "
.TH MPT_SAS 7D "Apr 24, 2014"
.TH MPT_SAS 7D "Jul 16, 2009"
.SH NAME
mpt_sas \- SAS-2/3 host bus adapter driver
mpt_sas \- SAS-2 host bus adapter driver
.SH SYNOPSIS
.sp
.in +2
.nf
scsi@unit-address
.fi
.in -2

.SH DESCRIPTION
.sp
.LP
The \fBmpt_sas\fR host bus adapter driver is a nexus driver that supports the
LSI SAS200x/2x08 and SAS300x/3x08 series of chips. These chips support SAS/SATA
interfaces, including tagged and untagged queuing, SATA 3G/SAS 3G/SAS 6G/SAS
12G.
LSI SAS200x/2108 series of chips. These chips support SAS/SATA interfaces,
including tagged and untagged queuing, SATA 3G/SAS 3G/SAS 6G.
.SS "Configuration"
.sp
.LP
The \fBmpt_sas\fR driver is configured by defining properties in
\fBmpt_sas.conf\fR. These properties override the global SCSI settings. The
\fBmpt_sas\fR driver supports one modifiable property:
.sp
.ne 2
.na
\fB\fBmpxio-disable\fR\fR
.ad
.sp .6
.RS 4n
Solaris I/O multipathing is enabled or disabled on SAS devices with the
\fBmpxio-disable\fR property. Specifying \fBmpxio-disable="no"\fR activates I/O
multipathing, while \fBmpxio-disable="yes"\fR disables I/O multipathing.
.sp
Solaris I/O multipathing can be enabled or disabled on a per port basis. Per
port settings override the global setting for the specified ports.
.sp
The following example shows how to disable multipathing on port 0 whose parent
is \fB/pci@0,0/pci8086,2940@1c/pci1000,72@0\fR:
.sp
.in +2
.nf
name="mpt_sas"    parent="/pci@0,0/pci8086,2940@1c/pci1000,72@0"
mpxio-disable="yes";
.fi
.in -2

.RE

.SH EXAMPLES
```



```
.LP
\fBExample 1 \fRUsing the \fBmpt_sas\fR Configuration File to Disable MPXIO
.sp
.LP
Create a file called \fB/kernel/drv/mpt_sas.conf\fR and add the following line:

.sp
.in +2
.nf
name="mpt_sas"    parent="/pci@0,0/pci8086,2940@1c/pci1000,72@0"
mpxio-disable="yes";
.fi
.in -2

.SH FILES
.sp
.ne 2
.na
\fB\fB/kernel/drv/mpt_sas\fR\fR
.ad
.sp .6
.RS 4n
32-bit ELF kernel module
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/sparcv9/mpt_sas\fR\fR
.ad
.sp .6
.RS 4n
64-bit SPARC ELF kernel module
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/amd64/mpt_sas\fR\fR
.ad
.sp .6
.RS 4n
64-bit x86 ELF kernel module
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/mpt_sas.conf\fR\fR
.ad
.sp .6
.RS 4n
Optional configuration file
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for a description of the following attributes:
.sp

.sp
.TS
box;
l | l
l | l .
```

```
124 ATTRIBUTE TYPE  ATTRIBUTE VALUE
125 _
126 Architecture    SPARC, x86
127 .TE

129 .SH SEE ALSO
130 .sp
131 .LP
132 \fBprtconf\fR(1M), \fBdriver.conf\fR(4), \fBpci\fR(4), \fBattributes\fR(5),
133 \fBscsi_abort\fR(9F), \fBscsi_device\fR(9S), \fBscsi_extended_sense\fR(9S),
134 \fBscsi_inquiry\fR(9S), \fBscsi_hba_attach_setup\fR(9F),
135 \fBscsi_ifgetcap\fR(9F), \fBscsi_ifsetcap\fR(9F), \fBscsi_pkt\fR(9S),
136 \fBscsi_reset\fR(9F), \fBscsi_sync_pkt\fR(9F), \fBscsi_transport\fR(9F),
```

```
**********************************************************
    2864 Mon Jun 16 21:18:03 2014
new/usr/src/pkg/manifests/driver-storage-mpt_sas.mf
NEX-1889 upstream
**********************************************************
```

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

#
# Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2014, Nexenta Systems, Inc. All Rights Reserved
#

#
# The default for payload-bearing actions in this package is to appear in the
# global zone only.  See the include file for greater detail, as well as
# information about overriding the defaults.
#
<include global_zone_only_component>
set name=pkg.fmri value=pkg:/driver/storage/mpt_sas@$(PKGVERS)
set name=pkg.description value="LSI MPT SAS 2.0/2.5 Controller HBA Driver"
set name=pkg.summary value="LSI MPT SAS 2.0/2.5 Controller HBA Driver"
set name=pkg.description value="LSI MPT SAS 2.0 Controller HBA Driver"
set name=pkg.summary value="LSI MPT SAS 2.0 Controller HBA Driver"
set name=info.classification \
    value=org.opensolaris.category.2008:Drivers/Storage
set name=variant.arch value=$(ARCH)
dir path=kernel group=sys
dir path=kernel/drv group=sys
dir path=kernel/drv/$(ARCH64) group=sys
dir path=usr/share/man
dir path=usr/share/man/man7d
driver name=mpt_sas class=scsi-self-identifying \
    alias=pci1000,64 \
    alias=pci1000,70 \
    alias=pci1000,72 \
    alias=pci1000,76 \
    alias=pciex1000,64 \
    alias=pciex1000,65 \
    alias=pciex1000,6e \
    alias=pciex1000,70 \
    alias=pciex1000,72 \
    alias=pciex1000,74 \
    alias=pciex1000,76 \
    alias=pciex1000,77 \
    alias=pciex1000,7e \
    alias=pciex1000,80 \
    alias=pciex1000,81 \
```



```
    alias=pciex1000,82 \
    alias=pciex1000,83 \
    alias=pciex1000,84 \
    alias=pciex1000,85 \
    alias=pciex1000,86 \
    alias=pciex1000,87 \
    alias=pciex1000,90 \
    alias=pciex1000,91 \
    alias=pciex1000,92 \
    alias=pciex1000,93 \
    alias=pciex1000,94 \
    alias=pciex1000,95 \
    alias=pciex1000,96 \
    alias=pciex1000,97
    alias=pciex1000,87
file path=kernel/drv/$(ARCH64)/mpt_sas group=sys
$(i386_ONLY)file path=kernel/drv/mpt_sas group=sys
file path=kernel/drv/mpt_sas.conf group=sys \
    original_name=SUNWmptsas:kernel/drv/mpt_sas.conf preserve=true
file path=usr/share/man/man7d/mpt_sas.7d
legacy pkg=SUNWmptsas desc="LSI MPT SAS 2.0/2.5 Controller HBA Driver" \
    name="LSI MPT SAS 2.0/2.5 Controller HBA Driver"
legacy pkg=SUNWmptsas desc="LSI MPT SAS 2.0 Controller HBA Driver" \
    name="LSI MPT SAS 2.0 Controller HBA Driver"
license cr_Sun license=cr_Sun
license lic_CDDL license=lic_CDDL
```

```
**********************************************************
    1812 Mon Jun 16 21:18:03 2014
new/usr/src/uts/common/io/scsi/adapters/mpt_sas/mpt_sas.conf
NEX-1889 upstream
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#


#
# The mpt_sas driver, as a pHCI driver, must specify the vHCI class it
# belongs to(scsi_vhci).
#
ddi-vhci-class="scsi_vhci";
#
# I/O multipathing feature (MPxIO) can be enabled or disabled using
# mpxio-disable property. Setting mpxio-disable="no" will activate
# I/O multipathing; setting mpxio-disable="yes" disables the feature.
#
# Global mpxio-disable property:
#
# To globally enable MPxIO on all LSI MPT SAS 2.0/2.5 controllers set:
# To globally enable MPxIO on all LSI MPT SAS 2.0 controllers set:
# mpxio-disable="no";
#
# To globally disable MPxIO on all LSI MPT SAS 2.0/2.5 controllers set:
# To globally disable MPxIO on all LSI MPT SAS 2.0 controllers set:
# mpxio-disable="yes";
#
# You can also enable or disable MPxIO on a per HBA basis.
# Per HBA settings override the global setting for the specified HBAs.
# To disable MPxIO on a controller whose parent is /pci@7c0/pci@0/pci@9
# and the unit-address is "0" set:
# name="mpt_sas" parent="/pci@7c0/pci@0/pci@9" unit-address="0" mpxio-disable="y
#
mpxio-disable="no";
```

```
**********************************************************
  441881 Mon Jun 16 21:18:03 2014
new/usr/src/uts/common/io/scsi/adapters/mpt_sas/mptsas.c
NEX-1889 upstream
**********************************************************
```

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

/*
 * Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
 * Copyright (c) 2014, Joyent, Inc. All rights reserved.
 * Copyright 2014 OmniTI Computer Consulting, Inc. All rights reserved.
 * Copyright (c) 2014, Tegile Systems Inc. All rights reserved.
 */

/*
 * Copyright (c) 2000 to 2010, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 * mptsas - This is a driver based on LSI Logic's MPT2.0 interface.
 *
 */
```



```
#if defined(lint) || defined(DEBUG)
#define MPTSAS_DEBUG
#endif

/*
 * standard header files.
 */
#include <sys/note.h>
#include <sys/scsi/scsi.h>
#include <sys/pci.h>
#include <sys/file.h>
#include <sys/policy.h>
#include <sys/model.h>
#include <sys/sysevent.h>
#include <sys/sysevent/eventdefs.h>
#include <sys/sysevent/dr.h>
#include <sys/sata/sata_defs.h>
#include <sys/scsi/generic/sas.h>
#include <sys/scsi/impl/scsi_sas.h>

#pragma pack(1)
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_type.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_cnfg.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_init.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_ioc.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_sas.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_tool.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_raid.h>
#pragma pack()

/*
 * private header files.
 *
 */
#include <sys/scsi/impl/scsi_reset_notify.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_var.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_ioctl.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_smhba.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_hash.h>
#include <sys/raidioctl.h>

#include <sys/fs/dv_node.h>     /* devfs_clean */

/*
 * FMA header files
 */
#include <sys/ddifm.h>
#include <sys/fm/protocol.h>
#include <sys/fm/util.h>
#include <sys/fm/io/ddi.h>

/*
 * autoconfiguration data and routines.
 */
static int mptsas_attach(dev_info_t *dip, ddi_attach_cmd_t cmd);
static int mptsas_detach(dev_info_t *devi, ddi_detach_cmd_t cmd);
static int mptsas_power(dev_info_t *dip, int component, int level);

/*
 * cb_ops function
 */
static int mptsas_ioctl(dev_t dev, int cmd, intptr_t data, int mode,
        cred_t *credp, int *rval);
#ifdef __sparc
static int mptsas_reset(dev_info_t *devi, ddi_reset_cmd_t cmd);
```

```
#else  /* __sparc */
static int mptsas_quiesce(dev_info_t *devi);
#endif  /* __sparc */

/*
 * Resource initilaization for hardware
 */
static void mptsas_setup_cmd_reg(mptsas_t *mpt);
static void mptsas_disable_bus_master(mptsas_t *mpt);
static void mptsas_hba_fini(mptsas_t *mpt);
static void mptsas_cfg_fini(mptsas_t *mptsas_blkp);
static int mptsas_hba_setup(mptsas_t *mpt);
static void mptsas_hba_teardown(mptsas_t *mpt);
static int mptsas_config_space_init(mptsas_t *mpt);
static void mptsas_config_space_fini(mptsas_t *mpt);
static void mptsas_iport_register(mptsas_t *mpt);
static int mptsas_smp_setup(mptsas_t *mpt);
static void mptsas_smp_teardown(mptsas_t *mpt);
static int mptsas_cache_create(mptsas_t *mpt);
static void mptsas_cache_destroy(mptsas_t *mpt);
static int mptsas_alloc_request_frames(mptsas_t *mpt);
static int mptsas_alloc_reply_frames(mptsas_t *mpt);
static int mptsas_alloc_free_queue(mptsas_t *mpt);
static int mptsas_alloc_post_queue(mptsas_t *mpt);
static void mptsas_alloc_reply_args(mptsas_t *mpt);
static int mptsas_alloc_extra_sgl_frame(mptsas_t *mpt, mptsas_cmd_t *cmd);
static void mptsas_free_extra_sgl_frame(mptsas_t *mpt, mptsas_cmd_t *cmd);
static int mptsas_init_chip(mptsas_t *mpt, int first_time);

/*
 * SCSA function prototypes
 */
static int mptsas_scsi_start(struct scsi_address *ap, struct scsi_pkt *pkt);
static int mptsas_scsi_reset(struct scsi_address *ap, int level);
static int mptsas_scsi_abort(struct scsi_address *ap, struct scsi_pkt *pkt);
static int mptsas_scsi_getcap(struct scsi_address *ap, char *cap, int tgtonly);
static int mptsas_scsi_setcap(struct scsi_address *ap, char *cap, int value,
    int tgtonly);
static void mptsas_scsi_dmafree(struct scsi_address *ap, struct scsi_pkt *pkt);
static struct scsi_pkt *mptsas_scsi_init_pkt(struct scsi_address *ap,
    struct scsi_pkt *pkt, struct buf *bp, int cmdlen, int statuslen,
        int tgtlen, int flags, int (*callback)(), caddr_t arg);
static void mptsas_scsi_sync_pkt(struct scsi_address *ap, struct scsi_pkt *pkt);
static void mptsas_scsi_destroy_pkt(struct scsi_address *ap,
    struct scsi_pkt *pkt);
static int mptsas_scsi_tgt_init(dev_info_t *hba_dip, dev_info_t *tgt_dip,
    scsi_hba_tran_t *hba_tran, struct scsi_device *sd);
static void mptsas_scsi_tgt_free(dev_info_t *hba_dip, dev_info_t *tgt_dip,
    scsi_hba_tran_t *hba_tran, struct scsi_device *sd);
static int mptsas_scsi_reset_notify(struct scsi_address *ap, int flag,
    void (*callback)(caddr_t), caddr_t arg);
static int mptsas_get_name(struct scsi_device *sd, char *name, int len);
static int mptsas_get_bus_addr(struct scsi_device *sd, char *name, int len);
static int mptsas_scsi_quiesce(dev_info_t *dip);
static int mptsas_scsi_unquiesce(dev_info_t *dip);
static int mptsas_bus_config(dev_info_t *pdip, uint_t flags,
    ddi_bus_config_op_t op, void *arg, dev_info_t **childp);

/*
 * SMP functions
 */
static int mptsas_smp_start(struct smp_pkt *smp_pkt);

/*
 * internal function prototypes.
 */
```



```
static void mptsas_list_add(mptsas_t *mpt);
static void mptsas_list_del(mptsas_t *mpt);

static int mptsas_quiesce_bus(mptsas_t *mpt);
static int mptsas_unquiesce_bus(mptsas_t *mpt);

static int mptsas_alloc_handshake_msg(mptsas_t *mpt, size_t alloc_size);
static void mptsas_free_handshake_msg(mptsas_t *mpt);

static void mptsas_ncmds_checkdrain(void *arg);

static int mptsas_prepare_pkt(mptsas_cmd_t *cmd);
static int mptsas_accept_pkt(mptsas_t *mpt, mptsas_cmd_t *sp);
static int mptsas_accept_txwq_and_pkt(mptsas_t *mpt, mptsas_cmd_t *sp);
static void mptsas_accept_tx_waitq(mptsas_t *mpt);

static int mptsas_do_detach(dev_info_t *dev);
static int mptsas_do_scsi_reset(mptsas_t *mpt, uint16_t devhdl);
static int mptsas_do_scsi_abort(mptsas_t *mpt, int target, int lun,
    struct scsi_pkt *pkt);
static int mptsas_scsi_capchk(char *cap, int tgtonly, int *cidxp);

static void mptsas_handle_qfull(mptsas_t *mpt, mptsas_cmd_t *cmd);
static void mptsas_handle_event(void *args);
static int mptsas_handle_event_sync(void *args);
static void mptsas_handle_dr(void *args);
static void mptsas_handle_topo_change(mptsas_topo_change_list_t *topo_node,
    dev_info_t *pdip);

static void mptsas_restart_cmd(void *);

static void mptsas_flush_hba(mptsas_t *mpt);
static void mptsas_flush_target(mptsas_t *mpt, ushort_t target, int lun,
        uint8_t tasktype);
static void mptsas_set_pkt_reason(mptsas_t *mpt, mptsas_cmd_t *cmd,
    uchar_t reason, uint_t stat);

static uint_t mptsas_intr(caddr_t arg1, caddr_t arg2);
static void mptsas_process_intr(mptsas_t *mpt,
    pMpi2ReplyDescriptorsUnion_t reply_desc_union);
static void mptsas_handle_scsi_io_success(mptsas_t *mpt,
    pMpi2ReplyDescriptorsUnion_t reply_desc);
static void mptsas_handle_address_reply(mptsas_t *mpt,
    pMpi2ReplyDescriptorsUnion_t reply_desc);
static int mptsas_wait_intr(mptsas_t *mpt, int polltime);
static void mptsas_sge_setup(mptsas_t *mpt, mptsas_cmd_t *cmd,
    uint32_t *control, pMpi2SCSIIORequest_t frame, ddi_acc_handle_t acc_hdl);

static void mptsas_watch(void *arg);
static void mptsas_watchsubr(mptsas_t *mpt);
static void mptsas_cmd_timeout(mptsas_t *mpt, mptsas_target_t *ptgt);

static void mptsas_start_passthru(mptsas_t *mpt, mptsas_cmd_t *cmd);
static int mptsas_do_passthru(mptsas_t *mpt, uint8_t *request, uint8_t *reply,
    uint8_t *data, uint32_t request_size, uint32_t reply_size,
    uint32_t data_size, uint8_t direction, uint8_t *dataout,
    uint32_t data_size, uint32_t direction, uint8_t *dataout,
    uint32_t dataout_size, short timeout, int mode);
static int mptsas_free_devhdl(mptsas_t *mpt, uint16_t devhdl);

static uint8_t mptsas_get_fw_diag_buffer_number(mptsas_t *mpt,
    uint32_t unique_id);
static void mptsas_start_diag(mptsas_t *mpt, mptsas_cmd_t *cmd);
static int mptsas_post_fw_diag_buffer(mptsas_t *mpt,
    mptsas_fw_diagnostic_buffer_t *pBuffer, uint32_t *return_code);
static int mptsas_release_fw_diag_buffer(mptsas_t *mpt,
```

```
259     mptsas_fw_diagnostic_buffer_t *pBuffer, uint32_t *return_code,
260     uint32_t diag_type);
261 static int mptsas_diag_register(mptsas_t *mpt,
262     mptsas_fw_diag_register_t *diag_register, uint32_t *return_code);
263 static int mptsas_diag_unregister(mptsas_t *mpt,
264     mptsas_fw_diag_unregister_t *diag_unregister, uint32_t *return_code);
265 static int mptsas_diag_query(mptsas_t *mpt, mptsas_fw_diag_query_t *diag_query,
266     uint32_t *return_code);
267 static int mptsas_diag_read_buffer(mptsas_t *mpt,
268     mptsas_diag_read_buffer_t *diag_read_buffer, uint8_t *ioctl_buf,
269     uint32_t *return_code, int ioctl_mode);
270 static int mptsas_diag_release(mptsas_t *mpt,
271     mptsas_fw_diag_release_t *diag_release, uint32_t *return_code);
272 static int mptsas_do_diag_action(mptsas_t *mpt, uint32_t action,
273     uint8_t *diag_action, uint32_t length, uint32_t *return_code,
274     int ioctl_mode);
275 static int mptsas_diag_action(mptsas_t *mpt, mptsas_diag_action_t *data,
276     int mode);

278 static int mptsas_pkt_alloc_extern(mptsas_t *mpt, mptsas_cmd_t *cmd,
279     int cmdlen, int tgtlen, int statuslen, int kf);
280 static void mptsas_pkt_destroy_extern(mptsas_t *mpt, mptsas_cmd_t *cmd);

282 static int mptsas_kmem_cache_constructor(void *buf, void *cdrarg, int kmflags);
283 static void mptsas_kmem_cache_destructor(void *buf, void *cdrarg);

285 static int mptsas_cache_frames_constructor(void *buf, void *cdrarg,
286     int kmflags);
287 static void mptsas_cache_frames_destructor(void *buf, void *cdrarg);

289 static void mptsas_check_scsi_io_error(mptsas_t *mpt, pMpi2SCSIIOReply_t reply,
290     mptsas_cmd_t *cmd);
291 static void mptsas_check_task_mgt(mptsas_t *mpt,
292     pMpi2SCSIManagementReply_t reply, mptsas_cmd_t *cmd);
293 static int mptsas_send_scsi_cmd(mptsas_t *mpt, struct scsi_address *ap,
294     mptsas_target_t *ptgt, uchar_t *cdb, int cdblen, struct buf *data_bp,
295     int *resid);

297 static int mptsas_alloc_active_slots(mptsas_t *mpt, int flag);
298 static void mptsas_free_active_slots(mptsas_t *mpt);
299 static int mptsas_start_cmd(mptsas_t *mpt, mptsas_cmd_t *cmd);

301 static void mptsas_restart_hba(mptsas_t *mpt);
302 static void mptsas_restart_waitq(mptsas_t *mpt);

304 static void mptsas_deliver_doneq_thread(mptsas_t *mpt);
305 static void mptsas_doneq_add(mptsas_t *mpt, mptsas_cmd_t *cmd);
306 static void mptsas_doneq_mv(mptsas_t *mpt, uint64_t t);

308 static mptsas_cmd_t *mptsas_doneq_thread_rm(mptsas_t *mpt, uint64_t t);
309 static void mptsas_doneq_empty(mptsas_t *mpt);
310 static void mptsas_doneq_thread(mptsas_doneq_thread_arg_t *arg);

312 static mptsas_cmd_t *mptsas_waitq_rm(mptsas_t *mpt);
313 static void mptsas_waitq_delete(mptsas_t *mpt, mptsas_cmd_t *cmd);
314 static mptsas_cmd_t *mptsas_tx_waitq_rm(mptsas_t *mpt);
315 static void mptsas_tx_waitq_delete(mptsas_t *mpt, mptsas_cmd_t *cmd);


318 static void mptsas_start_watch_reset_delay();
319 static void mptsas_setup_bus_reset_delay(mptsas_t *mpt);
320 static void mptsas_watch_reset_delay(void *arg);
321 static int mptsas_watch_reset_delay_subr(mptsas_t *mpt);

323 /*
324  * helper functions
```



```
325  */
326 static void mptsas_dump_cmd(mptsas_t *mpt, mptsas_cmd_t *cmd);

328 static dev_info_t *mptsas_find_child(dev_info_t *pdip, char *name);
329 static dev_info_t *mptsas_find_child_phy(dev_info_t *pdip, uint8_t phy);
330 static dev_info_t *mptsas_find_child_addr(dev_info_t *pdip, uint64_t sasaddr,
331     int lun);
332 static mdi_pathinfo_t *mptsas_find_path_addr(dev_info_t *pdip, uint64_t sasaddr,
333     int lun);
334 static mdi_pathinfo_t *mptsas_find_path_phy(dev_info_t *pdip, uint8_t phy);
335 static dev_info_t *mptsas_find_smp_child(dev_info_t *pdip, char *str_wwn);

337 static int mptsas_parse_address(char *name, uint64_t *wwid, uint8_t *phy,
338     int *lun);
339 static int mptsas_parse_smp_name(char *name, uint64_t *wwn);

341 static mptsas_target_t *mptsas_phy_to_tgt(mptsas_t *mpt,
342     mptsas_phymask_t phymask, uint8_t phy);
343 static mptsas_target_t *mptsas_wwid_to_ptgt(mptsas_t *mpt,
344     mptsas_phymask_t phymask, uint64_t wwid);
345 static mptsas_smp_t *mptsas_wwid_to_psmp(mptsas_t *mpt,
346     mptsas_phymask_t phymask, uint64_t wwid);

348 static int mptsas_inquiry(mptsas_t *mpt, mptsas_target_t *ptgt, int lun,
349     uchar_t page, unsigned char *buf, int len, int *rlen, uchar_t evpd);

351 static int mptsas_get_target_device_info(mptsas_t *mpt, uint32_t page_address,
352     uint16_t *handle, mptsas_target_t **pptgt);
353 static void mptsas_update_phymask(mptsas_t *mpt);

355 static int mptsas_send_sep(mptsas_t *mpt, mptsas_target_t *ptgt,
356     uint32_t *status, uint8_t cmd);
357 static dev_info_t *mptsas_get_dip_from_dev(dev_t dev,
358     mptsas_phymask_t *phymask);
359 static mptsas_target_t *mptsas_addr_to_ptgt(mptsas_t *mpt, char *addr,
360     mptsas_phymask_t phymask);
361 static int mptsas_flush_led_status(mptsas_t *mpt, mptsas_target_t *ptgt);


364 /*
365  * Enumeration / DR functions
366  */
367 static void mptsas_config_all(dev_info_t *pdip);
368 static int mptsas_config_one_addr(dev_info_t *pdip, uint64_t sasaddr, int lun,
369     dev_info_t **lundip);
370 static int mptsas_config_one_phy(dev_info_t *pdip, uint8_t phy, int lun,
371     dev_info_t **lundip);

373 static int mptsas_config_target(dev_info_t *pdip, mptsas_target_t *ptgt);
374 static int mptsas_offline_target(dev_info_t *pdip, char *name);

376 static int mptsas_config_raid(dev_info_t *pdip, uint16_t target,
377     dev_info_t **dip);

379 static int mptsas_config_luns(dev_info_t *pdip, mptsas_target_t *ptgt);
380 static int mptsas_probe_lun(dev_info_t *pdip, int lun,
381     dev_info_t **dip, mptsas_target_t *ptgt);

383 static int mptsas_create_lun(dev_info_t *pdip, struct scsi_inquiry *sd_inq,
384     dev_info_t **dip, mptsas_target_t *ptgt, int lun);

386 static int mptsas_create_phys_lun(dev_info_t *pdip, struct scsi_inquiry *sd,
387     char *guid, dev_info_t **dip, mptsas_target_t *ptgt, int lun);
388 static int mptsas_create_virt_lun(dev_info_t *pdip, struct scsi_inquiry *sd,
389     char *guid, dev_info_t **dip, mdi_pathinfo_t **pip, mptsas_target_t *ptgt,
390     int lun);
```

```
392 static void mptsas_offline_missed_luns(dev_info_t *pdip,
393     uint16_t *repluns, int lun_cnt, mptsas_target_t *ptgt);
394 static int mptsas_offline_lun(dev_info_t *pdip, dev_info_t *rdip,
395     mdi_pathinfo_t *rpip, uint_t flags);

397 static int mptsas_config_smp(dev_info_t *pdip, uint64_t sas_wwn,
398     dev_info_t **smp_dip);
399 static int mptsas_offline_smp(dev_info_t *pdip, mptsas_smp_t *smp_node,
400     uint_t flags);

402 static int mptsas_event_query(mptsas_t *mpt, mptsas_event_query_t *data,
403     int mode, int *rval);
404 static int mptsas_event_enable(mptsas_t *mpt, mptsas_event_enable_t *data,
405     int mode, int *rval);
406 static int mptsas_event_report(mptsas_t *mpt, mptsas_event_report_t *data,
407     int mode, int *rval);
408 static void mptsas_record_event(void *args);
409 static int mptsas_reg_access(mptsas_t *mpt, mptsas_reg_access_t *data,
410     int mode);

412 mptsas_target_t *mptsas_tgt_alloc(mptsas_t *, uint16_t, uint64_t,
413     uint32_t, mptsas_phymask_t, uint8_t);
414 static mptsas_smp_t *mptsas_smp_alloc(mptsas_t *, mptsas_smp_t *);
415 static int mptsas_online_smp(dev_info_t *pdip, mptsas_smp_t *smp_node,
416     dev_info_t **smp_dip);

418 /*
419  * Power management functions
420  */
421 static int mptsas_get_pci_cap(mptsas_t *mpt);
422 static int mptsas_init_pm(mptsas_t *mpt);

424 /*
425  * MPT MSI tunable:
426  *
427  * By default MSI is enabled on all supported platforms.
428  */
429 boolean_t mptsas_enable_msi = B_TRUE;
430 boolean_t mptsas_physical_bind_failed_page_83 = B_FALSE;

432 static int mptsas_register_intrs(mptsas_t *);
433 static void mptsas_unregister_intrs(mptsas_t *);
434 static int mptsas_add_intrs(mptsas_t *, int);
435 static void mptsas_rem_intrs(mptsas_t *);

437 /*
438  * FMA Prototypes
439  */
440 static void mptsas_fm_init(mptsas_t *mpt);
441 static void mptsas_fm_fini(mptsas_t *mpt);
442 static int mptsas_fm_error_cb(dev_info_t *, ddi_fm_error_t *, const void *);

444 extern pri_t minclsyspri, maxclsyspri;

446 /*
447  * This device is created by the SCSI pseudo nexus driver (SCSI vHCI).  It is
448  * under this device that the paths to a physical device are created when
449  * MPxIO is used.
450  */
451 extern dev_info_t       *scsi_vhci_dip;

453 /*
454  * Tunable timeout value for Inquiry VPD page 0x83
455  * By default the value is 30 seconds.
456  */
```



```
457 int mptsas_inq83_retry_timeout = 30;

459 /*
460  * This is used to allocate memory for message frame storage, not for
461  * data I/O DMA. All message frames must be stored in the first 4G of
462  * physical memory.
463  */
464 ddi_dma_attr_t mptsas_dma_attrs = {
465         DMA_ATTR_V0,    /* attribute layout version             */
466         0x0ull,         /* address low - should be 0 (longlong) */
467         0xffffffffull,  /* address high - 32-bit max range      */
468         0x00ffffffull,  /* count max - max DMA object size      */
469         4,              /* allocation alignment requirements    */
470         0x78,           /* burstsizes - binary encoded values   */
471         1,              /* minxfer - gran. of DMA engine        */
472         0x00ffffffull,  /* maxxfer - gran. of DMA engine        */
473         0xffffffffull,  /* max segment size (DMA boundary)      */
474         MPTSAS_MAX_DMA_SEGS, /* scatter/gather list length      */
475         512,            /* granularity - device transfer size   */
476         0               /* flags, set to 0                      */
477 };

479 /*
480  * This is used for data I/O DMA memory allocation. (full 64-bit DMA
481  * physical addresses are supported.)
482  */
483 ddi_dma_attr_t mptsas_dma_attrs64 = {
484         DMA_ATTR_V0,    /* attribute layout version             */
485         0x0ull,         /* address low - should be 0 (longlong) */
486         0xffffffffffffffffull,  /* address high - 64-bit max    */
487         0x00ffffffull,  /* count max - max DMA object size      */
488         4,              /* allocation alignment requirements    */
489         0x78,           /* burstsizes - binary encoded values   */
490         1,              /* minxfer - gran. of DMA engine        */
491         0x00ffffffull,  /* maxxfer - gran. of DMA engine        */
492         0xffffffffull,  /* max segment size (DMA boundary)      */
493         MPTSAS_MAX_DMA_SEGS, /* scatter/gather list length      */
494         512,            /* granularity - device transfer size   */
495         0               /* flags, set to 0 */
494         DDI_DMA_RELAXED_ORDERING        /* flags, enable relaxed ordering */
496 };
_____unchanged_portion_omitted_
561 #define TARGET_PROP     "target"
562 #define LUN_PROP        "lun"
563 #define LUN64_PROP      "lun64"
564 #define SAS_PROP        "sas-mpt"
565 #define MDI_GUID        "wwn"
566 #define NDI_GUID        "guid"
567 #define MPTSAS_DEV_GONE "mptsas_dev_gone"

569 /*
570  * Local static data
571  */
572 #if defined(MPTSAS_DEBUG)
573 uint32_t mptsas_debug_flags = 0x0;
572 uint32_t mptsas_debug_flags = 0;
574 #endif  /* defined(MPTSAS_DEBUG) */
575 uint32_t mptsas_debug_resets = 0;

577 static kmutex_t         mptsas_global_mutex;
578 static void             *mptsas_state;          /* soft state ptr */
579 static krwlock_t        mptsas_global_rwlock;

581 static kmutex_t         mptsas_log_mutex;
582 static char             mptsas_log_buf[256];
583 _NOTE(MUTEX_PROTECTS_DATA(mptsas_log_mutex, mptsas_log_buf))
```

```
585 static mptsas_t *mptsas_head, *mptsas_tail;
586 static clock_t mptsas_scsi_watchdog_tick;
587 static clock_t mptsas_tick;
588 static timeout_id_t mptsas_reset_watch;
589 static timeout_id_t mptsas_timeout_id;
590 static int mptsas_timeouts_enabled = 0;

592 /*
593  * The only software retriction on switching msg buffers to 64 bit seems to
594  * be the Auto Request Sense interface. The high 32 bits for all such
595  * requests appear to be required to sit in the same 4G segment.
596  * See initialization of SenseBufferAddressHigh in mptsas_init.c, and
597  * the use of SenseBufferLowAddress in requests. Note that there is
598  * currently a dependency on scsi_alloc_consistent_buf() adhering to
599  * this requirement.
600  * There is also a question about improved performance over PCI/PCIX
601  * if transfers are within the first 4Gb.
602  */
603 static int mptsas_use_64bit_msgaddr = 0;

605 /*
606  * warlock directives
607  */
608 _NOTE(SCHEME_PROTECTS_DATA("unique per pkt", scsi_pkt \
609         mptsas_cmd NcrTableIndirect buf scsi_cdb scsi_status))
610 _NOTE(SCHEME_PROTECTS_DATA("unique per pkt", smp_pkt))
611 _NOTE(SCHEME_PROTECTS_DATA("stable data", scsi_device scsi_address))
612 _NOTE(SCHEME_PROTECTS_DATA("No Mutex Needed", mptsas_tgt_private))
613 _NOTE(SCHEME_PROTECTS_DATA("No Mutex Needed", scsi_hba_tran::tran_tgt_private))

615 /*
616  * SM - HBA statics
617  */
618 char    *mptsas_driver_rev = MPTSAS_MOD_STRING;

620 #ifdef MPTSAS_DEBUG
621 void debug_enter(char *);
622 #endif

624 /*
625  * Notes:
626  *      - scsi_hba_init(9F) initializes SCSI HBA modules
627  *      - must call scsi_hba_fini(9F) if modload() fails
628  */
629 int
630 _init(void)
631 {
632         int status;
633         /* CONSTCOND */
634         ASSERT(NO_COMPETING_THREADS);

636         NDBG0(("_init"));

638         status = ddi_soft_state_init(&mptsas_state, MPTSAS_SIZE,
639             MPTSAS_INITIAL_SOFT_SPACE);
640         if (status != 0) {
641                 return (status);
642         }

644         if ((status = scsi_hba_init(&modlinkage)) != 0) {
645                 ddi_soft_state_fini(&mptsas_state);
646                 return (status);
647         }

649         mutex_init(&mptsas_global_mutex, NULL, MUTEX_DRIVER, NULL);
```



```
650         rw_init(&mptsas_global_rwlock, NULL, RW_DRIVER, NULL);
651         mutex_init(&mptsas_log_mutex, NULL, MUTEX_DRIVER, NULL);

653         if ((status = mod_install(&modlinkage)) != 0) {
654                 mutex_destroy(&mptsas_log_mutex);
655                 rw_destroy(&mptsas_global_rwlock);
656                 mutex_destroy(&mptsas_global_mutex);
657                 ddi_soft_state_fini(&mptsas_state);
658                 scsi_hba_fini(&modlinkage);
659         }

661         return (status);
662 }
_____unchanged_portion_omitted_

1025 /*
1026  * Notes:
1027  *      Set up all device state and allocate data structures,
1028  *      mutexes, condition variables, etc. for device operation.
1029  *      Add interrupts needed.
1030  *      Return DDI_SUCCESS if device is ready, else return DDI_FAILURE.
1031  */
1032 static int
1033 mptsas_attach(dev_info_t *dip, ddi_attach_cmd_t cmd)
1034 {
1035         mptsas_t                *mpt = NULL;
1036         int                     instance, i, j;
1037         int                     doneq_thread_num;
1038         char                    intr_added = 0;
1039         char                    map_setup = 0;
1040         char                    config_setup = 0;
1041         char                    hba_attach_setup = 0;
1042         char                    smp_attach_setup = 0;
1043         char                    mutex_init_done = 0;
1044         char                    event_taskq_create = 0;
1045         char                    dr_taskq_create = 0;
1046         char                    doneq_thread_create = 0;
1047         scsi_hba_tran_t         *hba_tran;
1048         uint_t                  mem_bar = MEM_SPACE;
1049         int                     rval = DDI_FAILURE;

1051         /* CONSTCOND */
1052         ASSERT(NO_COMPETING_THREADS);

1054         if (scsi_hba_iport_unit_address(dip)) {
1055                 return (mptsas_iport_attach(dip, cmd));
1056         }

1058         switch (cmd) {
1059         case DDI_ATTACH:
1060                 break;

1062         case DDI_RESUME:
1063                 if ((hba_tran = ddi_get_driver_private(dip)) == NULL)
1064                         return (DDI_FAILURE);

1066                 mpt = TRAN2MPT(hba_tran);

1068                 if (!mpt) {
1069                         return (DDI_FAILURE);
1070                 }

1072                 /*
1073                  * Reset hardware and softc to "no outstanding commands"
1074                  * Note that a check condition can result on first command
1075                  * to a target.
```

```
1076                 */
1077                mutex_enter(&mpt->m_mutex);

1079                /*
1080                 * raise power.
1081                 */
1082                if (mpt->m_options & MPTSAS_OPT_PM) {
1083                        mutex_exit(&mpt->m_mutex);
1084                        (void) pm_busy_component(dip, 0);
1085                        rval = pm_power_has_changed(dip, 0, PM_LEVEL_D0);
1086                        if (rval == DDI_SUCCESS) {
1087                                mutex_enter(&mpt->m_mutex);
1088                        } else {
1089                                /*
1090                                 * The pm_raise_power() call above failed,
1091                                 * and that can only occur if we were unable
1092                                 * to reset the hardware.  This is probably
1093                                 * due to unhealty hardware, and because
1094                                 * important filesystems(such as the root
1095                                 * filesystem) could be on the attached disks,
1096                                 * it would not be a good idea to continue,
1097                                 * as we won't be entirely certain we are
1098                                 * writing correct data.  So we panic() here
1099                                 * to not only prevent possible data corruption,
1100                                 * but to give developers or end users a hope
1101                                 * of identifying and correcting any problems.
1102                                 */
1103                                fm_panic("mptsas could not reset hardware "
1104                                    "during resume");
1105                        }
1106                }

1108                mpt->m_suspended = 0;

1110                /*
1111                 * Reinitialize ioc
1112                 */
1113                mpt->m_softstate |= MPTSAS_SS_MSG_UNIT_RESET;
1114                if (mptsas_init_chip(mpt, FALSE) == DDI_FAILURE) {
1115                        mutex_exit(&mpt->m_mutex);
1116                        if (mpt->m_options & MPTSAS_OPT_PM) {
1117                                (void) pm_idle_component(dip, 0);
1118                        }
1119                        fm_panic("mptsas init chip fail during resume");
1120                }
1121                /*
1122                 * mptsas_update_driver_data needs interrupts so enable them
1123                 * first.
1124                 */
1125                MPTSAS_ENABLE_INTR(mpt);
1126                mptsas_update_driver_data(mpt);

1128                /* start requests, if possible */
1129                mptsas_restart_hba(mpt);

1131                mutex_exit(&mpt->m_mutex);

1133                /*
1134                 * Restart watch thread
1135                 */
1136                mutex_enter(&mptsas_global_mutex);
1137                if (mptsas_timeout_id == 0) {
1138                        mptsas_timeout_id = timeout(mptsas_watch, NULL,
1139                            mptsas_tick);
1140                        mptsas_timeouts_enabled = 1;
1141                }
```



```
1142                mutex_exit(&mptsas_global_mutex);

1144                /* report idle status to pm framework */
1145                if (mpt->m_options & MPTSAS_OPT_PM) {
1146                        (void) pm_idle_component(dip, 0);
1147                }

1149                return (DDI_SUCCESS);

1151        default:
1152                return (DDI_FAILURE);

1154        }

1156        instance = ddi_get_instance(dip);

1158        /*
1159         * Allocate softc information.
1160         */
1161        if (ddi_soft_state_zalloc(mptsas_state, instance) != DDI_SUCCESS) {
1162                mptsas_log(NULL, CE_WARN,
1163                    "mptsas%d: cannot allocate soft state", instance);
1164                goto fail;
1165        }

1167        mpt = ddi_get_soft_state(mptsas_state, instance);

1169        if (mpt == NULL) {
1170                mptsas_log(NULL, CE_WARN,
1171                    "mptsas%d: cannot get soft state", instance);
1172                goto fail;
1173        }

1175        /* Indicate that we are 'sizeof (scsi_*(9S))' clean. */
1176        scsi_size_clean(dip);

1178        mpt->m_dip = dip;
1179        mpt->m_instance = instance;

1181        /* Make a per-instance copy of the structures */
1182        mpt->m_io_dma_attr = mptsas_dma_attrs64;
1183        if (mptsas_use_64bit_msgaddr) {
1184                mpt->m_msg_dma_attr = mptsas_dma_attrs64;
1185        } else {
1186                mpt->m_msg_dma_attr = mptsas_dma_attrs;
1187        }
1188        mpt->m_reg_acc_attr = mptsas_dev_attr;
1189        mpt->m_dev_acc_attr = mptsas_dev_attr;

1191        /*
1192         * Initialize FMA
1193         */
1194        mpt->m_fm_capabilities = ddi_getprop(DDI_DEV_T_ANY, mpt->m_dip,
1195            DDI_PROP_CANSLEEP | DDI_PROP_DONTPASS, "fm-capable",
1196            DDI_FM_EREPORT_CAPABLE | DDI_FM_ACCCHK_CAPABLE |
1197            DDI_FM_DMACHK_CAPABLE | DDI_FM_ERRCB_CAPABLE);

1199        mptsas_fm_init(mpt);

1201        if (mptsas_alloc_handshake_msg(mpt,
1202            sizeof (Mpi2SCSITaskManagementRequest_t)) == DDI_FAILURE) {
1203                mptsas_log(mpt, CE_WARN, "cannot initialize handshake msg.");
1204                goto fail;
1205        }

1207        /*
```

```
1208          * Setup configuration space
1209          */
1210         if (mptsas_config_space_init(mpt) == FALSE) {
1211                 mptsas_log(mpt, CE_WARN, "mptsas_config_space_init failed");
1212                 goto fail;
1213         }
1214         config_setup++;

1216         if (ddi_regs_map_setup(dip, mem_bar, (caddr_t *)&mpt->m_reg,
1217             0, 0, &mpt->m_reg_acc_attr, &mpt->m_datap) != DDI_SUCCESS) {
1218                 mptsas_log(mpt, CE_WARN, "map setup failed");
1219                 goto fail;
1220         }
1221         map_setup++;

1223         /*
1224          * A taskq is created for dealing with the event handler
1225          */
1226         if ((mpt->m_event_taskq = ddi_taskq_create(dip, "mptsas_event_taskq",
1227             1, TASKQ_DEFAULTPRI, 0)) == NULL) {
1228                 mptsas_log(mpt, CE_NOTE, "ddi_taskq_create failed");
1229                 goto fail;
1230         }
1231         event_taskq_create++;

1233         /*
1234          * A taskq is created for dealing with dr events
1235          */
1236         if ((mpt->m_dr_taskq = ddi_taskq_create(dip,
1237             "mptsas_dr_taskq",
1238             1, TASKQ_DEFAULTPRI, 0)) == NULL) {
1239                 mptsas_log(mpt, CE_NOTE, "ddi_taskq_create for discovery "
1240                     "failed");
1241                 goto fail;
1242         }
1243         dr_taskq_create++;

1245         mpt->m_doneq_thread_threshold = ddi_prop_get_int(DDI_DEV_T_ANY, dip,
1246             0, "mptsas_doneq_thread_threshold_prop", 10);
1247         mpt->m_doneq_length_threshold = ddi_prop_get_int(DDI_DEV_T_ANY, dip,
1248             0, "mptsas_doneq_length_threshold_prop", 8);
1249         mpt->m_doneq_thread_n = ddi_prop_get_int(DDI_DEV_T_ANY, dip,
1250             0, "mptsas_doneq_thread_n_prop", 8);

1252         if (mpt->m_doneq_thread_n) {
1253                 cv_init(&mpt->m_doneq_thread_cv, NULL, CV_DRIVER, NULL);
1254                 mutex_init(&mpt->m_doneq_mutex, NULL, MUTEX_DRIVER, NULL);

1256                 mutex_enter(&mpt->m_doneq_mutex);
1257                 mpt->m_doneq_thread_id =
1258                     kmem_zalloc(sizeof (mptsas_doneq_thread_list_t)
1259                     * mpt->m_doneq_thread_n, KM_SLEEP);

1261                 for (j = 0; j < mpt->m_doneq_thread_n; j++) {
1262                         cv_init(&mpt->m_doneq_thread_id[j].cv, NULL,
1263                             CV_DRIVER, NULL);
1264                         mutex_init(&mpt->m_doneq_thread_id[j].mutex, NULL,
1265                             MUTEX_DRIVER, NULL);
1266                         mutex_enter(&mpt->m_doneq_thread_id[j].mutex);
1267                         mpt->m_doneq_thread_id[j].flag |=
1268                             MPTSAS_DONEQ_THREAD_ACTIVE;
1269                         mpt->m_doneq_thread_id[j].arg.mpt = mpt;
1270                         mpt->m_doneq_thread_id[j].arg.t = j;
1271                         mpt->m_doneq_thread_id[j].threadp =
1272                             thread_create(NULL, 0, mptsas_doneq_thread,
1273                             &mpt->m_doneq_thread_id[j].arg,
```



```
1274                             0, &p0, TS_RUN, minclsyspri);
1275                         mpt->m_doneq_thread_id[j].donetail =
1276                             &mpt->m_doneq_thread_id[j].doneq;
1277                         mutex_exit(&mpt->m_doneq_thread_id[j].mutex);
1278                 }
1279                 mutex_exit(&mpt->m_doneq_mutex);
1280                 doneq_thread_create++;
1281         }

1283         /*
1284          * Disable hardware interrupt since we're not ready to
1285          * handle it yet.
1286          */
1287         MPTSAS_DISABLE_INTR(mpt);
1288         if (mptsas_register_intrs(mpt) == FALSE)
1289                 goto fail;
1290         intr_added++;

1292         /* Initialize mutex used in interrupt handler */
1293         mutex_init(&mpt->m_mutex, NULL, MUTEX_DRIVER,
1294             DDI_INTR_PRI(mpt->m_intr_pri));
1295         mutex_init(&mpt->m_passthru_mutex, NULL, MUTEX_DRIVER, NULL);
1296         mutex_init(&mpt->m_tx_waitq_mutex, NULL, MUTEX_DRIVER,
1297             DDI_INTR_PRI(mpt->m_intr_pri));
1298         for (i = 0; i < MPTSAS_MAX_PHYS; i++) {
1299                 mutex_init(&mpt->m_phy_info[i].smhba_info.phy_mutex,
1300                     NULL, MUTEX_DRIVER,
1301                     DDI_INTR_PRI(mpt->m_intr_pri));
1302         }

1304         cv_init(&mpt->m_cv, NULL, CV_DRIVER, NULL);
1305         cv_init(&mpt->m_passthru_cv, NULL, CV_DRIVER, NULL);
1306         cv_init(&mpt->m_fw_cv, NULL, CV_DRIVER, NULL);
1307         cv_init(&mpt->m_config_cv, NULL, CV_DRIVER, NULL);
1308         cv_init(&mpt->m_fw_diag_cv, NULL, CV_DRIVER, NULL);
1309         mutex_init_done++;

1283         /*
1284          * Disable hardware interrupt since we're not ready to
1285          * handle it yet.
1286          */
1287         MPTSAS_DISABLE_INTR(mpt);
1288         if (mptsas_register_intrs(mpt) == FALSE)
1289                 goto fail;
1290         intr_added++;

1311         mutex_enter(&mpt->m_mutex);
1312         /*
1313          * Initialize power management component
1314          */
1315         if (mpt->m_options & MPTSAS_OPT_PM) {
1316                 if (mptsas_init_pm(mpt)) {
1317                         mutex_exit(&mpt->m_mutex);
1318                         mptsas_log(mpt, CE_WARN, "mptsas pm initialization "
1319                             "failed");
1320                         goto fail;
1321                 }
1322         }

1324         /*
1325          * Initialize chip using Message Unit Reset, if allowed
1326          */
1327         mpt->m_softstate |= MPTSAS_SS_MSG_UNIT_RESET;
1328         if (mptsas_init_chip(mpt, TRUE) == DDI_FAILURE) {
1329                 mutex_exit(&mpt->m_mutex);
1330                 mptsas_log(mpt, CE_WARN, "mptsas chip initialization failed");
```

```
1331                goto fail;
1332        }

1334        /*
1335         * Fill in the phy_info structure and get the base WWID
1336         */
1337        if (mptsas_get_manufacture_page5(mpt) == DDI_FAILURE) {
1338                mptsas_log(mpt, CE_WARN,
1339                    "mptsas_get_manufacture_page5 failed!");
1340                goto fail;
1341        }

1343        if (mptsas_get_sas_io_unit_page_hndshk(mpt)) {
1344                mptsas_log(mpt, CE_WARN,
1345                    "mptsas_get_sas_io_unit_page_hndshk failed!");
1346                goto fail;
1347        }

1349        if (mptsas_get_manufacture_page0(mpt) == DDI_FAILURE) {
1350                mptsas_log(mpt, CE_WARN,
1351                    "mptsas_get_manufacture_page0 failed!");
1352                goto fail;
1353        }

1355        mutex_exit(&mpt->m_mutex);

1357        /*
1358         * Register the iport for multiple port HBA
1359         */
1360        mptsas_iport_register(mpt);

1362        /*
1363         * initialize SCSI HBA transport structure
1364         */
1365        if (mptsas_hba_setup(mpt) == FALSE)
1366                goto fail;
1367        hba_attach_setup++;

1369        if (mptsas_smp_setup(mpt) == FALSE)
1370                goto fail;
1371        smp_attach_setup++;

1373        if (mptsas_cache_create(mpt) == FALSE)
1374                goto fail;

1376        mpt->m_scsi_reset_delay = ddi_prop_get_int(DDI_DEV_T_ANY,
1377            dip, 0, "scsi-reset-delay", SCSI_DEFAULT_RESET_DELAY);
1378        if (mpt->m_scsi_reset_delay == 0) {
1379                mptsas_log(mpt, CE_NOTE,
1380                    "scsi_reset_delay of 0 is not recommended,"
1381                    " resetting to SCSI_DEFAULT_RESET_DELAY\n");
1382                mpt->m_scsi_reset_delay = SCSI_DEFAULT_RESET_DELAY;
1383        }

1385        /*
1386         * Initialize the wait and done FIFO queue
1387         */
1388        mpt->m_donetail = &mpt->m_doneq;
1389        mpt->m_waitqtail = &mpt->m_waitq;
1390        mpt->m_tx_waitqtail = &mpt->m_tx_waitq;
1391        mpt->m_tx_draining = 0;

1393        /*
1394         * ioc cmd queue initialize
1395         */
1396        mpt->m_ioc_event_cmdtail = &mpt->m_ioc_event_cmdq;
```



```
1397        mpt->m_dev_handle = 0xFFFF;

1399        MPTSAS_ENABLE_INTR(mpt);

1401        /*
1402         * enable event notification
1403         */
1404        mutex_enter(&mpt->m_mutex);
1405        if (mptsas_ioc_enable_event_notification(mpt)) {
1406                mutex_exit(&mpt->m_mutex);
1407                goto fail;
1408        }
1409        mutex_exit(&mpt->m_mutex);

1411        /*
1412         * Initialize PHY info for smhba
1413         */
1414        if (mptsas_smhba_setup(mpt)) {
1415                mptsas_log(mpt, CE_WARN, "mptsas phy initialization "
1416                    "failed");
1417                goto fail;
1418        }

1420        /* Check all dma handles allocated in attach */
1421        if ((mptsas_check_dma_handle(mpt->m_dma_req_frame_hdl)
1422            != DDI_SUCCESS) ||
1423            (mptsas_check_dma_handle(mpt->m_dma_reply_frame_hdl)
1424            != DDI_SUCCESS) ||
1425            (mptsas_check_dma_handle(mpt->m_dma_free_queue_hdl)
1426            != DDI_SUCCESS) ||
1427            (mptsas_check_dma_handle(mpt->m_dma_post_queue_hdl)
1428            != DDI_SUCCESS) ||
1429            (mptsas_check_dma_handle(mpt->m_hshk_dma_hdl)
1430            != DDI_SUCCESS)) {
1431                goto fail;
1432        }

1434        /* Check all acc handles allocated in attach */
1435        if ((mptsas_check_acc_handle(mpt->m_datap) != DDI_SUCCESS) ||
1436            (mptsas_check_acc_handle(mpt->m_acc_req_frame_hdl)
1437            != DDI_SUCCESS) ||
1438            (mptsas_check_acc_handle(mpt->m_acc_reply_frame_hdl)
1439            != DDI_SUCCESS) ||
1440            (mptsas_check_acc_handle(mpt->m_acc_free_queue_hdl)
1441            != DDI_SUCCESS) ||
1442            (mptsas_check_acc_handle(mpt->m_acc_post_queue_hdl)
1443            != DDI_SUCCESS) ||
1444            (mptsas_check_acc_handle(mpt->m_hshk_acc_hdl)
1445            != DDI_SUCCESS) ||
1446            (mptsas_check_acc_handle(mpt->m_config_handle)
1447            != DDI_SUCCESS)) {
1448                goto fail;
1449        }

1451        /*
1452         * After this point, we are not going to fail the attach.
1453         */
1454        /*
1455         * used for mptsas_watch
1456         */
1457        mptsas_list_add(mpt);

1459        mutex_enter(&mptsas_global_mutex);
1460        if (mptsas_timeouts_enabled == 0) {
1461                mptsas_scsi_watchdog_tick = ddi_prop_get_int(DDI_DEV_T_ANY,
1462                    dip, 0, "scsi-watchdog-tick", DEFAULT_WD_TICK);
```

```
1464                mptsas_tick = mptsas_scsi_watchdog_tick *
1465                    drv_usectohz((clock_t)1000000);

1467                mptsas_timeout_id = timeout(mptsas_watch, NULL, mptsas_tick);
1468                mptsas_timeouts_enabled = 1;
1469        }
1470        mutex_exit(&mptsas_global_mutex);

1472        /* Print message of HBA present */
1473        ddi_report_dev(dip);

1475        /* report idle status to pm framework */
1476        if (mpt->m_options & MPTSAS_OPT_PM) {
1477                (void) pm_idle_component(dip, 0);
1478        }

1480        return (DDI_SUCCESS);

1482 fail:
1483        mptsas_log(mpt, CE_WARN, "attach failed");
1484        mptsas_fm_ereport(mpt, DDI_FM_DEVICE_NO_RESPONSE);
1485        ddi_fm_service_impact(mpt->m_dip, DDI_SERVICE_LOST);
1486        if (mpt) {
1487                mutex_enter(&mptsas_global_mutex);

1489                if (mptsas_timeout_id && (mptsas_head == NULL)) {
1490                        timeout_id_t tid = mptsas_timeout_id;
1491                        mptsas_timeouts_enabled = 0;
1492                        mptsas_timeout_id = 0;
1493                        mutex_exit(&mptsas_global_mutex);
1494                        (void) untimeout(tid);
1495                        mutex_enter(&mptsas_global_mutex);
1496                }
1497                mutex_exit(&mptsas_global_mutex);
1498                /* deallocate in reverse order */
1499                mptsas_cache_destroy(mpt);

1501                if (smp_attach_setup) {
1502                        mptsas_smp_teardown(mpt);
1503                }
1504                if (hba_attach_setup) {
1505                        mptsas_hba_teardown(mpt);
1506                }

1508                if (mpt->m_targets)
1509                        refhash_destroy(mpt->m_targets);
1510                if (mpt->m_smp_targets)
1511                        refhash_destroy(mpt->m_smp_targets);

1513                if (mpt->m_active) {
1514                        mptsas_free_active_slots(mpt);
1515                }
1516                if (intr_added) {
1517                        mptsas_unregister_intrs(mpt);
1518                }

1520                if (doneq_thread_create) {
1521                        mutex_enter(&mpt->m_doneq_mutex);
1522                        doneq_thread_num = mpt->m_doneq_thread_n;
1523                        for (j = 0; j < mpt->m_doneq_thread_n; j++) {
1524                                mutex_enter(&mpt->m_doneq_thread_id[j].mutex);
1525                                mpt->m_doneq_thread_id[j].flag &=
1526                                    (~MPTSAS_DONEQ_THREAD_ACTIVE);
1527                                cv_signal(&mpt->m_doneq_thread_id[j].cv);
1528                                mutex_exit(&mpt->m_doneq_thread_id[j].mutex);
```



```
1529                        }
1530                        while (mpt->m_doneq_thread_n) {
1531                                cv_wait(&mpt->m_doneq_thread_cv,
1532                                    &mpt->m_doneq_mutex);
1533                        }
1534                        for (j = 0; j < doneq_thread_num; j++) {
1535                                cv_destroy(&mpt->m_doneq_thread_id[j].cv);
1536                                mutex_destroy(&mpt->m_doneq_thread_id[j].mutex);
1537                        }
1538                        kmem_free(mpt->m_doneq_thread_id,
1539                            sizeof (mptsas_doneq_thread_list_t)
1540                            * doneq_thread_num);
1541                        mutex_exit(&mpt->m_doneq_mutex);
1542                        cv_destroy(&mpt->m_doneq_thread_cv);
1543                        mutex_destroy(&mpt->m_doneq_mutex);
1544                }
1545                if (event_taskq_create) {
1546                        ddi_taskq_destroy(mpt->m_event_taskq);
1547                }
1548                if (dr_taskq_create) {
1549                        ddi_taskq_destroy(mpt->m_dr_taskq);
1550                }
1551                if (mutex_init_done) {
1552                        mutex_destroy(&mpt->m_tx_waitq_mutex);
1553                        mutex_destroy(&mpt->m_passthru_mutex);
1554                        mutex_destroy(&mpt->m_mutex);
1555                        for (i = 0; i < MPTSAS_MAX_PHYS; i++) {
1556                                mutex_destroy(
1557                                    &mpt->m_phy_info[i].smhba_info.phy_mutex);
1558                        }
1559                        cv_destroy(&mpt->m_cv);
1560                        cv_destroy(&mpt->m_passthru_cv);
1561                        cv_destroy(&mpt->m_fw_cv);
1562                        cv_destroy(&mpt->m_config_cv);
1563                        cv_destroy(&mpt->m_fw_diag_cv);
1564                }

1566                if (map_setup) {
1567                        mptsas_cfg_fini(mpt);
1568                }
1569                if (config_setup) {
1570                        mptsas_config_space_fini(mpt);
1571                }
1572                mptsas_free_handshake_msg(mpt);
1573                mptsas_hba_fini(mpt);

1575                mptsas_fm_fini(mpt);
1576                ddi_soft_state_free(mptsas_state, instance);
1577                ddi_prop_remove_all(dip);
1578        }
1579        return (DDI_FAILURE);
1580 }
_____unchanged_portion_omitted_

2233 static int
2234 mptsas_cache_create(mptsas_t *mpt)
2235 {
2236        int instance = mpt->m_instance;
2237        char buf[64];

2239        /*
2240         * create kmem cache for packets
2241         */
2242        (void) sprintf(buf, "mptsas%d_cache", instance);
2243        mpt->m_kmem_cache = kmem_cache_create(buf,
2244            sizeof (struct mptsas_cmd) + scsi_pkt_size(), 16,
```

```
2225             sizeof (struct mptsas_cmd) + scsi_pkt_size(), 8,
2245             mptsas_kmem_cache_constructor, mptsas_kmem_cache_destructor,
2246             NULL, (void *)mpt, NULL, 0);

2248         if (mpt->m_kmem_cache == NULL) {
2249                 mptsas_log(mpt, CE_WARN, "creating kmem cache failed");
2250                 return (FALSE);
2251         }

2253         /*
2254          * create kmem cache for extra SGL frames if SGL cannot
2255          * be accomodated into main request frame.
2256          */
2257         (void) sprintf(buf, "mptsas%d_cache_frames", instance);
2258         mpt->m_cache_frames = kmem_cache_create(buf,
2259             sizeof (mptsas_cache_frames_t), 16,
2240             sizeof (mptsas_cache_frames_t), 8,
2260             mptsas_cache_frames_constructor, mptsas_cache_frames_destructor,
2261             NULL, (void *)mpt, NULL, 0);

2263         if (mpt->m_cache_frames == NULL) {
2264                 mptsas_log(mpt, CE_WARN, "creating cache for frames failed");
2265                 return (FALSE);
2266         }

2268         return (TRUE);
2269 }
______unchanged_portion_omitted_

3942 static int
3943 mptsas_cache_frames_constructor(void *buf, void *cdrarg, int kmflags)
3944 {
3945         mptsas_cache_frames_t   *p = buf;
3946         mptsas_t                *mpt = cdrarg;
3947         ddi_dma_attr_t          frame_dma_attr;
3948         size_t                  mem_size, alloc_len;
3949         ddi_dma_cookie_t        cookie;
3950         uint_t                  ncookie;
3951         int (*callback)(caddr_t) = (kmflags == KM_SLEEP)
3952             ? DDI_DMA_SLEEP: DDI_DMA_DONTWAIT;

3954         frame_dma_attr = mpt->m_msg_dma_attr;
3955         frame_dma_attr.dma_attr_align = 0x10;
3956         frame_dma_attr.dma_attr_sgllen = 1;

3958         if (ddi_dma_alloc_handle(mpt->m_dip, &frame_dma_attr, callback, NULL,
3959             &p->m_dma_hdl) != DDI_SUCCESS) {
3960                 mptsas_log(mpt, CE_WARN, "Unable to allocate dma handle for"
3961                     " extra SGL.");
3962                 return (DDI_FAILURE);
3963         }

3965         mem_size = (mpt->m_max_request_frames - 1) * mpt->m_req_frame_size;

3967         if (ddi_dma_mem_alloc(p->m_dma_hdl, mem_size, &mpt->m_dev_acc_attr,
3968             DDI_DMA_CONSISTENT, callback, NULL, (caddr_t *)&p->m_frames_addr,
3969             &alloc_len, &p->m_acc_hdl) != DDI_SUCCESS) {
3970                 ddi_dma_free_handle(&p->m_dma_hdl);
3971                 p->m_dma_hdl = NULL;
3972                 mptsas_log(mpt, CE_WARN, "Unable to allocate dma memory for"
3973                     " extra SGL.");
3974                 return (DDI_FAILURE);
3975         }

3977         if (ddi_dma_addr_bind_handle(p->m_dma_hdl, NULL, p->m_frames_addr,
3978             alloc_len, DDI_DMA_RDWR | DDI_DMA_CONSISTENT, callback, NULL,
```



```
3979             &cookie, &ncookie) != DDI_DMA_MAPPED) {
3980                 (void) ddi_dma_mem_free(&p->m_acc_hdl);
3981                 ddi_dma_free_handle(&p->m_dma_hdl);
3982                 p->m_dma_hdl = NULL;
3983                 mptsas_log(mpt, CE_WARN, "Unable to bind DMA resources for"
3984                     " extra SGL");
3985                 return (DDI_FAILURE);
3986         }

3988         /*
3989          * Store the SGL memory address.  This chip uses this
3990          * address to dma to and from the driver.  The second
3991          * address is the address mpt uses to fill in the SGL.
3992          */
3993         p->m_phys_addr = cookie.dmac_laddress;
3974         p->m_phys_addr = cookie.dmac_address;

3995         return (DDI_SUCCESS);
3996 }
______unchanged_portion_omitted_

4199 static void
4200 mptsas_sge_mainframe(mptsas_cmd_t *cmd, pMpi2SCSIIORequest_t frame,
4201                 ddi_acc_handle_t acc_hdl, uint_t cookiec,
4202                 uint32_t end_flags)
4181 mptsas_sge_setup(mptsas_t *mpt, mptsas_cmd_t *cmd, uint32_t *control,
4182         pMpi2SCSIIORequest_t frame, ddi_acc_handle_t acc_hdl)
4203 {
4204         pMpi2SGESimple64_t      sge;
4184         uint_t                  cookiec;
4205         mptti_t                 *dmap;
4206         uint32_t                flags;
4187         pMpi2SGESimple64_t      sge;
4188         pMpi2SGEChain64_t       sgechain;
4189         ASSERT(cmd->cmd_flags & CFLAG_DMAVALID);

4208         dmap = cmd->cmd_sg;
4191         /*
4192          * Save the number of entries in the DMA
4193          * Scatter/Gather list
4194          */
4195         cookiec = cmd->cmd_cookiec;

4197         NDBG1(("mptsas_sge_setup: cookiec=%d", cookiec));

4199         /*
4200          * Set read/write bit in control.
4201          */
4202         if (cmd->cmd_flags & CFLAG_DMASEND) {
4203                 *control |= MPI2_SCSIIO_CONTROL_WRITE;
4204         } else {
4205                 *control |= MPI2_SCSIIO_CONTROL_READ;
4206         }

4208         ddi_put32(acc_hdl, &frame->DataLength, cmd->cmd_dmacount);

4210         /*
4211          * We have 2 cases here.  First where we can fit all the
4212          * SG elements into the main frame, and the case
4213          * where we can't.
4214          * If we have more cookies than we can attach to a frame
4215          * we will need to use a chain element to point
4216          * a location of memory where the rest of the S/G
4217          * elements reside.
4218          */
4219         if (cookiec <= MPTSAS_MAX_FRAME_SGES64(mpt)) {
```

```
4220                dmap = cmd->cmd_sg;
4210        sge = (pMpi2SGESimple64_t)(&frame->SGL);
4211        while (cookiec--) {
4212                ddi_put32(acc_hdl, &sge->Address.Low,
4213                    dmap->addr.address64.Low);
4214                ddi_put32(acc_hdl, &sge->Address.High,
4215                    dmap->addr.address64.High);
4216                ddi_put32(acc_hdl, &sge->FlagsLength, dmap->count);
4223                        ddi_put32(acc_hdl,
4224                            &sge->Address.Low, dmap->addr.address64.Low);
4225                        ddi_put32(acc_hdl,
4226                            &sge->Address.High, dmap->addr.address64.High);
4227                        ddi_put32(acc_hdl, &sge->FlagsLength,
4228                            dmap->count);
4217                flags = ddi_get32(acc_hdl, &sge->FlagsLength);
4218                flags |= ((uint32_t)
4219                    (MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
4220                    MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
4221                    MPI2_SGE_FLAGS_64_BIT_ADDRESSING) <<
4222                    MPI2_SGE_FLAGS_SHIFT);

4224                /*
4225                 * If this is the last cookie, we set the flags
4226                 * to indicate so
4227                 */
4228                if (cookiec == 0) {
4229                        flags |= end_flags;
4241                                flags |=
4242                                    ((uint32_t)(MPI2_SGE_FLAGS_LAST_ELEMENT
4243                                    | MPI2_SGE_FLAGS_END_OF_BUFFER
4244                                    | MPI2_SGE_FLAGS_END_OF_LIST) <<
4245                                    MPI2_SGE_FLAGS_SHIFT);
4230                }
4231                if (cmd->cmd_flags & CFLAG_DMASEND) {
4232                        flags |= (MPI2_SGE_FLAGS_HOST_TO_IOC <<
4233                            MPI2_SGE_FLAGS_SHIFT);
4234                } else {
4235                        flags |= (MPI2_SGE_FLAGS_IOC_TO_HOST <<
4236                            MPI2_SGE_FLAGS_SHIFT);
4237                }
4238                ddi_put32(acc_hdl, &sge->FlagsLength, flags);
4239                dmap++;
4240                sge++;
4241        }
4242 }

4244 static void
4245 mptsas_sge_chain(mptsas_t *mpt, mptsas_cmd_t *cmd,
4246     pMpi2SCSIIORequest_t frame, ddi_acc_handle_t acc_hdl)
4247 {
4248        pMpi2SGESimple64_t      sge;
4249        pMpi2SGEChain64_t       sgechain;
4250        uint64_t                nframe_phys_addr;
4251        uint_t                  cookiec;
4252        mptti_t                 *dmap;
4253        uint32_t                flags;
4254        int                     i, j, k, l, frames, sgemax;
4255        int                     temp, maxframe_sges;
4256        uint8_t                 chainflags;
4257        uint16_t                chainlength;
4258        mptsas_cache_frames_t   *p;

4260        cookiec = cmd->cmd_cookiec;

4258        } else {
4262        /*
```



```
4263         * Hereby we start to deal with multiple frames.
4264         * The process is as follows:
4265         * 1. Determine how many frames are needed for SGL element
4266         *    storage; Note that all frames are stored in contiguous
4267         *    memory space and in 64-bit DMA mode each element is
4268         *    3 double-words (12 bytes) long.
4269         * 2. Fill up the main frame. We need to do this separately
4270         *    since it contains the SCSI IO request header and needs
4271         *    dedicated processing. Note that the last 4 double-words
4272         *    of the SCSI IO header is for SGL element storage
4273         *    (MPI2_SGE_IO_UNION).
4274         * 3. Fill the chain element in the main frame, so the DMA
4275         *    engine can use the following frames.
4276         * 4. Enter a loop to fill the remaining frames. Note that the
4277         *    last frame contains no chain element.  The remaining
4278         *    frames go into the mpt SGL buffer allocated on the fly,
4279         *    not immediately following the main message frame, as in
4280         *    Gen1.
4281         * Some restrictions:
4282         * 1. For 64-bit DMA, the simple element and chain element
4283         *    are both of 3 double-words (12 bytes) in size, even
4284         *    though all frames are stored in the first 4G of mem
4285         *    range and the higher 32-bits of the address are always 0.
4286         * 2. On some controllers (like the 1064/1068), a frame can
4287         *    hold SGL elements with the last 1 or 2 double-words
4288         *    (4 or 8 bytes) un-used. On these controllers, we should
4289         *    recognize that there's not enough room for another SGL
4290         *    element and move the sge pointer to the next frame.
4291         */
4289                int             i, j, k, l, frames, sgemax;
4290                int             temp;
4291                uint8_t         chainflags;
4292                uint16_t        chainlength;
4293                mptsas_cache_frames_t *p;

4293        /*
4294         * Sgemax is the number of SGE's that will fit
4295         * each extra frame and frames is total
4296         * number of frames we'll need.  1 sge entry per
4297         * frame is reseverd for the chain element thus the -1 below.
4298         */
4299        sgemax = ((mpt->m_req_frame_size / sizeof (MPI2_SGE_SIMPLE64)) - 1);
4300        maxframe_sges = MPTSAS_MAX_FRAME_SGES64(mpt);
4301        temp = (cookiec - (maxframe_sges - 1)) / sgemax;
4301                sgemax = ((mpt->m_req_frame_size / sizeof (MPI2_SGE_SIMPLE64))
4302                    - 1);
4303                temp = (cookiec - (MPTSAS_MAX_FRAME_SGES64(mpt) - 1)) / sgemax;

4303        /*
4304         * A little check to see if we need to round up the number
4305         * of frames we need
4306         */
4307        if ((cookiec - (maxframe_sges - 1)) - (temp * sgemax) > 1) {
4309                if ((cookiec - (MPTSAS_MAX_FRAME_SGES64(mpt) - 1)) - (temp *
4310                    sgemax) > 1) {
4308                frames = (temp + 1);
4309        } else {
4310                frames = temp;
4311        }
4312        dmap = cmd->cmd_sg;
4313        sge = (pMpi2SGESimple64_t)(&frame->SGL);

4315        /*
4316         * First fill in the main frame
4317         */
4318        j = maxframe_sges - 1;
```

```
4319         mptsas_sge_mainframe(cmd, frame, acc_hdl, j,
4320             ((uint32_t)(MPI2_SGE_FLAGS_LAST_ELEMENT) <<
4321             MPI2_SGE_FLAGS_SHIFT));
4322         dmap += j;
4323         sge += j;
4324         j++;
4321                 for (j = 1; j < MPTSAS_MAX_FRAME_SGES64(mpt); j++) {
4322                         ddi_put32(acc_hdl, &sge->Address.Low,
4323                             dmap->addr.address64.Low);
4324                         ddi_put32(acc_hdl, &sge->Address.High,
4325                             dmap->addr.address64.High);
4326                         ddi_put32(acc_hdl, &sge->FlagsLength, dmap->count);
4327                         flags = ddi_get32(acc_hdl, &sge->FlagsLength);
4328                         flags |= ((uint32_t)(MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
4329                             MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
4330                             MPI2_SGE_FLAGS_64_BIT_ADDRESSING) <<
4331                             MPI2_SGE_FLAGS_SHIFT);

4326         /*
4334                          * If this is the last SGE of this frame
4335                          * we set the end of list flag
4336                          */
4337                         if (j == (MPTSAS_MAX_FRAME_SGES64(mpt) - 1)) {
4338                                 flags |= ((uint32_t)
4339                                     (MPI2_SGE_FLAGS_LAST_ELEMENT) <<
4340                                     MPI2_SGE_FLAGS_SHIFT);
4341                         }
4342                         if (cmd->cmd_flags & CFLAG_DMASEND) {
4343                                 flags |=
4344                                     (MPI2_SGE_FLAGS_HOST_TO_IOC <<
4345                                     MPI2_SGE_FLAGS_SHIFT);
4346                         } else {
4347                                 flags |=
4348                                     (MPI2_SGE_FLAGS_IOC_TO_HOST <<
4349                                     MPI2_SGE_FLAGS_SHIFT);
4350                         }
4351                         ddi_put32(acc_hdl, &sge->FlagsLength, flags);
4352                         dmap++;
4353                         sge++;
4354                 }

4356                 /*
4327          * Fill in the chain element in the main frame.
4328          * About calculation on ChainOffset:
4329          * 1. Struct msg_scsi_io_request has 4 double-words (16 bytes)
4330          *    in the end reserved for SGL element storage
4331          *    (MPI2_SGE_IO_UNION); we should count it in our
4332          *    calculation.  See its definition in the header file.
4333          * 2. Constant j is the counter of the current SGL element
4334          *    that will be processed, and (j - 1) is the number of
4335          *    SGL elements that have been processed (stored in the
4336          *    main frame).
4337          * 3. ChainOffset value should be in units of double-words (4
4338          *    bytes) so the last value should be divided by 4.
4339          */
4340         ddi_put8(acc_hdl, &frame->ChainOffset,
4341             (sizeof (MPI2_SCSI_IO_REQUEST) -
4342             sizeof (MPI2_SGE_IO_UNION) +
4343             (j - 1) * sizeof (MPI2_SGE_SIMPLE64)) >> 2);
4344         sgechain = (pMpi2SGEChain64_t)sge;
4345         chainflags = (MPI2_SGE_FLAGS_CHAIN_ELEMENT |
4346             MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
4347             MPI2_SGE_FLAGS_64_BIT_ADDRESSING);
4348         ddi_put8(acc_hdl, &sgechain->Flags, chainflags);

4350         /*
```



```
4351          * The size of the next frame is the accurate size of space
4352          * (in bytes) used to store the SGL elements. j is the counter
4353          * of SGL elements. (j - 1) is the number of SGL elements that
4354          * have been processed (stored in frames).
4355          */
4356         if (frames >= 2) {
4357                 chainlength = mpt->m_req_frame_size /
4358                     sizeof (MPI2_SGE_SIMPLE64) *
4359                     sizeof (MPI2_SGE_SIMPLE64);
4360         } else {
4361                 chainlength = ((cookiec - (j - 1)) *
4362                     sizeof (MPI2_SGE_SIMPLE64));
4363         }

4365         p = cmd->cmd_extra_frames;

4367         ddi_put16(acc_hdl, &sgechain->Length, chainlength);
4368         ddi_put32(acc_hdl, &sgechain->Address.Low,
4369             (p->m_phys_addr&0xffffffffull));
4370         ddi_put32(acc_hdl, &sgechain->Address.High, p->m_phys_addr>>32);
4399                 p->m_phys_addr);
4400                 /* SGL is allocated in the first 4G mem range */
4401                 ddi_put32(acc_hdl, &sgechain->Address.High, 0);

4372         /*
4373          * If there are more than 2 frames left we have to
4374          * fill in the next chain offset to the location of
4375          * the chain element in the next frame.
4376          * sgemax is the number of simple elements in an extra
4377          * frame. Note that the value NextChainOffset should be
4378          * in double-words (4 bytes).
4379          */
4380         if (frames >= 2) {
4381                 ddi_put8(acc_hdl, &sgechain->NextChainOffset,
4382                     (sgemax * sizeof (MPI2_SGE_SIMPLE64)) >> 2);
4383         } else {
4384                 ddi_put8(acc_hdl, &sgechain->NextChainOffset, 0);
4385         }

4387         /*
4388          * Jump to next frame;
4389          * Starting here, chain buffers go into the per command SGL.
4390          * This buffer is allocated when chain buffers are needed.
4391          */
4392         sge = (pMpi2SGESimple64_t)p->m_frames_addr;
4393         i = cookiec;

4395         /*
4396          * Start filling in frames with SGE's.  If we
4397          * reach the end of frame and still have SGE's
4398          * to fill we need to add a chain element and
4399          * use another frame.  j will be our counter
4400          * for what cookie we are at and i will be
4401          * the total cookiec. k is the current frame
4402          */
4403         for (k = 1; k <= frames; k++) {
4404                 for (l = 1; (l <= (sgemax + 1)) && (j <= i); j++, l++) {

4406                         /*
4407                          * If we have reached the end of frame
4408                          * and we have more SGE's to fill in
4409                          * we have to fill the final entry
4410                          * with a chain element and then
4411                          * continue to the next frame
4412                          */
4413                         if ((l == (sgemax + 1)) && (k != frames)) {
```

```
4414                            sgechain = (pMpi2SGEChain64_t)sge;
4415                            j--;
4416                            chainflags = (
4417                                MPI2_SGE_FLAGS_CHAIN_ELEMENT |
4418                                MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
4419                                MPI2_SGE_FLAGS_64_BIT_ADDRESSING);
4420                            ddi_put8(p->m_acc_hdl,
4421                                &sgechain->Flags, chainflags);
4422                            /*
4423                             * k is the frame counter and (k + 1)
4424                             * is the number of the next frame.
4425                             * Note that frames are in contiguous
4426                             * memory space.
4427                             */
4428                            nframe_phys_addr = p->m_phys_addr +
4429                                (mpt->m_req_frame_size * k);
4430                            ddi_put32(p->m_acc_hdl,
4431                                &sgechain->Address.Low,
4432                                nframe_phys_addr&0xffffffffull);
4461                                    (p->m_phys_addr +
4462                                    (mpt->m_req_frame_size * k)));
4433                            ddi_put32(p->m_acc_hdl,
4434                                &sgechain->Address.High,
4435                                nframe_phys_addr>>32);
4464                                    &sgechain->Address.High, 0);

4437                            /*
4438                             * If there are more than 2 frames left
4439                             * we have to next chain offset to
4440                             * the location of the chain element
4441                             * in the next frame and fill in the
4442                             * length of the next chain
4443                             */
4444                            if ((frames - k) >= 2) {
4445                                  ddi_put8(p->m_acc_hdl,
4446                                      &sgechain->NextChainOffset,
4447                                      (sgemax *
4448                                      sizeof (MPI2_SGE_SIMPLE64))
4449                                      >> 2);
4450                                  ddi_put16(p->m_acc_hdl,
4451                                      &sgechain->Length,
4452                                      mpt->m_req_frame_size /
4453                                      sizeof (MPI2_SGE_SIMPLE64) *
4454                                      sizeof (MPI2_SGE_SIMPLE64));
4455                            } else {
4456                                  /*
4457                                   * This is the last frame. Set
4458                                   * the NextChainOffset to 0 and
4459                                   * Length is the total size of
4460                                   * all remaining simple elements
4461                                   */
4462                                  ddi_put8(p->m_acc_hdl,
4463                                      &sgechain->NextChainOffset,
4464                                      0);
4465                                  ddi_put16(p->m_acc_hdl,
4466                                      &sgechain->Length,
4467                                      (cookiec - j) *
4468                                      sizeof (MPI2_SGE_SIMPLE64));
4469                            }

4471                            /* Jump to the next frame */
4472                            sge = (pMpi2SGESimple64_t)
4473                                ((char *)p->m_frames_addr +
4474                                (int)mpt->m_req_frame_size * k);

4476                            continue;
```



```
4477                            }

4479                            ddi_put32(p->m_acc_hdl,
4480                                &sge->Address.Low,
4481                                dmap->addr.address64.Low);
4482                            ddi_put32(p->m_acc_hdl,
4483                                &sge->Address.High,
4484                                dmap->addr.address64.High);
4485                            ddi_put32(p->m_acc_hdl,
4486                                &sge->FlagsLength, dmap->count);
4487                            flags = ddi_get32(p->m_acc_hdl,
4488                                &sge->FlagsLength);
4489                            flags |= ((uint32_t)(
4490                                MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
4491                                MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
4492                                MPI2_SGE_FLAGS_64_BIT_ADDRESSING) <<
4493                                MPI2_SGE_FLAGS_SHIFT);

4495                            /*
4496                             * If we are at the end of the frame and
4497                             * there is another frame to fill in
4498                             * we set the last simple element as last
4499                             * element
4500                             */
4501                            if ((l == sgemax) && (k != frames)) {
4502                                  flags |= ((uint32_t)
4503                                      (MPI2_SGE_FLAGS_LAST_ELEMENT) <<
4504                                      MPI2_SGE_FLAGS_SHIFT);
4505                            }

4507                            /*
4508                             * If this is the final cookie we
4509                             * indicate it by setting the flags
4510                             */
4511                            if (j == i) {
4512                                  flags |= ((uint32_t)
4513                                      (MPI2_SGE_FLAGS_LAST_ELEMENT |
4514                                      MPI2_SGE_FLAGS_END_OF_BUFFER |
4515                                      MPI2_SGE_FLAGS_END_OF_LIST) <<
4516                                      MPI2_SGE_FLAGS_SHIFT);
4517                            }
4518                            if (cmd->cmd_flags & CFLAG_DMASEND) {
4519                                  flags |=
4520                                      (MPI2_SGE_FLAGS_HOST_TO_IOC <<
4521                                      MPI2_SGE_FLAGS_SHIFT);
4522                            } else {
4523                                  flags |=
4524                                      (MPI2_SGE_FLAGS_IOC_TO_HOST <<
4525                                      MPI2_SGE_FLAGS_SHIFT);
4526                            }
4527                            ddi_put32(p->m_acc_hdl,
4528                                &sge->FlagsLength, flags);
4529                            dmap++;
4530                            sge++;
4531                      }
4532                }

4534                /*
4535                 * Sync DMA with the chain buffers that were just created
4536                 */
4537                (void) ddi_dma_sync(p->m_dma_hdl, 0, 0, DDI_DMA_SYNC_FORDEV);
4538        }

4540        static void
4541        mptsas_ieee_sge_mainframe(mptsas_cmd_t *cmd, pMpi2SCSIIORequest_t frame,
4542            ddi_acc_handle_t acc_hdl, uint_t cookiec,
```

```
4543     uint8_t end_flag)
4544 {
4545         pMpi2IeeeSgeSimple64_t  ieeesge;
4546         mptti_t                 *dmap;
4547         uint8_t                 flags;

4549         dmap = cmd->cmd_sg;

4551         NDBG1(("mptsas_ieee_sge_mainframe: cookiec=%d, %s", cookiec,
4552             cmd->cmd_flags & CFLAG_DMASEND?"Out":"In"));

4554         ieeesge = (pMpi2IeeeSgeSimple64_t)(&frame->SGL);
4555         while (cookiec--) {
4556                 ddi_put32(acc_hdl, &ieeesge->Address.Low,
4557                     dmap->addr.address64.Low);
4558                 ddi_put32(acc_hdl, &ieeesge->Address.High,
4559                     dmap->addr.address64.High);
4560                 ddi_put32(acc_hdl, &ieeesge->Length, dmap->count);
4561                 NDBG1(("mptsas_ieee_sge_mainframe: len=%d", dmap->count));
4562                 flags = (MPI2_IEEE_SGE_FLAGS_SIMPLE_ELEMENT |
4563                     MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR);

4565                 /*
4566                  * If this is the last cookie, we set the flags
4567                  * to indicate so
4568                  */
4569                 if (cookiec == 0) {
4570                         flags |= end_flag;
4571                 }

4573                 /*
4574                  * XXX: Hmmm, what about the direction based on
4575                  * cmd->cmd_flags & CFLAG_DMASEND?
4576                  */
4577                 ddi_put8(acc_hdl, &ieeesge->Flags, flags);
4578                 dmap++;
4579                 ieeesge++;
4580         }
4581 }

4583 static void
4584 mptsas_ieee_sge_chain(mptsas_t *mpt, mptsas_cmd_t *cmd,
4585     pMpi2SCSIIORequest_t frame, ddi_acc_handle_t acc_hdl)
4586 {
4587         pMpi2IeeeSgeSimple64_t  ieeesge;
4588         pMpi25IeeeSgeChain64_t  ieeesgechain;
4589         uint64_t                nframe_phys_addr;
4590         uint_t                  cookiec;
4591         mptti_t                 *dmap;
4592         uint8_t                 flags;
4593         int                     i, j, k, l, frames, sgemax;
4594         int                     temp, maxframe_sges;
4595         uint8_t                 chainflags;
4596         uint32_t                chainlength;
4597         mptsas_cache_frames_t   *p;

4599         cookiec = cmd->cmd_cookiec;

4601         NDBG1(("mptsas_ieee_sge_chain: cookiec=%d", cookiec));

4603         /*
4604          * Hereby we start to deal with multiple frames.
4605          * The process is as follows:
4606          * 1. Determine how many frames are needed for SGL element
4607          *    storage; Note that all frames are stored in contiguous
4608          *    memory space and in 64-bit DMA mode each element is
```



```
4609          *    4 double-words (16 bytes) long.
4610          * 2. Fill up the main frame. We need to do this separately
4611          *    since it contains the SCSI IO request header and needs
4612          *    dedicated processing. Note that the last 4 double-words
4613          *    of the SCSI IO header is for SGL element storage
4614          *    (MPI2_SGE_IO_UNION).
4615          * 3. Fill the chain element in the main frame, so the DMA
4616          *    engine can use the following frames.
4617          * 4. Enter a loop to fill the remaining frames. Note that the
4618          *    last frame contains no chain element.  The remaining
4619          *    frames go into the mpt SGL buffer allocated on the fly,
4620          *    not immediately following the main message frame, as in
4621          *    Gen1.
4622          * Some restrictions:
4623          * 1. For 64-bit DMA, the simple element and chain element
4624          *    are both of 4 double-words (16 bytes) in size, even
4625          *    though all frames are stored in the first 4G of mem
4626          *    range and the higher 32-bits of the address are always 0.
4627          * 2. On some controllers (like the 1064/1068), a frame can
4628          *    hold SGL elements with the last 1 or 2 double-words
4629          *    (4 or 8 bytes) un-used. On these controllers, we should
4630          *    recognize that there's not enough room for another SGL
4631          *    element and move the sge pointer to the next frame.
4632          */

4634         /*
4635          * Sgemax is the number of SGE's that will fit
4636          * each extra frame and frames is total
4637          * number of frames we'll need.  1 sge entry per
4638          * frame is reseverd for the chain element thus the -1 below.
4639          */
4640         sgemax = ((mpt->m_req_frame_size / sizeof (MPI2_IEEE_SGE_SIMPLE64))
4641             - 1);
4642         maxframe_sges = MPTSAS_MAX_FRAME_SGES64(mpt);
4643         temp = (cookiec - (maxframe_sges - 1)) / sgemax;

4645         /*
4646          * A little check to see if we need to round up the number
4647          * of frames we need
4648          */
4649         if ((cookiec - (maxframe_sges - 1)) - (temp * sgemax) > 1) {
4650                 frames = (temp + 1);
4651         } else {
4652                 frames = temp;
4653         }
4654         NDBG1(("mptsas_ieee_sge_chain: temp=%d, frames=%d", temp, frames));
4655         dmap = cmd->cmd_sg;
4656         ieeesge = (pMpi2IeeeSgeSimple64_t)(&frame->SGL);

4658         /*
4659          * First fill in the main frame
4660          */
4661         j = maxframe_sges - 1;
4662         mptsas_ieee_sge_mainframe(cmd, frame, acc_hdl, j, 0);
4663         dmap += j;
4664         ieeesge += j;
4665         j++;

4667         /*
4668          * Fill in the chain element in the main frame.
4669          * About calculation on ChainOffset:
4670          * 1. Struct msg_scsi_io_request has 4 double-words (16 bytes)
4671          *    in the end reserved for SGL element storage
4672          *    (MPI2_SGE_IO_UNION); we should count it in our
4673          *    calculation.  See its definition in the header file.
4674          * 2. Constant j is the counter of the current SGL element
```

```
4675          *    that will be processed, and (j - 1) is the number of
4676          *    SGL elements that have been processed (stored in the
4677          *    main frame).
4678          * 3. ChainOffset value should be in units of quad-words (16
4679          *    bytes) so the last value should be divided by 16.
4680          */
4681         ddi_put8(acc_hdl, &frame->ChainOffset,
4682             (sizeof (MPI2_SCSI_IO_REQUEST) -
4683             sizeof (MPI2_SGE_IO_UNION) +
4684             (j - 1) * sizeof (MPI2_IEEE_SGE_SIMPLE64)) >> 4);
4685         ieeesgechain = (pMpi25IeeeSgeChain64_t)ieeesge;
4686         chainflags = (MPI2_IEEE_SGE_FLAGS_CHAIN_ELEMENT |
4687             MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR);
4688         ddi_put8(acc_hdl, &ieeesgechain->Flags, chainflags);

4690         /*
4691          * The size of the next frame is the accurate size of space
4692          * (in bytes) used to store the SGL elements. j is the counter
4693          * of SGL elements. (j - 1) is the number of SGL elements that
4694          * have been processed (stored in frames).
4695          */
4696         if (frames >= 2) {
4697                 chainlength = mpt->m_req_frame_size /
4698                     sizeof (MPI2_IEEE_SGE_SIMPLE64) *
4699                     sizeof (MPI2_IEEE_SGE_SIMPLE64);
4700         } else {
4701                 chainlength = ((cookiec - (j - 1)) *
4702                     sizeof (MPI2_IEEE_SGE_SIMPLE64));
4703         }

4705         p = cmd->cmd_extra_frames;

4707         ddi_put32(acc_hdl, &ieeesgechain->Length, chainlength);
4708         ddi_put32(acc_hdl, &ieeesgechain->Address.Low,
4709             p->m_phys_addr&0xffffffffull);
4710         ddi_put32(acc_hdl, &ieeesgechain->Address.High, p->m_phys_addr>>32);

4712         /*
4713          * If there are more than 2 frames left we have to
4714          * fill in the next chain offset to the location of
4715          * the chain element in the next frame.
4716          * sgemax is the number of simple elements in an extra
4717          * frame. Note that the value NextChainOffset should be
4718          * in double-words (4 bytes).
4719          */
4720         if (frames >= 2) {
4721                 ddi_put8(acc_hdl, &ieeesgechain->NextChainOffset,
4722                     (sgemax * sizeof (MPI2_IEEE_SGE_SIMPLE64)) >> 4);
4723         } else {
4724                 ddi_put8(acc_hdl, &ieeesgechain->NextChainOffset, 0);
4725         }

4727         /*
4728          * Jump to next frame;
4729          * Starting here, chain buffers go into the per command SGL.
4730          * This buffer is allocated when chain buffers are needed.
4731          */
4732         ieeesge = (pMpi2IeeeSgeSimple64_t)p->m_frames_addr;
4733         i = cookiec;

4735         /*
4736          * Start filling in frames with SGE's.  If we
4737          * reach the end of frame and still have SGE's
4738          * to fill we need to add a chain element and
4739          * use another frame.  j will be our counter
4740          * for what cookie we are at and i will be
```



```
4741          * the total cookiec. k is the current frame
4742          */
4743         for (k = 1; k <= frames; k++) {
4744                 for (l = 1; (l <= (sgemax + 1)) && (j <= i); j++, l++) {

4746                         /*
4747                          * If we have reached the end of frame
4748                          * and we have more SGE's to fill in
4749                          * we have to fill the final entry
4750                          * with a chain element and then
4751                          * continue to the next frame
4752                          */
4753                         if ((l == (sgemax + 1)) && (k != frames)) {
4754                                 ieeesgechain = (pMpi25IeeeSgeChain64_t)ieeesge;
4755                                 j--;
4756                                 chainflags =
4757                                     MPI2_IEEE_SGE_FLAGS_CHAIN_ELEMENT |
4758                                     MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR;
4759                                 ddi_put8(p->m_acc_hdl,
4760                                     &ieeesgechain->Flags, chainflags);
4761                                 /*
4762                                  * k is the frame counter and (k + 1)
4763                                  * is the number of the next frame.
4764                                  * Note that frames are in contiguous
4765                                  * memory space.
4766                                  */
4767                                 nframe_phys_addr = p->m_phys_addr +
4768                                     (mpt->m_req_frame_size * k);
4769                                 ddi_put32(p->m_acc_hdl,
4770                                     &ieeesgechain->Address.Low,
4771                                     nframe_phys_addr&0xffffffffull);
4772                                 ddi_put32(p->m_acc_hdl,
4773                                     &ieeesgechain->Address.High,
4774                                     nframe_phys_addr>>32);

4776                                 /*
4777                                  * If there are more than 2 frames left
4778                                  * we have to next chain offset to
4779                                  * the location of the chain element
4780                                  * in the next frame and fill in the
4781                                  * length of the next chain
4782                                  */
4783                                 if ((frames - k) >= 2) {
4784                                         ddi_put8(p->m_acc_hdl,
4785                                             &ieeesgechain->NextChainOffset,
4786                                             (sgemax *
4787                                             sizeof (MPI2_IEEE_SGE_SIMPLE64))
4788                                             >> 4);
4789                                         ddi_put32(p->m_acc_hdl,
4790                                             &ieeesgechain->Length,
4791                                             mpt->m_req_frame_size /
4792                                             sizeof (MPI2_IEEE_SGE_SIMPLE64) *
4793                                             sizeof (MPI2_IEEE_SGE_SIMPLE64));
4794                                 } else {
4795                                         /*
4796                                          * This is the last frame. Set
4797                                          * the NextChainOffset to 0 and
4798                                          * Length is the total size of
4799                                          * all remaining simple elements
4800                                          */
4801                                         ddi_put8(p->m_acc_hdl,
4802                                             &ieeesgechain->NextChainOffset,
4803                                             0);
4804                                         ddi_put32(p->m_acc_hdl,
4805                                             &ieeesgechain->Length,
4806                                             (cookiec - j) *
```

```
4807                                            sizeof (MPI2_IEEE_SGE_SIMPLE64));
4808                                }
4809
4810                                /* Jump to the next frame */
4811                                ieeesge = (pMpi2IeeeSgeSimple64_t)
4812                                    ((char *)p->m_frames_addr +
4813                                    (int)mpt->m_req_frame_size * k);
4814
4815                                continue;
4816                        }
4817
4818                        ddi_put32(p->m_acc_hdl,
4819                            &ieeesge->Address.Low,
4820                            dmap->addr.address64.Low);
4821                        ddi_put32(p->m_acc_hdl,
4822                            &ieeesge->Address.High,
4823                            dmap->addr.address64.High);
4824                        ddi_put32(p->m_acc_hdl,
4825                            &ieeesge->Length, dmap->count);
4826                        flags = (MPI2_IEEE_SGE_FLAGS_SIMPLE_ELEMENT |
4827                            MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR);
4828
4829                        /*
4830                         * If we are at the end of the frame and
4831                         * there is another frame to fill in
4832                         * do we need to do anything?
4833                         * if ((l == sgemax) && (k != frames)) {
4834                         * }
4835                         */
4836
4837                        /*
4838                         * If this is the final cookie set end of list.
4839                         */
4840                        if (j == i) {
4841                                flags |= MPI25_IEEE_SGE_FLAGS_END_OF_LIST;
4842                        }
4843
4844                        ddi_put8(p->m_acc_hdl, &ieeesge->Flags, flags);
4845                        dmap++;
4846                        ieeesge++;
4847                }
4848        }
4849
4850        /*
4851         * Sync DMA with the chain buffers that were just created
4852         */
4853        (void) ddi_dma_sync(p->m_dma_hdl, 0, 0, DDI_DMA_SYNC_FORDEV);
4854 }
4855
4856 static void
4857 mptsas_sge_setup(mptsas_t *mpt, mptsas_cmd_t *cmd, uint32_t *control,
4858     pMpi2SCSIIORequest_t frame, ddi_acc_handle_t acc_hdl)
4859 {
4860        ASSERT(cmd->cmd_flags & CFLAG_DMAVALID);
4861
4862        NDBG1(("mptsas_sge_setup: cookiec=%d", cmd->cmd_cookiec));
4863
4864        /*
4865         * Set read/write bit in control.
4866         */
4867        if (cmd->cmd_flags & CFLAG_DMASEND) {
4868                *control |= MPI2_SCSIIO_CONTROL_WRITE;
4869        } else {
4870                *control |= MPI2_SCSIIO_CONTROL_READ;
4871        }
```



```
4872
4873        ddi_put32(acc_hdl, &frame->DataLength, cmd->cmd_dmacount);
4874
4875        /*
4876         * We have 4 cases here.  First where we can fit all the
4877         * SG elements into the main frame, and the case
4878         * where we can't. The SG element is also different when using
4879         * MPI2.5 interface.
4880         * If we have more cookies than we can attach to a frame
4881         * we will need to use a chain element to point
4882         * a location of memory where the rest of the S/G
4883         * elements reside.
4884         */
4885        if (cmd->cmd_cookiec <= MPTSAS_MAX_FRAME_SGES64(mpt)) {
4886                if (mpt->m_MPI25) {
4887                        mptsas_ieee_sge_mainframe(cmd, frame, acc_hdl,
4888                            cmd->cmd_cookiec,
4889                            MPI25_IEEE_SGE_FLAGS_END_OF_LIST);
4890                } else {
4891                        mptsas_sge_mainframe(cmd, frame, acc_hdl,
4892                            cmd->cmd_cookiec,
4893                            ((uint32_t)(MPI2_SGE_FLAGS_LAST_ELEMENT
4894                            | MPI2_SGE_FLAGS_END_OF_BUFFER
4895                            | MPI2_SGE_FLAGS_END_OF_LIST) <<
4896                            MPI2_SGE_FLAGS_SHIFT));
4897                }
4898        } else {
4899                if (mpt->m_MPI25) {
4900                        mptsas_ieee_sge_chain(mpt, cmd, frame, acc_hdl);
4901                } else {
4902                        mptsas_sge_chain(mpt, cmd, frame, acc_hdl);
4903                }
4904        }
4905 }
4906
4907 /*
4908  * Interrupt handling
4909  * Utility routine.  Poll for status of a command sent to HBA
4910  * without interrupts (a FLAG_NOINTR command).
4911  */
4912 int
4913 mptsas_poll(mptsas_t *mpt, mptsas_cmd_t *poll_cmd, int polltime)
4914 {
4915        int     rval = TRUE;
4916
4917        NDBG5(("mptsas_poll: cmd=0x%p", (void *)poll_cmd));
4918
4919        if ((poll_cmd->cmd_flags & CFLAG_TM_CMD) == 0) {
4920                mptsas_restart_hba(mpt);
4921        }
4922
4923        /*
4924         * Wait, using drv_usecwait(), long enough for the command to
4925         * reasonably return from the target if the target isn't
4926         * "dead".  A polled command may well be sent from scsi_poll, and
4927         * there are retries built in to scsi_poll if the transport
4928         * accepted the packet (TRAN_ACCEPT).  scsi_poll waits 1 second
4929         * and retries the transport up to scsi_poll_busycnt times
4930         * (currently 60) if
4931         * 1. pkt_reason is CMD_INCOMPLETE and pkt_state is 0, or
4932         * 2. pkt_reason is CMD_CMPLT and *pkt_scbp has STATUS_BUSY
4933         *
4934         * limit the waiting to avoid a hang in the event that the
4935         * cmd never gets started but we are still receiving interrupts
4936         */
4937        while (!(poll_cmd->cmd_flags & CFLAG_FINISHED)) {
4938                if (mptsas_wait_intr(mpt, polltime) == FALSE) {
```

```
4939                        NDBG5(("mptsas_poll: command incomplete"));
4940                        rval = FALSE;
4941                        break;
4942                }
4943        }

4945        if (rval == FALSE) {

4947                /*
4948                 * this isn't supposed to happen, the hba must be wedged
4949                 * Mark this cmd as a timeout.
4950                 */
4951                mptsas_set_pkt_reason(mpt, poll_cmd, CMD_TIMEOUT,
4952                    (STAT_TIMEOUT|STAT_ABORTED));

4954                if (poll_cmd->cmd_queued == FALSE) {

4956                        NDBG5(("mptsas_poll: not on waitq"));

4958                        poll_cmd->cmd_pkt->pkt_state |=
4959                            (STATE_GOT_BUS|STATE_GOT_TARGET|STATE_SENT_CMD);
4960                } else {

4962                        /* find and remove it from the waitq */
4963                        NDBG5(("mptsas_poll: delete from waitq"));
4964                        mptsas_waitq_delete(mpt, poll_cmd);
4965                }

4967        }
4968        mptsas_fma_check(mpt, poll_cmd);
4969        NDBG5(("mptsas_poll: done"));
4970        return (rval);
4971 }
```

*_____unchanged_portion_omitted_*

```
5815 static void
5816 mptsas_process_intr(mptsas_t *mpt,
5817     pMpi2ReplyDescriptorsUnion_t reply_desc_union)
5818 {
5819        uint8_t reply_type;

5821        ASSERT(mutex_owned(&mpt->m_mutex));

5823        /*
5824         * The reply is valid, process it according to its
5825         * type.  Also, set a flag for updated the reply index
5826         * after they've all been processed.
5827         */
5828        reply_type = ddi_get8(mpt->m_acc_post_queue_hdl,
5829            &reply_desc_union->Default.ReplyFlags);
5830        reply_type &= MPI2_RPY_DESCRIPT_FLAGS_TYPE_MASK;
5831        if (reply_type == MPI2_RPY_DESCRIPT_FLAGS_SCSI_IO_SUCCESS ||
5832            reply_type == MPI25_RPY_DESCRIPT_FLAGS_FAST_PATH_SCSI_IO_SUCCESS) {
5494        if (reply_type == MPI2_RPY_DESCRIPT_FLAGS_SCSI_IO_SUCCESS) {
5833                mptsas_handle_scsi_io_success(mpt, reply_desc_union);
5834        } else if (reply_type == MPI2_RPY_DESCRIPT_FLAGS_ADDRESS_REPLY) {
5835                mptsas_handle_address_reply(mpt, reply_desc_union);
5836        } else {
5837                mptsas_log(mpt, CE_WARN, "?Bad reply type %x", reply_type);
5838                ddi_fm_service_impact(mpt->m_dip, DDI_SERVICE_UNAFFECTED);
5839        }

5841        /*
5842         * Clear the reply descriptor for re-use and increment
5843         * index.
5844         */
```



```
5845        ddi_put64(mpt->m_acc_post_queue_hdl,
5846            &((uint64_t *)(void *)mpt->m_post_queue)[mpt->m_post_index],
5847            0xFFFFFFFFFFFFFFFF);
5848        (void) ddi_dma_sync(mpt->m_dma_post_queue_hdl, 0, 0,
5849            DDI_DMA_SYNC_FORDEV);
5850 }
```

*_____unchanged_portion_omitted_*

```
9629 #ifdef MPTSAS_DEBUG
9630 /*PRINTFLIKE1*/
9631 void
9632 mptsas_printf(char *fmt, ...)
9633 {
9634        dev_info_t      *dev = 0;
9635        va_list         ap;

9637        mutex_enter(&mptsas_log_mutex);

9639        va_start(ap, fmt);
9640        (void) vsprintf(mptsas_log_buf, fmt, ap);
9641        va_end(ap);

9643 #ifdef PROM_PRINTF
9644        prom_printf("%s:\t%s\n", mptsas_label, mptsas_log_buf);
9645 #else
9646        scsi_log(dev, mptsas_label, CE_CONT, "!%s\n", mptsas_log_buf);
9308        scsi_log(dev, mptsas_label, SCSI_DEBUG, "%s\n", mptsas_log_buf);
9647 #endif
9648        mutex_exit(&mptsas_log_mutex);
9649 }
```

*_____unchanged_portion_omitted_*

```
10007 static void
10008 mptsas_passthru_sge(ddi_acc_handle_t acc_hdl, mptsas_pt_request_t *pt,
10009     pMpi2SGESimple64_t sgep)
9670 mptsas_start_passthru(mptsas_t *mpt, mptsas_cmd_t *cmd)
10010 {
10011        uint32_t                sge_flags;
10012        uint32_t                data_size, dataout_size;
9672        caddr_t                 memp;
9673        pMPI2RequestHeader_t    request_hdrp;
9674        struct scsi_pkt         *pkt = cmd->cmd_pkt;
9675        mptsas_pt_request_t     *pt = pkt->pkt_ha_private;
9676        uint32_t                request_size, data_size, dataout_size;
9677        uint32_t                direction;
10013        ddi_dma_cookie_t        data_cookie;
10014        ddi_dma_cookie_t        dataout_cookie;
9680        uint32_t                request_desc_low, request_desc_high = 0;
9681        uint32_t                i, sense_bufp;
9682        uint8_t                 desc_type;
9683        uint8_t                 *request, function;
9684        ddi_dma_handle_t        dma_hdl = mpt->m_dma_req_frame_hdl;
9685        ddi_acc_handle_t        acc_hdl = mpt->m_acc_req_frame_hdl;

9687        desc_type = MPI2_REQ_DESCRIPT_FLAGS_DEFAULT_TYPE;

9689        request = pt->request;
9690        direction = pt->direction;
9691        request_size = pt->request_size;
10016        data_size = pt->data_size;
10017        dataout_size = pt->dataout_size;
10018        data_cookie = pt->data_cookie;
10019        dataout_cookie = pt->dataout_cookie;

9697        /*
9698         * Store the passthrough message in memory location
```

```
9699          * corresponding to our slot number
9700          */
9701         memp = mpt->m_req_frame + (mpt->m_req_frame_size * cmd->cmd_slot);
9702         request_hdrp = (pMPI2RequestHeader_t)memp;
9703         bzero(memp, mpt->m_req_frame_size);

9705         for (i = 0; i < request_size; i++) {
9706                 bcopy(request + i, memp + i, 1);
9707         }

9709         if (data_size || dataout_size) {
9710                 pMpi2SGESimple64_t      sgep;
9711                 uint32_t                sge_flags;

9713                 sgep = (pMpi2SGESimple64_t)((uint8_t *)request_hdrp +
9714                     request_size);
10021         if (dataout_size) {

10022                 sge_flags = dataout_size |
10023                     ((uint32_t)(MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
10024                     MPI2_SGE_FLAGS_END_OF_BUFFER |
10025                     MPI2_SGE_FLAGS_HOST_TO_IOC |
10026                     MPI2_SGE_FLAGS_64_BIT_ADDRESSING) <<
10027                     MPI2_SGE_FLAGS_SHIFT);
10028                 ddi_put32(acc_hdl, &sgep->FlagsLength, sge_flags);
10029                 ddi_put32(acc_hdl, &sgep->Address.Low,
10030                     (uint32_t)(dataout_cookie.dmac_laddress & 0xffffffffull));
9725                         (uint32_t)(dataout_cookie.dmac_laddress &
9726                         0xffffffffull));
10031                 ddi_put32(acc_hdl, &sgep->Address.High,
10032                     (uint32_t)(dataout_cookie.dmac_laddress >> 32));
9728                         (uint32_t)(dataout_cookie.dmac_laddress
9729                         >> 32));
10033                 sgep++;
10034         }
10035         sge_flags = data_size;
10036         sge_flags |= ((uint32_t)(MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
10037             MPI2_SGE_FLAGS_LAST_ELEMENT |
10038             MPI2_SGE_FLAGS_END_OF_BUFFER |
10039             MPI2_SGE_FLAGS_END_OF_LIST |
10040             MPI2_SGE_FLAGS_64_BIT_ADDRESSING) <<
10041             MPI2_SGE_FLAGS_SHIFT);
10042         if (pt->direction == MPTSAS_PASS_THRU_DIRECTION_WRITE) {
9739                 if (direction == MPTSAS_PASS_THRU_DIRECTION_WRITE) {
10043                 sge_flags |= ((uint32_t)(MPI2_SGE_FLAGS_HOST_TO_IOC) <<
10044                     MPI2_SGE_FLAGS_SHIFT);
10045         } else {
10046                 sge_flags |= ((uint32_t)(MPI2_SGE_FLAGS_IOC_TO_HOST) <<
10047                     MPI2_SGE_FLAGS_SHIFT);
10048         }
10049         ddi_put32(acc_hdl, &sgep->FlagsLength, sge_flags);
9746                 ddi_put32(acc_hdl, &sgep->FlagsLength,
9747                     sge_flags);
10050         ddi_put32(acc_hdl, &sgep->Address.Low,
10051             (uint32_t)(data_cookie.dmac_laddress & 0xffffffffull));
9749                     (uint32_t)(data_cookie.dmac_laddress &
9750                     0xffffffffull));
10052         ddi_put32(acc_hdl, &sgep->Address.High,
10053             (uint32_t)(data_cookie.dmac_laddress >> 32));
10054 }

10056 static void
10057 mptsas_passthru_ieee_sge(ddi_acc_handle_t acc_hdl, mptsas_pt_request_t *pt,
10058     pMpi2IeeeSgeSimple64_t ieeesgep)
10059 {
10060         uint8_t                 sge_flags;
```



```
10061         uint32_t                data_size, dataout_size;
10062         ddi_dma_cookie_t        data_cookie;
10063         ddi_dma_cookie_t        dataout_cookie;

10065         data_size = pt->data_size;
10066         dataout_size = pt->dataout_size;
10067         data_cookie = pt->data_cookie;
10068         dataout_cookie = pt->dataout_cookie;

10070         sge_flags = (MPI2_IEEE_SGE_FLAGS_SIMPLE_ELEMENT |
10071             MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR);
10072         if (dataout_size) {
10073                 ddi_put32(acc_hdl, &ieeesgep->Length, dataout_size);
10074                 ddi_put32(acc_hdl, &ieeesgep->Address.Low,
10075                     (uint32_t)(dataout_cookie.dmac_laddress &
10076                     0xffffffffull));
10077                 ddi_put32(acc_hdl, &ieeesgep->Address.High,
10078                     (uint32_t)(dataout_cookie.dmac_laddress >> 32));
10079                 ddi_put8(acc_hdl, &ieeesgep->Flags, sge_flags);
10080                 ieeesgep++;
10081         }
10082         sge_flags |= MPI25_IEEE_SGE_FLAGS_END_OF_LIST;
10083         ddi_put32(acc_hdl, &ieeesgep->Length, data_size);
10084         ddi_put32(acc_hdl, &ieeesgep->Address.Low,
10085             (uint32_t)(data_cookie.dmac_laddress & 0xffffffffull));
10086         ddi_put32(acc_hdl, &ieeesgep->Address.High,
10087             (uint32_t)(data_cookie.dmac_laddress >> 32));
10088         ddi_put8(acc_hdl, &ieeesgep->Flags, sge_flags);
10089 }

10091 static void
10092 mptsas_start_passthru(mptsas_t *mpt, mptsas_cmd_t *cmd)
10093 {
10094         caddr_t                 memp;
10095         pMPI2RequestHeader_t    request_hdrp;
10096         struct scsi_pkt         *pkt = cmd->cmd_pkt;
10097         mptsas_pt_request_t     *pt = pkt->pkt_ha_private;
10098         uint32_t                request_size;
10099         uint32_t                request_desc_low, request_desc_high = 0;
10100         uint64_t                sense_bufp;
10101         uint8_t                 desc_type;
10102         uint8_t                 *request, function;
10103         ddi_dma_handle_t        dma_hdl = mpt->m_dma_req_frame_hdl;
10104         ddi_acc_handle_t        acc_hdl = mpt->m_acc_req_frame_hdl;

10106         desc_type = MPI2_REQ_DESCRIPT_FLAGS_DEFAULT_TYPE;

10108         request = pt->request;
10109         request_size = pt->request_size;

10111         /*
10112          * Store the passthrough message in memory location
10113          * corresponding to our slot number
10114          */
10115         memp = mpt->m_req_frame + (mpt->m_req_frame_size * cmd->cmd_slot);
10116         request_hdrp = (pMPI2RequestHeader_t)memp;
10117         bzero(memp, mpt->m_req_frame_size);

10119         bcopy(request, memp, request_size);

10121         NDBG15(("mptsas_start_passthru: Func 0x%x, MsgFlags 0x%x, "
10122             "size=%d, in %d, out %d", request_hdrp->Function,
10123             request_hdrp->MsgFlags, request_size,
10124             pt->data_size, pt->dataout_size));

10126         /*
```

```
10127          * Add an SGE, even if the length is zero.
10128          */
10129         if (mpt->m_MPI25 && pt->simple == 0) {
10130                 mptsas_passthru_ieee_sge(acc_hdl, pt,
10131                     (pMpi2IeeeSgeSimple64_t)
10132                     ((uint8_t *)request_hdrp + pt->sgl_offset));
10133         } else {
10134                 mptsas_passthru_sge(acc_hdl, pt,
10135                     (pMpi2SGESimple64_t)
10136                     ((uint8_t *)request_hdrp + pt->sgl_offset));
10137         }

10139         function = request_hdrp->Function;
10140         if ((function == MPI2_FUNCTION_SCSI_IO_REQUEST) ||
10141             (function == MPI2_FUNCTION_RAID_SCSI_IO_PASSTHROUGH)) {
10142                 pMpi2SCSIIORequest_t    scsi_io_req;

10144                 NDBG15(("mptsas_start_passthru: Is SCSI IO Req"));
10145                 scsi_io_req = (pMpi2SCSIIORequest_t)request_hdrp;
10146                 /*
10147                  * Put SGE for data and data_out buffer at the end of
10148                  * scsi_io_request message header.(64 bytes in total)
10149                  * Following above SGEs, the residual space will be
10150                  * used by sense data.
10151                  */
10152                 ddi_put8(acc_hdl,
10153                     &scsi_io_req->SenseBufferLength,
10154                     (uint8_t)(request_size - 64));

10156                 sense_bufp = (uint32_t)(mpt->m_req_frame_dma_addr +
10157                     (mpt->m_req_frame_size * cmd->cmd_slot) & 0xffffffffull);
9771                sense_bufp = mpt->m_req_frame_dma_addr +
9772                    (mpt->m_req_frame_size * cmd->cmd_slot);
10158                 sense_bufp += 64;
10159                 ddi_put32(acc_hdl,
10160                     &scsi_io_req->SenseBufferLowAddress, sense_bufp);

10162                 /*
10163                  * Set SGLOffset0 value
10164                  */
10165                 ddi_put8(acc_hdl, &scsi_io_req->SGLOffset0,
10166                     offsetof(MPI2_SCSI_IO_REQUEST, SGL) / 4);

10168                 /*
10169                  * Setup descriptor info.  RAID passthrough must use the
10170                  * default request descriptor which is already set, so if this
10171                  * is a SCSI IO request, change the descriptor to SCSI IO.
10172                  */
10173                 if (function == MPI2_FUNCTION_SCSI_IO_REQUEST) {
10174                         desc_type = MPI2_REQ_DESCRIPT_FLAGS_SCSI_IO;
10175                         request_desc_high = (ddi_get16(acc_hdl,
10176                             &scsi_io_req->DevHandle) << 16);
10177                 }
10178         }

10180         /*
10181          * We must wait till the message has been completed before
10182          * beginning the next message so we wait for this one to
10183          * finish.
10184          */
10185         (void) ddi_dma_sync(dma_hdl, 0, 0, DDI_DMA_SYNC_FORDEV);
10186         request_desc_low = (cmd->cmd_slot << 16) + desc_type;
10187         cmd->cmd_rfm = NULL;
10188         MPTSAS_START_CMD(mpt, request_desc_low, request_desc_high);
10189         if ((mptsas_check_dma_handle(dma_hdl) != DDI_SUCCESS) ||
10190             (mptsas_check_acc_handle(acc_hdl) != DDI_SUCCESS)) {
```



```
10191                 ddi_fm_service_impact(mpt->m_dip, DDI_SERVICE_UNAFFECTED);
10192         }
10193 }

10195 typedef void (mptsas_pre_f)(mptsas_t *, mptsas_pt_request_t *);
10196 static mptsas_pre_f     mpi_pre_ioc_facts;
10197 static mptsas_pre_f     mpi_pre_port_facts;
10198 static mptsas_pre_f     mpi_pre_fw_download;
10199 static mptsas_pre_f     mpi_pre_fw_25_download;
10200 static mptsas_pre_f     mpi_pre_fw_upload;
10201 static mptsas_pre_f     mpi_pre_fw_25_upload;
10202 static mptsas_pre_f     mpi_pre_sata_passthrough;
10203 static mptsas_pre_f     mpi_pre_smp_passthrough;
10204 static mptsas_pre_f     mpi_pre_config;
10205 static mptsas_pre_f     mpi_pre_sas_io_unit_control;
10206 static mptsas_pre_f     mpi_pre_scsi_io_req;

10208 /*
10209  * Prepare the pt for a SAS2 FW_DOWNLOAD request.
10210  */
10211 static void
10212 mpi_pre_fw_download(mptsas_t *mpt, mptsas_pt_request_t *pt)
10213 {
10214         pMpi2FWDownloadTCSGE_t tcsge;
10215         pMpi2FWDownloadRequest req;

10217         /*
10218          * If SAS3, call separate function.
10219          */
10220         if (mpt->m_MPI25) {
10221                 mpi_pre_fw_25_download(mpt, pt);
10222                 return;
10223         }

10225         /*
10226          * User requests should come in with the Transaction
10227          * context element where the SGL will go. Putting the
10228          * SGL after that seems to work, but don't really know
10229          * why. Other drivers tend to create an extra SGL and
10230          * refer to the TCE through that.
10231          */
10232         req = (pMpi2FWDownloadRequest)pt->request;
10233         tcsge = (pMpi2FWDownloadTCSGE_t)&req->SGL;
10234         if (tcsge->ContextSize != 0 || tcsge->DetailsLength != 12 ||
10235             tcsge->Flags != MPI2_SGE_FLAGS_TRANSACTION_ELEMENT) {
10236                 mptsas_log(mpt, CE_WARN, "FW Download tce invalid!");
10237         }

10239         pt->sgl_offset = offsetof(MPI2_FW_DOWNLOAD_REQUEST, SGL) +
10240             sizeof (*tcsge);
10241         if (pt->request_size != pt->sgl_offset)
10242                 NDBG15(("mpi_pre_fw_download(): Incorrect req size, "
10243                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10244                     (int)pt->request_size, (int)pt->sgl_offset,
10245                     (int)pt->dataout_size));
10246         if (pt->data_size < sizeof (MPI2_FW_DOWNLOAD_REPLY))
10247                 NDBG15(("mpi_pre_fw_download(): Incorrect rep size, "
10248                     "0x%x, should be 0x%x", pt->data_size,
10249                     (int)sizeof (MPI2_FW_DOWNLOAD_REPLY)));
10250 }

10252 /*
10253  * Prepare the pt for a SAS3 FW_DOWNLOAD request.
10254  */
10255 static void
10256 mpi_pre_fw_25_download(mptsas_t *mpt, mptsas_pt_request_t *pt)
```

```
10257 {
10258         pMpi2FWDownloadTCSGE_t tcsge;
10259         pMpi2FWDownloadRequest req2;
10260         pMpi25FWDownloadRequest req25;

10262         /*
10263          * User requests should come in with the Transaction
10264          * context element where the SGL will go. The new firmware
10265          * Doesn't use TCE and has space in the main request for
10266          * this information. So move to the right place.
10267          */
10268         req2 = (pMpi2FWDownloadRequest)pt->request;
10269         req25 = (pMpi25FWDownloadRequest)pt->request;
10270         tcsge = (pMpi2FWDownloadTCSGE_t)&req2->SGL;
10271         if (tcsge->ContextSize != 0 || tcsge->DetailsLength != 12 ||
10272             tcsge->Flags != MPI2_SGE_FLAGS_TRANSACTION_ELEMENT) {
10273                 mptsas_log(mpt, CE_WARN, "FW Download tce invalid!");
10274         }
10275         req25->ImageOffset = tcsge->ImageOffset;
10276         req25->ImageSize = tcsge->ImageSize;

10278         pt->sgl_offset = offsetof(MPI25_FW_DOWNLOAD_REQUEST, SGL);
10279         if (pt->request_size != pt->sgl_offset)
10280                 NDBG15(("mpi_pre_fw_25_download(): Incorrect req size, "
10281                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10282                     pt->request_size, pt->sgl_offset,
10283                     pt->dataout_size));
10284         if (pt->data_size < sizeof (MPI2_FW_DOWNLOAD_REPLY))
10285                 NDBG15(("mpi_pre_fw_25_download(): Incorrect rep size, "
10286                     "0x%x, should be 0x%x", pt->data_size,
10287                     (int)sizeof (MPI2_FW_UPLOAD_REPLY)));
10288 }

10290 /*
10291  * Prepare the pt for a SAS2 FW_UPLOAD request.
10292  */
10293 static void
10294 mpi_pre_fw_upload(mptsas_t *mpt, mptsas_pt_request_t *pt)
10295 {
10296         pMpi2FWUploadTCSGE_t tcsge;
10297         pMpi2FWUploadRequest_t req;

10299         /*
10300          * If SAS3, call separate function.
10301          */
10302         if (mpt->m_MPI25) {
10303                 mpi_pre_fw_25_upload(mpt, pt);
10304                 return;
10305         }

10307         /*
10308          * User requests should come in with the Transaction
10309          * context element where the SGL will go. Putting the
10310          * SGL after that seems to work, but don't really know
10311          * why. Other drivers tend to create an extra SGL and
10312          * refer to the TCE through that.
10313          */
10314         req = (pMpi2FWUploadRequest_t)pt->request;
10315         tcsge = (pMpi2FWUploadTCSGE_t)&req->SGL;
10316         if (tcsge->ContextSize != 0 || tcsge->DetailsLength != 12 ||
10317             tcsge->Flags != MPI2_SGE_FLAGS_TRANSACTION_ELEMENT) {
10318                 mptsas_log(mpt, CE_WARN, "FW Upload tce invalid!");
10319         }

10321         pt->sgl_offset = offsetof(MPI2_FW_UPLOAD_REQUEST, SGL) +
10322             sizeof (*tcsge);
```



```
10323         if (pt->request_size != pt->sgl_offset)
10324                 NDBG15(("mpi_pre_fw_upload(): Incorrect req size, "
10325                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10326                     pt->request_size, pt->sgl_offset,
10327                     pt->dataout_size));
10328         if (pt->data_size < sizeof (MPI2_FW_UPLOAD_REPLY))
10329                 NDBG15(("mpi_pre_fw_upload(): Incorrect rep size, "
10330                     "0x%x, should be 0x%x", pt->data_size,
10331                     (int)sizeof (MPI2_FW_UPLOAD_REPLY)));
10332 }

10334 /*
10335  * Prepare the pt a SAS3 FW_UPLOAD request.
10336  */
10337 static void
10338 mpi_pre_fw_25_upload(mptsas_t *mpt, mptsas_pt_request_t *pt)
10339 {
10340         pMpi2FWUploadTCSGE_t tcsge;
10341         pMpi2FWUploadRequest_t req2;
10342         pMpi25FWUploadRequest_t req25;

10344         /*
10345          * User requests should come in with the Transaction
10346          * context element where the SGL will go. The new firmware
10347          * Doesn't use TCE and has space in the main request for
10348          * this information. So move to the right place.
10349          */
10350         req2 = (pMpi2FWUploadRequest_t)pt->request;
10351         req25 = (pMpi25FWUploadRequest_t)pt->request;
10352         tcsge = (pMpi2FWUploadTCSGE_t)&req2->SGL;
10353         if (tcsge->ContextSize != 0 || tcsge->DetailsLength != 12 ||
10354             tcsge->Flags != MPI2_SGE_FLAGS_TRANSACTION_ELEMENT) {
10355                 mptsas_log(mpt, CE_WARN, "FW Upload tce invalid!");
10356         }
10357         req25->ImageOffset = tcsge->ImageOffset;
10358         req25->ImageSize = tcsge->ImageSize;

10360         pt->sgl_offset = offsetof(MPI25_FW_UPLOAD_REQUEST, SGL);
10361         if (pt->request_size != pt->sgl_offset)
10362                 NDBG15(("mpi_pre_fw_25_upload(): Incorrect req size, "
10363                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10364                     pt->request_size, pt->sgl_offset,
10365                     pt->dataout_size));
10366         if (pt->data_size < sizeof (MPI2_FW_UPLOAD_REPLY))
10367                 NDBG15(("mpi_pre_fw_25_upload(): Incorrect rep size, "
10368                     "0x%x, should be 0x%x", pt->data_size,
10369                     (int)sizeof (MPI2_FW_UPLOAD_REPLY)));
10370 }

10372 /*
10373  * Prepare the pt for an IOC_FACTS request.
10374  */
10375 static void
10376 mpi_pre_ioc_facts(mptsas_t *mpt, mptsas_pt_request_t *pt)
10377 {
10378 #ifndef __lock_lint
10379         _NOTE(ARGUNUSED(mpt))
10380 #endif
10381         if (pt->request_size != sizeof (MPI2_IOC_FACTS_REQUEST))
10382                 NDBG15(("mpi_pre_ioc_facts(): Incorrect req size, "
10383                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10384                     pt->request_size,
10385                     (int)sizeof (MPI2_IOC_FACTS_REQUEST),
10386                     pt->dataout_size));
10387         if (pt->data_size != sizeof (MPI2_IOC_FACTS_REPLY))
10388                 NDBG15(("mpi_pre_ioc_facts(): Incorrect rep size, "
```

```
10389                     "0x%x, should be 0x%x", pt->data_size,
10390                     (int)sizeof (MPI2_IOC_FACTS_REPLY)));
10391         pt->sgl_offset = (uint16_t)pt->request_size;
10392 }

10394 /*
10395  * Prepare the pt for a PORT_FACTS request.
10396  */
10397 static void
10398 mpi_pre_port_facts(mptsas_t *mpt, mptsas_pt_request_t *pt)
10399 {
10400 #ifndef __lock_lint
10401         _NOTE(ARGUNUSED(mpt))
10402 #endif
10403         if (pt->request_size != sizeof (MPI2_PORT_FACTS_REQUEST))
10404                 NDBG15(("mpi_pre_port_facts(): Incorrect req size, "
10405                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10406                     pt->request_size,
10407                     (int)sizeof (MPI2_PORT_FACTS_REQUEST),
10408                     pt->dataout_size));
10409         if (pt->data_size != sizeof (MPI2_PORT_FACTS_REPLY))
10410                 NDBG15(("mpi_pre_port_facts(): Incorrect rep size, "
10411                     "0x%x, should be 0x%x", pt->data_size,
10412                     (int)sizeof (MPI2_PORT_FACTS_REPLY)));
10413         pt->sgl_offset = (uint16_t)pt->request_size;
10414 }

10416 /*
10417  * Prepare pt for a SATA_PASSTHROUGH request.
10418  */
10419 static void
10420 mpi_pre_sata_passthrough(mptsas_t *mpt, mptsas_pt_request_t *pt)
10421 {
10422 #ifndef __lock_lint
10423         _NOTE(ARGUNUSED(mpt))
10424 #endif
10425         pt->sgl_offset = offsetof(MPI2_SATA_PASSTHROUGH_REQUEST, SGL);
10426         if (pt->request_size != pt->sgl_offset)
10427                 NDBG15(("mpi_pre_sata_passthrough(): Incorrect req size, "
10428                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10429                     pt->request_size, pt->sgl_offset,
10430                     pt->dataout_size));
10431         if (pt->data_size != sizeof (MPI2_SATA_PASSTHROUGH_REPLY))
10432                 NDBG15(("mpi_pre_sata_passthrough(): Incorrect rep size, "
10433                     "0x%x, should be 0x%x", pt->data_size,
10434                     (int)sizeof (MPI2_SATA_PASSTHROUGH_REPLY)));
10435 }

10437 static void
10438 mpi_pre_smp_passthrough(mptsas_t *mpt, mptsas_pt_request_t *pt)
10439 {
10440 #ifndef __lock_lint
10441         _NOTE(ARGUNUSED(mpt))
10442 #endif
10443         pt->sgl_offset = offsetof(MPI2_SMP_PASSTHROUGH_REQUEST, SGL);
10444         if (pt->request_size != pt->sgl_offset)
10445                 NDBG15(("mpi_pre_smp_passthrough(): Incorrect req size, "
10446                     "0x%x, should be 0x%x, dataoutsz 0x%x",
10447                     pt->request_size, pt->sgl_offset,
10448                     pt->dataout_size));
10449         if (pt->data_size != sizeof (MPI2_SMP_PASSTHROUGH_REPLY))
10450                 NDBG15(("mpi_pre_smp_passthrough(): Incorrect rep size, "
10451                     "0x%x, should be 0x%x", pt->data_size,
10452                     (int)sizeof (MPI2_SMP_PASSTHROUGH_REPLY)));
10453 }
```



```
10455 /*
10456  * Prepare pt for a CONFIG request.
10457  */
10458 static void
10459 mpi_pre_config(mptsas_t *mpt, mptsas_pt_request_t *pt)
10460 {
10461 #ifndef __lock_lint
10462         _NOTE(ARGUNUSED(mpt))
10463 #endif
10464         pt->sgl_offset = offsetof(MPI2_CONFIG_REQUEST, PageBufferSGE);
10465         if (pt->request_size != pt->sgl_offset)
10466                 NDBG15(("mpi_pre_config(): Incorrect req size, 0x%x, "
10467                     "should be 0x%x, dataoutsz 0x%x", pt->request_size,
10468                     pt->sgl_offset, pt->dataout_size));
10469         if (pt->data_size != sizeof (MPI2_CONFIG_REPLY))
10470                 NDBG15(("mpi_pre_config(): Incorrect rep size, 0x%x, "
10471                     "should be 0x%x", pt->data_size,
10472                     (int)sizeof (MPI2_CONFIG_REPLY)));
10473         pt->simple = 1;
10474 }

10476 /*
10477  * Prepare pt for a SCSI_IO_REQ request.
10478  */
10479 static void
10480 mpi_pre_scsi_io_req(mptsas_t *mpt, mptsas_pt_request_t *pt)
10481 {
10482 #ifndef __lock_lint
10483         _NOTE(ARGUNUSED(mpt))
10484 #endif
10485         pt->sgl_offset = offsetof(MPI2_SCSI_IO_REQUEST, SGL);
10486         if (pt->request_size != pt->sgl_offset)
10487                 NDBG15(("mpi_pre_config(): Incorrect req size, 0x%x, "
10488                     "should be 0x%x, dataoutsz 0x%x", pt->request_size,
10489                     pt->sgl_offset,
10490                     pt->dataout_size));
10491         if (pt->data_size != sizeof (MPI2_SCSI_IO_REPLY))
10492                 NDBG15(("mpi_pre_config(): Incorrect rep size, 0x%x, "
10493                     "should be 0x%x", pt->data_size,
10494                     (int)sizeof (MPI2_SCSI_IO_REPLY)));
10495 }

10497 /*
10498  * Prepare the mptsas_cmd for a SAS_IO_UNIT_CONTROL request.
10499  */
10500 static void
10501 mpi_pre_sas_io_unit_control(mptsas_t *mpt, mptsas_pt_request_t *pt)
10502 {
10503 #ifndef __lock_lint
10504         _NOTE(ARGUNUSED(mpt))
10505 #endif
10506         pt->sgl_offset = (uint16_t)pt->request_size;
10507 }

10509 /*
10510  * A set of functions to prepare an mptsas_cmd for the various
10511  * supported requests.
10512  */
10513 static struct mptsas_func {
10514         U8              Function;
10515         char            *Name;
10516         mptsas_pre_f    *f_pre;
10517 } mptsas_func_list[] = {
10518         { MPI2_FUNCTION_IOC_FACTS, "IOC_FACTS",         mpi_pre_ioc_facts },
10519         { MPI2_FUNCTION_PORT_FACTS, "PORT_FACTS",       mpi_pre_port_facts },
10520         { MPI2_FUNCTION_FW_DOWNLOAD, "FW_DOWNLOAD",     mpi_pre_fw_download },
```

```
10521         { MPI2_FUNCTION_FW_UPLOAD, "FW_UPLOAD",          mpi_pre_fw_upload },
10522         { MPI2_FUNCTION_SATA_PASSTHROUGH, "SATA_PASSTHROUGH",
10523             mpi_pre_sata_passthrough },
10524         { MPI2_FUNCTION_SMP_PASSTHROUGH, "SMP_PASSTHROUGH",
10525             mpi_pre_smp_passthrough},
10526         { MPI2_FUNCTION_SCSI_IO_REQUEST, "SCSI_IO_REQUEST",
10527             mpi_pre_scsi_io_req},
10528         { MPI2_FUNCTION_CONFIG, "CONFIG",                mpi_pre_config},
10529         { MPI2_FUNCTION_SAS_IO_UNIT_CONTROL, "SAS_IO_UNIT_CONTROL",
10530             mpi_pre_sas_io_unit_control },
10531         { 0xFF, NULL,                               NULL } /* list end */
10532 };

10534 static void
10535 mptsas_prep_sgl_offset(mptsas_t *mpt, mptsas_pt_request_t *pt)
10536 {
10537         pMPI2RequestHeader_t    hdr;
10538         struct mptsas_func      *f;

10540         hdr = (pMPI2RequestHeader_t)pt->request;

10542         for (f = mptsas_func_list; f->f_pre != NULL; f++) {
10543                 if (hdr->Function == f->Function) {
10544                         f->f_pre(mpt, pt);
10545                         NDBG15(("mptsas_prep_sgl_offset: Function %s,"
10546                             " sgl_offset 0x%x", f->Name,
10547                             pt->sgl_offset));
10548                         return;
10549                 }
10550         }
10551         NDBG15(("mptsas_prep_sgl_offset: Unknown Function 0x%02x,"
10552             " returning req_size 0x%x for sgl_offset",
10553             hdr->Function, pt->request_size));
10554         pt->sgl_offset = (uint16_t)pt->request_size;
10555 }


10558 static int
10559 mptsas_do_passthru(mptsas_t *mpt, uint8_t *request, uint8_t *reply,
10560     uint8_t *data, uint32_t request_size, uint32_t reply_size,
10561     uint32_t data_size, uint8_t direction, uint8_t *dataout,
9815     uint32_t data_size, uint32_t direction, uint8_t *dataout,
10562     uint32_t dataout_size, short timeout, int mode)
10563 {
10564         mptsas_pt_request_t             pt;
10565         mptsas_dma_alloc_state_t        data_dma_state;
10566         mptsas_dma_alloc_state_t        dataout_dma_state;
10567         caddr_t                         memp;
10568         mptsas_cmd_t                    *cmd = NULL;
10569         struct scsi_pkt                 *pkt;
10570         uint32_t                        reply_len = 0, sense_len = 0;
10571         pMPI2RequestHeader_t            request_hdrp;
10572         pMPI2RequestHeader_t            request_msg;
10573         pMPI2DefaultReply_t             reply_msg;
10574         Mpi2SCSIIOReply_t               rep_msg;
10575         int                             i, status = 0, pt_flags = 0, rv = 0;
10576         int                             rvalue;
10577         uint8_t                         function;

10579         ASSERT(mutex_owned(&mpt->m_mutex));

10581         reply_msg = (pMPI2DefaultReply_t)(&rep_msg);
10582         bzero(reply_msg, sizeof (MPI2_DEFAULT_REPLY));
10583         request_msg = kmem_zalloc(request_size, KM_SLEEP);

10585         mutex_exit(&mpt->m_mutex);
```



```
10586         /*
10587          * copy in the request buffer since it could be used by
10588          * another thread when the pt request into waitq
10589          */
10590         if (ddi_copyin(request, request_msg, request_size, mode)) {
10591                 mutex_enter(&mpt->m_mutex);
10592                 status = EFAULT;
10593                 mptsas_log(mpt, CE_WARN, "failed to copy request data");
10594                 goto out;
10595         }
10596         mutex_enter(&mpt->m_mutex);

10598         function = request_msg->Function;
10599         if (function == MPI2_FUNCTION_SCSI_TASK_MGMT) {
10600                 pMpi2SCSITaskManagementRequest_t        task;
10601                 task = (pMpi2SCSITaskManagementRequest_t)request_msg;
10602                 mptsas_setup_bus_reset_delay(mpt);
10603                 rv = mptsas_ioc_task_management(mpt, task->TaskType,
10604                     task->DevHandle, (int)task->LUN[1], reply, reply_size,
10605                     mode);

10607                 if (rv != TRUE) {
10608                         status = EIO;
10609                         mptsas_log(mpt, CE_WARN, "task management failed");
10610                 }
10611                 goto out;
10612         }

10614         if (data_size != 0) {
10615                 data_dma_state.size = data_size;
10616                 if (mptsas_dma_alloc(mpt, &data_dma_state) != DDI_SUCCESS) {
10617                         status = ENOMEM;
10618                         mptsas_log(mpt, CE_WARN, "failed to alloc DMA "
10619                             "resource");
10620                         goto out;
10621                 }
10622                 pt_flags |= MPTSAS_DATA_ALLOCATED;
10623                 if (direction == MPTSAS_PASS_THRU_DIRECTION_WRITE) {
10624                         mutex_exit(&mpt->m_mutex);
10625                         for (i = 0; i < data_size; i++) {
10626                                 if (ddi_copyin(data + i, (uint8_t *)
10627                                     data_dma_state.memp + i, 1, mode)) {
10628                                         mutex_enter(&mpt->m_mutex);
10629                                         status = EFAULT;
10630                                         mptsas_log(mpt, CE_WARN, "failed to "
10631                                             "copy read data");
10632                                         goto out;
10633                                 }
10634                         }
10635                         mutex_enter(&mpt->m_mutex);
10636                 }
10637         }
10638         else
10639                 bzero(&data_dma_state, sizeof (data_dma_state));

10641         if (dataout_size != 0) {
10642                 dataout_dma_state.size = dataout_size;
10643                 if (mptsas_dma_alloc(mpt, &dataout_dma_state) != DDI_SUCCESS) {
10644                         status = ENOMEM;
10645                         mptsas_log(mpt, CE_WARN, "failed to alloc DMA "
10646                             "resource");
10647                         goto out;
10648                 }
10649                 pt_flags |= MPTSAS_DATAOUT_ALLOCATED;
10650                 mutex_exit(&mpt->m_mutex);
10651                 for (i = 0; i < dataout_size; i++) {
```

```
10652                         if (ddi_copyin(dataout + i, (uint8_t *)
10653                             dataout_dma_state.memp + i, 1, mode)) {
10654                                 mutex_enter(&mpt->m_mutex);
10655                                 mptsas_log(mpt, CE_WARN, "failed to copy out"
10656                                     " data");
10657                                 status = EFAULT;
10658                                 goto out;
10659                         }
10660                 }
10661                 mutex_enter(&mpt->m_mutex);
10662         }
10663         else
10664                 bzero(&dataout_dma_state, sizeof (dataout_dma_state));

10666         if ((rvalue = (mptsas_request_from_pool(mpt, &cmd, &pkt))) == -1) {
10667                 status = EAGAIN;
10668                 mptsas_log(mpt, CE_NOTE, "event ack command pool is full");
10669                 goto out;
10670         }
10671         pt_flags |= MPTSAS_REQUEST_POOL_CMD;

10673         bzero((caddr_t)cmd, sizeof (*cmd));
10674         bzero((caddr_t)pkt, scsi_pkt_size());
10675         bzero((caddr_t)&pt, sizeof (pt));

10677         cmd->ioc_cmd_slot = (uint32_t)(rvalue);

10679         pt.request = (uint8_t *)request_msg;
10680         pt.direction = direction;
10681         pt.simple = 0;
10682         pt.request_size = request_size;
10683         pt.data_size = data_size;
10684         pt.dataout_size = dataout_size;
10685         pt.data_cookie = data_dma_state.cookie;
10686         pt.dataout_cookie = dataout_dma_state.cookie;
10687         mptsas_prep_sgl_offset(mpt, &pt);

10689         /*
10690          * Form a blank cmd/pkt to store the acknowledgement message
10691          */
10692         pkt->pkt_cdbp           = (opaque_t)&cmd->cmd_cdb[0];
10693         pkt->pkt_scbp           = (opaque_t)&cmd->cmd_scb;
10694         pkt->pkt_ha_private     = (opaque_t)&pt;
10695         pkt->pkt_flags          = FLAG_HEAD;
10696         pkt->pkt_time           = timeout;
10697         cmd->cmd_pkt            = pkt;
10698         cmd->cmd_flags          = CFLAG_CMDIOC | CFLAG_PASSTHRU;

10700         /*
10701          * Save the command in a slot
10702          */
10703         if (mptsas_save_cmd(mpt, cmd) == TRUE) {
10704                 /*
10705                  * Once passthru command get slot, set cmd_flags
10706                  * CFLAG_PREPARED.
10707                  */
10708                 cmd->cmd_flags |= CFLAG_PREPARED;
10709                 mptsas_start_passthru(mpt, cmd);
10710         } else {
10711                 mptsas_waitq_add(mpt, cmd);
10712         }

10714         while ((cmd->cmd_flags & CFLAG_FINISHED) == 0) {
10715                 cv_wait(&mpt->m_passthru_cv, &mpt->m_mutex);
10716         }
```



```
10718         if (cmd->cmd_flags & CFLAG_PREPARED) {
10719                 memp = mpt->m_req_frame + (mpt->m_req_frame_size *
10720                     cmd->cmd_slot);
10721                 request_hdrp = (pMPI2RequestHeader_t)memp;
10722         }

10724         if (cmd->cmd_flags & CFLAG_TIMEOUT) {
10725                 status = ETIMEDOUT;
10726                 mptsas_log(mpt, CE_WARN, "passthrough command timeout");
10727                 pt_flags |= MPTSAS_CMD_TIMEOUT;
10728                 goto out;
10729         }

10731         if (cmd->cmd_rfm) {
10732                 /*
10733                  * cmd_rfm is zero means the command reply is a CONTEXT
10734                  * reply and no PCI Write to post the free reply SMFA
10735                  * because no reply message frame is used.
10736                  * cmd_rfm is non-zero means the reply is a ADDRESS
10737                  * reply and reply message frame is used.
10738                  */
10739                 pt_flags |= MPTSAS_ADDRESS_REPLY;
10740                 (void) ddi_dma_sync(mpt->m_dma_reply_frame_hdl, 0, 0,
10741                     DDI_DMA_SYNC_FORCPU);
10742                 reply_msg = (pMPI2DefaultReply_t)
10743                     (mpt->m_reply_frame + (cmd->cmd_rfm -
10744                     mpt->m_reply_frame_dma_addr));
10745         }

10747         mptsas_fma_check(mpt, cmd);
10748         if (pkt->pkt_reason == CMD_TRAN_ERR) {
10749                 status = EAGAIN;
10750                 mptsas_log(mpt, CE_WARN, "passthru fma error");
10751                 goto out;
10752         }
10753         if (pkt->pkt_reason == CMD_RESET) {
10754                 status = EAGAIN;
10755                 mptsas_log(mpt, CE_WARN, "ioc reset abort passthru");
10756                 goto out;
10757         }

10759         if (pkt->pkt_reason == CMD_INCOMPLETE) {
10760                 status = EIO;
10761                 mptsas_log(mpt, CE_WARN, "passthrough command incomplete");
10762                 goto out;
10763         }

10765         mutex_exit(&mpt->m_mutex);
10766         if (cmd->cmd_flags & CFLAG_PREPARED) {
10767                 function = request_hdrp->Function;
10768                 if ((function == MPI2_FUNCTION_SCSI_IO_REQUEST) ||
10769                     (function == MPI2_FUNCTION_RAID_SCSI_IO_PASSTHROUGH)) {
10770                         reply_len = sizeof (MPI2_SCSI_IO_REPLY);
10771                         sense_len = reply_size - reply_len;
10772                 } else {
10773                         reply_len = reply_size;
10774                         sense_len = 0;
10775                 }

10777                 for (i = 0; i < reply_len; i++) {
10778                         if (ddi_copyout((uint8_t *)reply_msg + i, reply + i, 1,
10779                             mode)) {
10780                                 mutex_enter(&mpt->m_mutex);
10781                                 status = EFAULT;
10782                                 mptsas_log(mpt, CE_WARN, "failed to copy out "
10783                                     "reply data");
```

```
10784                                   goto out;
10785                           }
10786                   }
10787                   for (i = 0; i < sense_len; i++) {
10788                           if (ddi_copyout((uint8_t *)request_hdrp + 64 + i,
10789                               reply + reply_len + i, 1, mode)) {
10790                                   mutex_enter(&mpt->m_mutex);
10791                                   status = EFAULT;
10792                                   mptsas_log(mpt, CE_WARN, "failed to copy out "
10793                                       "sense data");
10794                                   goto out;
10795                           }
10796                   }
10797           }

10799           if (data_size) {
10800                   if (direction != MPTSAS_PASS_THRU_DIRECTION_WRITE) {
10801                           (void) ddi_dma_sync(data_dma_state.handle, 0, 0,
10802                               DDI_DMA_SYNC_FORCPU);
10803                           for (i = 0; i < data_size; i++) {
10804                                   if (ddi_copyout((uint8_t *)(
10805                                       data_dma_state.memp + i), data + i,  1,
10806                                       mode)) {
10807                                           mutex_enter(&mpt->m_mutex);
10808                                           status = EFAULT;
10809                                           mptsas_log(mpt, CE_WARN, "failed to "
10810                                               "copy out the reply data");
10811                                           goto out;
10812                                   }
10813                           }
10814                   }
10815           }
10816           mutex_enter(&mpt->m_mutex);
10817   out:
10818           /*
10819            * Put the reply frame back on the free queue, increment the free
10820            * index, and write the new index to the free index register.  But only
10821            * if this reply is an ADDRESS reply.
10822            */
10823           if (pt_flags & MPTSAS_ADDRESS_REPLY) {
10824                   ddi_put32(mpt->m_acc_free_queue_hdl,
10825                       &((uint32_t *)(void *)mpt->m_free_queue)[mpt->m_free_index],
10826                       cmd->cmd_rfm);
10827                   (void) ddi_dma_sync(mpt->m_dma_free_queue_hdl, 0, 0,
10828                       DDI_DMA_SYNC_FORDEV);
10829                   if (++mpt->m_free_index == mpt->m_free_queue_depth) {
10830                           mpt->m_free_index = 0;
10831                   }
10832                   ddi_put32(mpt->m_datap, &mpt->m_reg->ReplyFreeHostIndex,
10833                       mpt->m_free_index);
10834           }
10835           if (cmd && (cmd->cmd_flags & CFLAG_PREPARED)) {
10836                   mptsas_remove_cmd(mpt, cmd);
10837                   pt_flags &= (~MPTSAS_REQUEST_POOL_CMD);
10838           }
10839           if (pt_flags & MPTSAS_REQUEST_POOL_CMD)
10840                   mptsas_return_to_pool(mpt, cmd);
10841           if (pt_flags & MPTSAS_DATA_ALLOCATED) {
10842                   if (mptsas_check_dma_handle(data_dma_state.handle) !=
10843                       DDI_SUCCESS) {
10844                           ddi_fm_service_impact(mpt->m_dip,
10845                               DDI_SERVICE_UNAFFECTED);
10846                           status = EFAULT;
10847                   }
10848                   mptsas_dma_free(&data_dma_state);
10849           }
```



```
10850           if (pt_flags & MPTSAS_DATAOUT_ALLOCATED) {
10851                   if (mptsas_check_dma_handle(dataout_dma_state.handle) !=
10852                       DDI_SUCCESS) {
10853                           ddi_fm_service_impact(mpt->m_dip,
10854                               DDI_SERVICE_UNAFFECTED);
10855                           status = EFAULT;
10856                   }
10857                   mptsas_dma_free(&dataout_dma_state);
10858           }
10859           if (pt_flags & MPTSAS_CMD_TIMEOUT) {
10860                   if ((mptsas_restart_ioc(mpt)) == DDI_FAILURE) {
10861                           mptsas_log(mpt, CE_WARN, "mptsas_restart_ioc failed");
10862                   }
10863           }
10864           if (request_msg)
10865                   kmem_free(request_msg, request_size);

10867           return (status);
10868   }

10870   static int
10871   mptsas_pass_thru(mptsas_t *mpt, mptsas_pass_thru_t *data, int mode)
10872   {
10873           /*
10874            * If timeout is 0, set timeout to default of 60 seconds.
10875            */
10876           if (data->Timeout == 0) {
10877                   data->Timeout = MPTSAS_PASS_THRU_TIME_DEFAULT;
10878           }

10880           if (((data->DataSize == 0) &&
10881               (data->DataDirection == MPTSAS_PASS_THRU_DIRECTION_NONE)) ||
10882               ((data->DataSize != 0) &&
10883               ((data->DataDirection == MPTSAS_PASS_THRU_DIRECTION_READ) ||
10884               (data->DataDirection == MPTSAS_PASS_THRU_DIRECTION_WRITE) ||
10885               ((data->DataDirection == MPTSAS_PASS_THRU_DIRECTION_BOTH) &&
10886               (data->DataOutSize != 0))))) {
10887                   if (data->DataDirection == MPTSAS_PASS_THRU_DIRECTION_BOTH) {
10888                           data->DataDirection = MPTSAS_PASS_THRU_DIRECTION_READ;
10889                   } else {
10890                           data->DataOutSize = 0;
10891                   }
10892                   /*
10893                    * Send passthru request messages
10894                    */
10895                   return (mptsas_do_passthru(mpt,
10896                       (uint8_t *)((uintptr_t)data->PtrRequest),
10897                       (uint8_t *)((uintptr_t)data->PtrReply),
10898                       (uint8_t *)((uintptr_t)data->PtrData),
10899                       data->RequestSize, data->ReplySize,
10900                       data->DataSize, (uint8_t)data->DataDirection,
10148                      data->DataSize, data->DataDirection,
10901                       (uint8_t *)((uintptr_t)data->PtrDataOut),
10902                       data->DataOutSize, data->Timeout, mode));
10903           } else {
10904                   return (EINVAL);
10905           }
10906   }
```

*unchanged_portion_omitted_*

```
11979 static void
11980 mptsas_read_adapter_data(mptsas_t *mpt, mptsas_adapter_data_t *adapter_data)
11981 {
11982           char    *driver_verstr = MPTSAS_MOD_STRING;

11984           mptsas_lookup_pci_data(mpt, adapter_data);
```

```
11985         adapter_data->AdapterType = mpt->m_MPI25 ?
11986             MPTIOCTL_ADAPTER_TYPE_SAS3 :
11987             MPTIOCTL_ADAPTER_TYPE_SAS2;
11233         adapter_data->AdapterType = MPTIOCTL_ADAPTER_TYPE_SAS2;
11988         adapter_data->PCIDeviceHwId = (uint32_t)mpt->m_devid;
11989         adapter_data->PCIDeviceHwRev = (uint32_t)mpt->m_revid;
11990         adapter_data->SubSystemId = (uint32_t)mpt->m_ssid;
11991         adapter_data->SubsystemVendorId = (uint32_t)mpt->m_svid;
11992         (void) strcpy((char *)&adapter_data->DriverVersion[0], driver_verstr);
11993         adapter_data->BiosVersion = 0;
11994         (void) mptsas_get_bios_page3(mpt, &adapter_data->BiosVersion);
11995 }
______unchanged_portion_omitted_

15733 /* smp transport routine */
15734 static int mptsas_smp_start(struct smp_pkt *smp_pkt)
15735 {
15736         uint64_t                        wwn;
15737         Mpi2SmpPassthroughRequest_t     req;
15738         Mpi2SmpPassthroughReply_t       rep;
15739         uint8_t                         direction = 0;
14985         uint32_t                        direction = 0;
15740         mptsas_t                        *mpt;
15741         int                             ret;
15742         uint64_t                        tmp64;

15744         mpt = (mptsas_t *)smp_pkt->smp_pkt_address->
15745             smp_a_hba_tran->smp_tran_hba_private;

15747         bcopy(smp_pkt->smp_pkt_address->smp_a_wwn, &wwn, SAS_WWN_BYTE_SIZE);
15748         /*
15749          * Need to compose a SMP request message
15750          * and call mptsas_do_passthru() function
15751          */
15752         bzero(&req, sizeof (req));
15753         bzero(&rep, sizeof (rep));
15754         req.PassthroughFlags = 0;
15755         req.PhysicalPort = 0xff;
15756         req.ChainOffset = 0;
15757         req.Function = MPI2_FUNCTION_SMP_PASSTHROUGH;

15759         if ((smp_pkt->smp_pkt_reqsize & 0xffff0000ul) != 0) {
15760                 smp_pkt->smp_pkt_reason = ERANGE;
15761                 return (DDI_FAILURE);
15762         }
15763         req.RequestDataLength = LE_16((uint16_t)(smp_pkt->smp_pkt_reqsize - 4));

15765         req.MsgFlags = 0;
15766         tmp64 = LE_64(wwn);
15767         bcopy(&tmp64, &req.SASAddress, SAS_WWN_BYTE_SIZE);
15768         if (smp_pkt->smp_pkt_rspsize > 0) {
15769                 direction |= MPTSAS_PASS_THRU_DIRECTION_READ;
15770         }
15771         if (smp_pkt->smp_pkt_reqsize > 0) {
15772                 direction |= MPTSAS_PASS_THRU_DIRECTION_WRITE;
15773         }

15775         mutex_enter(&mpt->m_mutex);
15776         ret = mptsas_do_passthru(mpt, (uint8_t *)&req, (uint8_t *)&rep,
15777             (uint8_t *)smp_pkt->smp_pkt_rsp,
15778             offsetof(Mpi2SmpPassthroughRequest_t, SGL), sizeof (rep),
15779             smp_pkt->smp_pkt_rspsize - 4, direction,
15780             (uint8_t *)smp_pkt->smp_pkt_req, smp_pkt->smp_pkt_reqsize - 4,
15781             smp_pkt->smp_pkt_timeout, FKIOCTL);
15782         mutex_exit(&mpt->m_mutex);
15783         if (ret != 0) {
```



```
15784                 cmn_err(CE_WARN, "smp_start do passthru error %d", ret);
15785                 smp_pkt->smp_pkt_reason = (uchar_t)(ret);
15786                 return (DDI_FAILURE);
15787         }
15788         /* do passthrough success, check the smp status */
15789         if (LE_16(rep.IOCStatus) != MPI2_IOCSTATUS_SUCCESS) {
15790                 switch (LE_16(rep.IOCStatus)) {
15791                 case MPI2_IOCSTATUS_SCSI_DEVICE_NOT_THERE:
15792                         smp_pkt->smp_pkt_reason = ENODEV;
15793                         break;
15794                 case MPI2_IOCSTATUS_SAS_SMP_DATA_OVERRUN:
15795                         smp_pkt->smp_pkt_reason = EOVERFLOW;
15796                         break;
15797                 case MPI2_IOCSTATUS_SAS_SMP_REQUEST_FAILED:
15798                         smp_pkt->smp_pkt_reason = EIO;
15799                         break;
15800                 default:
15801                         mptsas_log(mpt, CE_NOTE, "smp_start: get unknown ioc"
15802                             "status:%x", LE_16(rep.IOCStatus));
15803                         smp_pkt->smp_pkt_reason = EIO;
15804                         break;
15805                 }
15806                 return (DDI_FAILURE);
15807         }
15808         if (rep.SASStatus != MPI2_SASSTATUS_SUCCESS) {
15809                 mptsas_log(mpt, CE_NOTE, "smp_start: get error SAS status:%x",
15810                     rep.SASStatus);
15811                 smp_pkt->smp_pkt_reason = EIO;
15812                 return (DDI_FAILURE);
15813         }

15815         return (DDI_SUCCESS);
15816 }
______unchanged_portion_omitted_
```

```
**********************************************************
   83458 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/io/scsi/adapters/mpt_sas/mptsas_impl.c
NEX-1889 upstream
**********************************************************
```

```
   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) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
  24  * Copyright 2012 Nexenta Systems, Inc. All rights reserved.
  25  * Copyright 2014 OmniTI Computer Consulting, Inc. All rights reserved.
  26  * Copyright (c) 2014, Tegile Systems Inc. All rights reserved.
  27  */

  29 /*
  30  * Copyright (c) 2000 to 2010, LSI Corporation.
  31  * All rights reserved.
  32  *
  33  * Redistribution and use in source and binary forms of all code within
  34  * this file that is exclusively owned by LSI, with or without
  35  * modification, is permitted provided that, in addition to the CDDL 1.0
  36  * License requirements, the following conditions are met:
  37  *
  38  *    Neither the name of the author nor the names of its contributors may be
  39  *    used to endorse or promote products derived from this software without
  40  *    specific prior written permission.
  41  *
  42  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
  43  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
  44  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
  45  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
  46  * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
  47  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
  48  * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
  49  * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
  50  * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
  51  * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
  52  * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
  53  * DAMAGE.
  54  */

  56 /*
  57  * mptsas_impl - This file contains all the basic functions for communicating
  58  * to MPT based hardware.
  59  */

  61 #if defined(lint) || defined(DEBUG)
```



```
  62 #define MPTSAS_DEBUG
  63 #endif

  65 /*
  66  * standard header files
  67  */
  68 #include <sys/note.h>
  69 #include <sys/scsi/scsi.h>
  70 #include <sys/pci.h>

  72 #pragma pack(1)
  73 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_type.h>
  74 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2.h>
  75 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_cnfg.h>
  76 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_init.h>
  77 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_ioc.h>
  78 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_sas.h>
  79 #include <sys/scsi/adapters/mpt_sas/mpi/mpi2_tool.h>
  80 #pragma pack()

  82 /*
  83  * private header files.
  84  */
  85 #include <sys/scsi/adapters/mpt_sas/mptsas_var.h>
  86 #include <sys/scsi/adapters/mpt_sas/mptsas_smhba.h>

  88 /*
  89  * FMA header files.
  90  */
  91 #include <sys/fm/io/ddi.h>

  93 #if defined(MPTSAS_DEBUG)
  94 extern uint32_t mptsas_debug_flags;
  95 #endif

  97 /*
  98  *  prototypes
  99  */
 100 static void mptsas_ioc_event_cmdq_add(mptsas_t *mpt, m_event_struct_t *cmd);
 101 static void mptsas_ioc_event_cmdq_delete(mptsas_t *mpt, m_event_struct_t *cmd);
 102 static m_event_struct_t *mptsas_ioc_event_find_by_cmd(mptsas_t *mpt,
 103     struct mptsas_cmd *cmd);

 105 /*
 106  * add ioc evnet cmd into the queue
 107  */
 108 static void
 109 mptsas_ioc_event_cmdq_add(mptsas_t *mpt, m_event_struct_t *cmd)
 110 {
 111         if ((cmd->m_event_linkp = mpt->m_ioc_event_cmdq) == NULL) {
 112                 mpt->m_ioc_event_cmdtail = &cmd->m_event_linkp;
 113                 mpt->m_ioc_event_cmdq = cmd;
 114         } else {
 115                 cmd->m_event_linkp = NULL;
 116                 *(mpt->m_ioc_event_cmdtail) = cmd;
 117                 mpt->m_ioc_event_cmdtail = &cmd->m_event_linkp;
 118         }
 119 }
_____unchanged_portion_omitted_

 296 int
 297 mptsas_access_config_page(mptsas_t *mpt, uint8_t action, uint8_t page_type,
 298     uint8_t page_number, uint32_t page_address, int (*callback) (mptsas_t *,
 299     caddr_t, ddi_acc_handle_t, uint16_t, uint32_t, va_list), ...)
 300 {
 301         va_list                 ap;
```

```
302         ddi_dma_attr_t          attrs;
303         ddi_dma_cookie_t        cookie;
304         ddi_acc_handle_t        accessp;
305         size_t                  len = 0;
306         mptsas_config_request_t config;
307         int                     rval = DDI_SUCCESS, config_flags = 0;
308         mptsas_cmd_t            *cmd;
309         struct scsi_pkt         *pkt;
310         pMpi2ConfigReply_t      reply;
311         uint16_t                iocstatus = 0;
312         uint32_t                iocloginfo;
313         caddr_t                 page_memp;
314         boolean_t               free_dma = B_FALSE;

316         va_start(ap, callback);
317         ASSERT(mutex_owned(&mpt->m_mutex));

319         /*
320          * Get a command from the pool.
321          */
322         if ((rval = (mptsas_request_from_pool(mpt, &cmd, &pkt))) == -1) {
323                 mptsas_log(mpt, CE_NOTE, "command pool is full for config "
324                     "page request");
325                 rval = DDI_FAILURE;
326                 goto page_done;
327         }
328         config_flags |= MPTSAS_REQUEST_POOL_CMD;

330         bzero((caddr_t)cmd, sizeof (*cmd));
331         bzero((caddr_t)pkt, scsi_pkt_size());
332         bzero((caddr_t)&config, sizeof (config));

334         /*
335          * Save the data for this request to be used in the call to start the
336          * config header request.
337          */
338         config.action = MPI2_CONFIG_ACTION_PAGE_HEADER;
339         config.page_type = page_type;
340         config.page_number = page_number;
341         config.page_address = page_address;

343         /*
344          * Form a blank cmd/pkt to store the acknowledgement message
345          */
346         pkt->pkt_ha_private     = (opaque_t)&config;
347         pkt->pkt_flags          = FLAG_HEAD;
348         pkt->pkt_time           = 60;
349         cmd->cmd_pkt            = pkt;
350         cmd->cmd_flags          = CFLAG_CMDIOC | CFLAG_CONFIG;

352         /*
353          * Save the config header request message in a slot.
354          */
355         if (mptsas_save_cmd(mpt, cmd) == TRUE) {
356                 cmd->cmd_flags |= CFLAG_PREPARED;
357                 mptsas_start_config_page_access(mpt, cmd);
358         } else {
359                 mptsas_waitq_add(mpt, cmd);
360         }

362         /*
363          * If this is a request for a RAID info page, or any page called during
364          * the RAID info page request, poll because these config page requests
365          * are nested.  Poll to avoid data corruption due to one page's data
366          * overwriting the outer page request's data.  This can happen when
367          * the mutex is released in cv_wait.
```



```
368          */
369         if ((page_type == MPI2_CONFIG_EXTPAGETYPE_RAID_CONFIG) ||
370             (page_type == MPI2_CONFIG_PAGETYPE_RAID_VOLUME) ||
371             (page_type == MPI2_CONFIG_PAGETYPE_RAID_PHYSDISK)) {
372                 (void) mptsas_poll(mpt, cmd, pkt->pkt_time * 1000);
373         } else {
374                 while ((cmd->cmd_flags & CFLAG_FINISHED) == 0) {
375                         cv_wait(&mpt->m_config_cv, &mpt->m_mutex);
376                 }
377         }

379         /*
380          * Check if the header request completed without timing out
381          */
382         if (cmd->cmd_flags & CFLAG_TIMEOUT) {
383                 mptsas_log(mpt, CE_WARN, "config header request timeout");
384                 rval = DDI_FAILURE;
385                 goto page_done;
386         }

388         /*
389          * cmd_rfm points to the reply message if a reply was given.  Check the
390          * IOCStatus to make sure everything went OK with the header request.
391          */
392         if (cmd->cmd_rfm) {
393                 config_flags |= MPTSAS_ADDRESS_REPLY;
394                 (void) ddi_dma_sync(mpt->m_dma_reply_frame_hdl, 0, 0,
395                     DDI_DMA_SYNC_FORCPU);
396                 reply = (pMpi2ConfigReply_t)(mpt->m_reply_frame + (cmd->cmd_rfm
397                     - mpt->m_reply_frame_dma_addr));
398                 config.page_type = ddi_get8(mpt->m_acc_reply_frame_hdl,
399                     &reply->Header.PageType);
400                 config.page_number = ddi_get8(mpt->m_acc_reply_frame_hdl,
401                     &reply->Header.PageNumber);
402                 config.page_length = ddi_get8(mpt->m_acc_reply_frame_hdl,
403                     &reply->Header.PageLength);
404                 config.page_version = ddi_get8(mpt->m_acc_reply_frame_hdl,
405                     &reply->Header.PageVersion);
406                 config.ext_page_type = ddi_get8(mpt->m_acc_reply_frame_hdl,
407                     &reply->ExtPageType);
408                 config.ext_page_length = ddi_get16(mpt->m_acc_reply_frame_hdl,
409                     &reply->ExtPageLength);

411                 iocstatus = ddi_get16(mpt->m_acc_reply_frame_hdl,
412                     &reply->IOCStatus);
413                 iocloginfo = ddi_get32(mpt->m_acc_reply_frame_hdl,
414                     &reply->IOCLogInfo);

416                 if (iocstatus) {
417                         NDBG13(("mptsas_access_config_page header: "
418                             "IOCStatus=0x%x, IOCLogInfo=0x%x", iocstatus,
419                             iocloginfo));
420                         rval = DDI_FAILURE;
421                         goto page_done;
422                 }

424                 if ((config.page_type & MPI2_CONFIG_PAGETYPE_MASK) ==
425                     MPI2_CONFIG_PAGETYPE_EXTENDED)
426                         len = (config.ext_page_length * 4);
427                 else
428                         len = (config.page_length * 4);

430         }

432         if (pkt->pkt_reason == CMD_RESET) {
433                 mptsas_log(mpt, CE_WARN, "ioc reset abort config header "
```

```
434                     "request");
435             rval = DDI_FAILURE;
436             goto page_done;
437         }
438
439         /*
440          * Put the reply frame back on the free queue, increment the free
441          * index, and write the new index to the free index register.  But only
442          * if this reply is an ADDRESS reply.
443          */
444         if (config_flags & MPTSAS_ADDRESS_REPLY) {
445                 ddi_put32(mpt->m_acc_free_queue_hdl,
446                     &((uint32_t *)(void *)mpt->m_free_queue)[mpt->m_free_index],
447                     cmd->cmd_rfm);
448                 (void) ddi_dma_sync(mpt->m_dma_free_queue_hdl, 0, 0,
449                     DDI_DMA_SYNC_FORDEV);
450                 if (++mpt->m_free_index == mpt->m_free_queue_depth) {
451                         mpt->m_free_index = 0;
452                 }
453                 ddi_put32(mpt->m_datap, &mpt->m_reg->ReplyFreeHostIndex,
454                     mpt->m_free_index);
455                 config_flags &= (~MPTSAS_ADDRESS_REPLY);
456         }
457
458         /*
459          * Allocate DMA buffer here.  Store the info regarding this buffer in
460          * the cmd struct so that it can be used for this specific command and
461          * de-allocated after the command completes.  The size of the reply
462          * will not be larger than the reply frame size.
463          */
464         attrs = mpt->m_msg_dma_attr;
465         attrs.dma_attr_sgllen = 1;
466         attrs.dma_attr_granular = (uint32_t)len;
467
468         if (mptsas_dma_addr_create(mpt, attrs,
469             &cmd->cmd_dmahandle, &accessp, &page_memp,
470             len, &cookie) == FALSE) {
471                 rval = DDI_FAILURE;
472                 mptsas_log(mpt, CE_WARN,
473                     "mptsas_dma_addr_create(len=0x%x) failed", (int)len);
474                 goto page_done;
475         }
476         /* NOW we can safely call mptsas_dma_addr_destroy(). */
477         free_dma = B_TRUE;
478
479         cmd->cmd_dma_addr = cookie.dmac_laddress;
480         bzero(page_memp, len);
481
482         /*
483          * Save the data for this request to be used in the call to start the
484          * config page read
485          */
486         config.action = action;
487         config.page_address = page_address;
488
489         /*
490          * Re-use the cmd that was used to get the header.  Reset some of the
491          * values.
492          */
493         bzero((caddr_t)pkt, scsi_pkt_size());
494         pkt->pkt_ha_private     = (opaque_t)&config;
495         pkt->pkt_flags          = FLAG_HEAD;
496         pkt->pkt_time           = 60;
497         cmd->cmd_flags          = CFLAG_PREPARED | CFLAG_CMDIOC | CFLAG_CONFIG;
498
499         /*
```



```
500          * Send the config page request.  cmd is re-used from header request.
501          */
502         mptsas_start_config_page_access(mpt, cmd);
503
504         /*
505          * If this is a request for a RAID info page, or any page called during
506          * the RAID info page request, poll because these config page requests
507          * are nested.  Poll to avoid data corruption due to one page's data
508          * overwriting the outer page request's data.  This can happen when
509          * the mutex is released in cv_wait.
510          */
511         if ((page_type == MPI2_CONFIG_EXTPAGETYPE_RAID_CONFIG) ||
512             (page_type == MPI2_CONFIG_PAGETYPE_RAID_VOLUME) ||
513             (page_type == MPI2_CONFIG_PAGETYPE_RAID_PHYSDISK)) {
514                 (void) mptsas_poll(mpt, cmd, pkt->pkt_time * 1000);
515         } else {
516                 while ((cmd->cmd_flags & CFLAG_FINISHED) == 0) {
517                         cv_wait(&mpt->m_config_cv, &mpt->m_mutex);
518                 }
519         }
520
521         /*
522          * Check if the request completed without timing out
523          */
524         if (cmd->cmd_flags & CFLAG_TIMEOUT) {
525                 mptsas_log(mpt, CE_WARN, "config page request timeout");
526                 rval = DDI_FAILURE;
527                 goto page_done;
528         }
529
530         /*
531          * cmd_rfm points to the reply message if a reply was given.  The reply
532          * frame and the config page are returned from this function in the
533          * param list.
534          */
535         if (cmd->cmd_rfm) {
536                 config_flags |= MPTSAS_ADDRESS_REPLY;
537                 (void) ddi_dma_sync(mpt->m_dma_reply_frame_hdl, 0, 0,
538                     DDI_DMA_SYNC_FORCPU);
539                 (void) ddi_dma_sync(cmd->cmd_dmahandle, 0, 0,
540                     DDI_DMA_SYNC_FORCPU);
541                 reply = (pMpi2ConfigReply_t)(mpt->m_reply_frame + (cmd->cmd_rfm
542                     - mpt->m_reply_frame_dma_addr));
543                 iocstatus = ddi_get16(mpt->m_acc_reply_frame_hdl,
544                     &reply->IOCStatus);
545                 iocstatus = MPTSAS_IOCSTATUS(iocstatus);
546                 iocloginfo = ddi_get32(mpt->m_acc_reply_frame_hdl,
547                     &reply->IOCLogInfo);
548         }
549
550         if (callback(mpt, page_memp, accessp, iocstatus, iocloginfo, ap)) {
551                 rval = DDI_FAILURE;
552                 goto page_done;
553         }
554
555         mptsas_fma_check(mpt, cmd);
556         /*
557          * Check the DMA/ACC handles and then free the DMA buffer.
558          */
559         if ((mptsas_check_dma_handle(cmd->cmd_dmahandle) != DDI_SUCCESS) ||
560             (mptsas_check_acc_handle(accessp) != DDI_SUCCESS)) {
561                 ddi_fm_service_impact(mpt->m_dip, DDI_SERVICE_UNAFFECTED);
562                 rval = DDI_FAILURE;
563         }
564
565         if (pkt->pkt_reason == CMD_TRAN_ERR) {
```

```
 566                 mptsas_log(mpt, CE_WARN, "config fma error");
 567                 rval = DDI_FAILURE;
 568                 goto page_done;
 569         }
 570         if (pkt->pkt_reason == CMD_RESET) {
 571                 mptsas_log(mpt, CE_WARN, "ioc reset abort config request");
 572                 rval = DDI_FAILURE;
 573                 goto page_done;
 574         }

 576 page_done:
 577         va_end(ap);
 578         /*
 579          * Put the reply frame back on the free queue, increment the free
 580          * index, and write the new index to the free index register.  But only
 581          * if this reply is an ADDRESS reply.
 582          */
 583         if (config_flags & MPTSAS_ADDRESS_REPLY) {
 584                 ddi_put32(mpt->m_acc_free_queue_hdl,
 585                     &((uint32_t *)(void *)mpt->m_free_queue)[mpt->m_free_index],
 586                     cmd->cmd_rfm);
 587                 (void) ddi_dma_sync(mpt->m_dma_free_queue_hdl, 0, 0,
 588                     DDI_DMA_SYNC_FORDEV);
 589                 if (++mpt->m_free_index == mpt->m_free_queue_depth) {
 590                         mpt->m_free_index = 0;
 591                 }
 592                 ddi_put32(mpt->m_datap, &mpt->m_reg->ReplyFreeHostIndex,
 593                     mpt->m_free_index);
 594         }

 596         if (free_dma)
 597                 mptsas_dma_addr_destroy(&cmd->cmd_dmahandle, &accessp);

 599         if (cmd && (cmd->cmd_flags & CFLAG_PREPARED)) {
 600                 mptsas_remove_cmd(mpt, cmd);
 601                 config_flags &= (~MPTSAS_REQUEST_POOL_CMD);
 602         }
 603         if (config_flags & MPTSAS_REQUEST_POOL_CMD)
 604                 mptsas_return_to_pool(mpt, cmd);

 606         if (config_flags & MPTSAS_CMD_TIMEOUT) {
 607                 mpt->m_softstate &= ~MPTSAS_SS_MSG_UNIT_RESET;
 608                 if ((mptsas_restart_ioc(mpt)) == DDI_FAILURE) {
 609                         mptsas_log(mpt, CE_WARN, "mptsas_restart_ioc failed");
 610                 }
 611         }

 613         return (rval);
 614 }
```

*unchanged_portion_omitted_*

```
1212 /*
1213  * Complete firmware download frame for v2.0 cards.
1214  */
1215 static void
1216 mptsas_uflash2(pMpi2FWDownloadRequest fwdownload,
1217     ddi_acc_handle_t acc_hdl, uint32_t size, uint8_t type,
1218     ddi_dma_cookie_t flsh_cookie)
1219 {
1220         pMpi2FWDownloadTCSGE_t  tcsge;
1221         pMpi2SGESimple64_t      sge;
1222         uint32_t                flagslength;

1224         ddi_put8(acc_hdl, &fwdownload->Function,
1225             MPI2_FUNCTION_FW_DOWNLOAD);
1226         ddi_put8(acc_hdl, &fwdownload->ImageType, type);
```



```
1227         ddi_put8(acc_hdl, &fwdownload->MsgFlags,
1228             MPI2_FW_DOWNLOAD_MSGFLGS_LAST_SEGMENT);
1229         ddi_put32(acc_hdl, &fwdownload->TotalImageSize, size);

1231         tcsge = (pMpi2FWDownloadTCSGE_t)&fwdownload->SGL;
1232         ddi_put8(acc_hdl, &tcsge->ContextSize, 0);
1233         ddi_put8(acc_hdl, &tcsge->DetailsLength, 12);
1234         ddi_put8(acc_hdl, &tcsge->Flags, 0);
1235         ddi_put32(acc_hdl, &tcsge->ImageOffset, 0);
1236         ddi_put32(acc_hdl, &tcsge->ImageSize, size);

1238         sge = (pMpi2SGESimple64_t)(tcsge + 1);
1239         flagslength = size;
1240         flagslength |= ((uint32_t)(MPI2_SGE_FLAGS_LAST_ELEMENT |
1241             MPI2_SGE_FLAGS_END_OF_BUFFER |
1242             MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
1243             MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
1244             MPI2_SGE_FLAGS_64_BIT_ADDRESSING |
1245             MPI2_SGE_FLAGS_HOST_TO_IOC |
1246             MPI2_SGE_FLAGS_END_OF_LIST) << MPI2_SGE_FLAGS_SHIFT);
1247         ddi_put32(acc_hdl, &sge->FlagsLength, flagslength);
1248         ddi_put32(acc_hdl, &sge->Address.Low,
1249             flsh_cookie.dmac_address);
1250         ddi_put32(acc_hdl, &sge->Address.High,
1251             (uint32_t)(flsh_cookie.dmac_laddress >> 32));
1252 }

1254 /*
1255  * Complete firmware download frame for v2.0 cards.
1256  */
1257 static void
1258 mptsas_uflash25(pMpi25FWDownloadRequest fwdownload,
1259     ddi_acc_handle_t acc_hdl, uint32_t size, uint8_t type,
1260     ddi_dma_cookie_t flsh_cookie)
1261 {
1262         pMpi2IeeeSgeSimple64_t  sge;
1263         uint8_t                 flags;

1265         ddi_put8(acc_hdl, &fwdownload->Function,
1266             MPI2_FUNCTION_FW_DOWNLOAD);
1267         ddi_put8(acc_hdl, &fwdownload->ImageType, type);
1268         ddi_put8(acc_hdl, &fwdownload->MsgFlags,
1269             MPI2_FW_DOWNLOAD_MSGFLGS_LAST_SEGMENT);
1270         ddi_put32(acc_hdl, &fwdownload->TotalImageSize, size);

1272         ddi_put32(acc_hdl, &fwdownload->ImageOffset, 0);
1273         ddi_put32(acc_hdl, &fwdownload->ImageSize, size);

1275         sge = (pMpi2IeeeSgeSimple64_t)&fwdownload->SGL;
1276         flags = MPI2_IEEE_SGE_FLAGS_SIMPLE_ELEMENT |
1277             MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR |
1278             MPI25_IEEE_SGE_FLAGS_END_OF_LIST;
1279         ddi_put8(acc_hdl, &sge->Flags, flags);
1280         ddi_put32(acc_hdl, &sge->Length, size);
1281         ddi_put32(acc_hdl, &sge->Address.Low,
1282             flsh_cookie.dmac_address);
1283         ddi_put32(acc_hdl, &sge->Address.High,
1284             (uint32_t)(flsh_cookie.dmac_laddress >> 32));
1285 }

1287 static int mptsas_enable_mpi25_flashupdate = 0;

1289 int
1290 mptsas_update_flash(mptsas_t *mpt, caddr_t ptrbuffer, uint32_t size,
1291     uint8_t type, int mode)
1292 {
```

```
1294         /*
1295          * In order to avoid allocating variables on the stack,
1296          * we make use of the pre-existing mptsas_cmd_t and
1297          * scsi_pkt which are included in the mptsas_t which
1298          * is passed to this routine.
1299          */

1301         ddi_dma_attr_t          flsh_dma_attrs;
1302         ddi_dma_cookie_t        flsh_cookie;
1303         ddi_dma_handle_t        flsh_dma_handle;
1304         ddi_acc_handle_t        flsh_accessp;
1305         caddr_t                 memp, flsh_memp;
1226         uint32_t                flagslength;
1227         pMpi2FWDownloadRequest  fwdownload;
1228         pMpi2FWDownloadTCSGE_t  tcsge;
1229         pMpi2SGESimple64_t      sge;
1306         mptsas_cmd_t            *cmd;
1307         struct scsi_pkt         *pkt;
1308         int                     i;
1309         int                     rvalue = 0;
1310         uint32_t                request_desc_low;

1312         if (mpt->m_MPI25) {
1313                 /*
1314                  * The code is there but not tested yet.
1315                  * User has to know there are risks here.
1316                  */
1317                 mptsas_log(mpt, CE_WARN, "mptsas_update_flash(): "
1318                     "Updating firmware through MPI 2.5 has not been "
1319                     "tested yet!\n"
1320                     "To enable set mptsas_enable_mpi25_flashupdate to 1\n");
1321                 if (!mptsas_enable_mpi25_flashupdate)
1322                         return (-1);
1323         }

1325         if ((rvalue = (mptsas_request_from_pool(mpt, &cmd, &pkt))) == -1) {
1326                 mptsas_log(mpt, CE_WARN, "mptsas_update_flash(): allocation "
1327                     "failed. event ack command pool is full\n");
1328                 return (rvalue);
1329         }

1331         bzero((caddr_t)cmd, sizeof (*cmd));
1332         bzero((caddr_t)pkt, scsi_pkt_size());
1333         cmd->ioc_cmd_slot = (uint32_t)rvalue;

1335         /*
1336          * dynamically create a customized dma attribute structure
1337          * that describes the flash file.
1338          */
1339         flsh_dma_attrs = mpt->m_msg_dma_attr;
1340         flsh_dma_attrs.dma_attr_sgllen = 1;

1342         if (mptsas_dma_addr_create(mpt, flsh_dma_attrs, &flsh_dma_handle,
1343             &flsh_accessp, &flsh_memp, size, &flsh_cookie) == FALSE) {
1344                 mptsas_log(mpt, CE_WARN,
1345                     "(unable to allocate dma resource.");
1346                 mptsas_return_to_pool(mpt, cmd);
1347                 return (-1);
1348         }

1350         bzero(flsh_memp, size);

1352         for (i = 0; i < size; i++) {
1353                 (void) ddi_copyin(ptrbuffer + i, flsh_memp + i, 1, mode);
1354         }
```



```
1355         (void) ddi_dma_sync(flsh_dma_handle, 0, 0, DDI_DMA_SYNC_FORDEV);

1357         /*
1358          * form a cmd/pkt to store the fw download message
1359          */
1360         pkt->pkt_cdbp           = (opaque_t)&cmd->cmd_cdb[0];
1361         pkt->pkt_scbp           = (opaque_t)&cmd->cmd_scb;
1362         pkt->pkt_ha_private     = (opaque_t)cmd;
1363         pkt->pkt_flags          = FLAG_HEAD;
1364         pkt->pkt_time           = 60;
1365         cmd->cmd_pkt            = pkt;
1366         cmd->cmd_scblen         = 1;
1367         cmd->cmd_flags          = CFLAG_CMDIOC | CFLAG_FW_CMD;

1369         /*
1370          * Save the command in a slot
1371          */
1372         if (mptsas_save_cmd(mpt, cmd) == FALSE) {
1373                 mptsas_dma_addr_destroy(&flsh_dma_handle, &flsh_accessp);
1374                 mptsas_return_to_pool(mpt, cmd);
1375                 return (-1);
1376         }

1378         /*
1379          * Fill in fw download message
1380          */
1381         ASSERT(cmd->cmd_slot != 0);
1382         memp = mpt->m_req_frame + (mpt->m_req_frame_size * cmd->cmd_slot);
1383         bzero(memp, mpt->m_req_frame_size);
1295         fwdownload = (void *)memp;
1296         ddi_put8(mpt->m_acc_req_frame_hdl, &fwdownload->Function,
1297             MPI2_FUNCTION_FW_DOWNLOAD);
1298         ddi_put8(mpt->m_acc_req_frame_hdl, &fwdownload->ImageType, type);
1299         ddi_put8(mpt->m_acc_req_frame_hdl, &fwdownload->MsgFlags,
1300             MPI2_FW_DOWNLOAD_MSGFLGS_LAST_SEGMENT);
1301         ddi_put32(mpt->m_acc_req_frame_hdl, &fwdownload->TotalImageSize, size);

1385         if (mpt->m_MPI25)
1386                 mptsas_uflash2((pMpi2FWDownloadRequest)memp,
1387                     mpt->m_acc_req_frame_hdl, size, type, flsh_cookie);
1388         else
1389                 mptsas_uflash25((pMpi25FWDownloadRequest)memp,
1390                     mpt->m_acc_req_frame_hdl, size, type, flsh_cookie);
1303         tcsge = (pMpi2FWDownloadTCSGE_t)&fwdownload->SGL;
1304         ddi_put8(mpt->m_acc_req_frame_hdl, &tcsge->ContextSize, 0);
1305         ddi_put8(mpt->m_acc_req_frame_hdl, &tcsge->DetailsLength, 12);
1306         ddi_put8(mpt->m_acc_req_frame_hdl, &tcsge->Flags, 0);
1307         ddi_put32(mpt->m_acc_req_frame_hdl, &tcsge->ImageOffset, 0);
1308         ddi_put32(mpt->m_acc_req_frame_hdl, &tcsge->ImageSize, size);

1310         sge = (pMpi2SGESimple64_t)(tcsge + 1);
1311         flagslength = size;
1312         flagslength |= ((uint32_t)(MPI2_SGE_FLAGS_LAST_ELEMENT |
1313             MPI2_SGE_FLAGS_END_OF_BUFFER |
1314             MPI2_SGE_FLAGS_SIMPLE_ELEMENT |
1315             MPI2_SGE_FLAGS_SYSTEM_ADDRESS |
1316             MPI2_SGE_FLAGS_64_BIT_ADDRESSING |
1317             MPI2_SGE_FLAGS_HOST_TO_IOC |
1318             MPI2_SGE_FLAGS_END_OF_LIST) << MPI2_SGE_FLAGS_SHIFT);
1319         ddi_put32(mpt->m_acc_req_frame_hdl, &sge->FlagsLength, flagslength);
1320         ddi_put32(mpt->m_acc_req_frame_hdl, &sge->Address.Low,
1321             flsh_cookie.dmac_address);
1322         ddi_put32(mpt->m_acc_req_frame_hdl, &sge->Address.High,
1323             (uint32_t)(flsh_cookie.dmac_laddress >> 32));

1392         /*
```

```
1393          * Start command
1394          */
1395         (void) ddi_dma_sync(mpt->m_dma_req_frame_hdl, 0, 0,
1396             DDI_DMA_SYNC_FORDEV);
1397         request_desc_low = (cmd->cmd_slot << 16) +
1398             MPI2_REQ_DESCRIPT_FLAGS_DEFAULT_TYPE;
1399         cmd->cmd_rfm = NULL;
1400         MPTSAS_START_CMD(mpt, request_desc_low, 0);

1402         rvalue = 0;
1403         (void) cv_reltimedwait(&mpt->m_fw_cv, &mpt->m_mutex,
1404             drv_usectohz(60 * MICROSEC), TR_CLOCK_TICK);
1405         if (!(cmd->cmd_flags & CFLAG_FINISHED)) {
1406                 mpt->m_softstate &= ~MPTSAS_SS_MSG_UNIT_RESET;
1407                 if ((mptsas_restart_ioc(mpt)) == DDI_FAILURE) {
1408                         mptsas_log(mpt, CE_WARN, "mptsas_restart_ioc failed");
1409                 }
1410                 rvalue = -1;
1411         }
1412         mptsas_remove_cmd(mpt, cmd);
1413         mptsas_dma_addr_destroy(&flsh_dma_handle, &flsh_accessp);

1415         return (rvalue);
1416 }

1418 static int
1419 mptsas_sasdevpage_0_cb(mptsas_t *mpt, caddr_t page_memp,
1420     ddi_acc_handle_t accessp, uint16_t iocstatus, uint32_t iocloginfo,
1421     va_list ap)
1422 {
1423 #ifndef __lock_lint
1424         _NOTE(ARGUNUSED(ap))
1425 #endif
1426         pMpi2SasDevicePage0_t   sasdevpage;
1427         int                     rval = DDI_SUCCESS, i;
1428         uint8_t                 *sas_addr = NULL;
1429         uint8_t                 tmp_sas_wwn[SAS_WWN_BYTE_SIZE];
1430         uint16_t                *devhdl, *bay_num, *enclosure;
1431         uint64_t                *sas_wwn;
1432         uint32_t                *dev_info;
1433         uint8_t                 *physport, *phynum;
1434         uint16_t                *pdevhdl, io_flags;
1367         uint16_t                *pdevhdl;
1435         uint32_t                page_address;

1437         if ((iocstatus != MPI2_IOCSTATUS_SUCCESS) &&
1438             (iocstatus != MPI2_IOCSTATUS_CONFIG_INVALID_PAGE)) {
1439                 mptsas_log(mpt, CE_WARN, "mptsas_get_sas_device_page0 "
1440                     "header: IOCStatus=0x%x, IOCLogInfo=0x%x",
1441                     iocstatus, iocloginfo);
1442                 rval = DDI_FAILURE;
1443                 return (rval);
1444         }
1445         page_address = va_arg(ap, uint32_t);
1446         /*
1447          * The INVALID_PAGE status is normal if using GET_NEXT_HANDLE and there
1448          * are no more pages.  If everything is OK up to this point but the
1449          * status is INVALID_PAGE, change rval to FAILURE and quit.  Also,
1450          * signal that device traversal is complete.
1451          */
1452         if (iocstatus == MPI2_IOCSTATUS_CONFIG_INVALID_PAGE) {
1453                 if ((page_address & MPI2_SAS_DEVICE_PGAD_FORM_MASK) ==
1454                     MPI2_SAS_DEVICE_PGAD_FORM_GET_NEXT_HANDLE) {
1455                         mpt->m_done_traverse_dev = 1;
1456                 }
1457                 rval = DDI_FAILURE;
```



```
1458                 return (rval);
1459         }
1460         devhdl = va_arg(ap, uint16_t *);
1461         sas_wwn = va_arg(ap, uint64_t *);
1462         dev_info = va_arg(ap, uint32_t *);
1463         physport = va_arg(ap, uint8_t *);
1464         phynum = va_arg(ap, uint8_t *);
1465         pdevhdl = va_arg(ap, uint16_t *);
1466         bay_num = va_arg(ap, uint16_t *);
1467         enclosure = va_arg(ap, uint16_t *);


1470         sasdevpage = (pMpi2SasDevicePage0_t)page_memp;

1472         *dev_info = ddi_get32(accessp, &sasdevpage->DeviceInfo);
1473         *devhdl = ddi_get16(accessp, &sasdevpage->DevHandle);
1474         sas_addr = (uint8_t *)(&sasdevpage->SASAddress);
1475         for (i = 0; i < SAS_WWN_BYTE_SIZE; i++) {
1476                 tmp_sas_wwn[i] = ddi_get8(accessp, sas_addr + i);
1477         }
1478         bcopy(tmp_sas_wwn, sas_wwn, SAS_WWN_BYTE_SIZE);
1479         *sas_wwn = LE_64(*sas_wwn);
1480         *physport = ddi_get8(accessp, &sasdevpage->PhysicalPort);
1481         *phynum = ddi_get8(accessp, &sasdevpage->PhyNum);
1482         *pdevhdl = ddi_get16(accessp, &sasdevpage->ParentDevHandle);
1483         *bay_num = ddi_get16(accessp, &sasdevpage->Slot);
1484         *enclosure = ddi_get16(accessp, &sasdevpage->EnclosureHandle);

1486         /*
1487          * This is where we would check the flag
1488          * MPI25_SAS_DEVICE0_FLAGS_FAST_PATH_CAPABLE
1489          * and set something that will allow us to use fastpath during
1490          * target transfers.
1491          */
1492         io_flags = ddi_get16(accessp, &sasdevpage->Flags);
1493         if (io_flags & MPI25_SAS_DEVICE0_FLAGS_FAST_PATH_CAPABLE) {
1494                 /*
1495                  * uint8_t  *fast_path;
1496                  * fast_path = va_arg(ap, uint8_t *);
1497                  * *fast_path = MPI25_REQ_DESCRIPT_FLAGS_FAST_PATH_SCSI_IO;
1498                  *
1499                  * Need to change all calls to mptsas_get_sas_device_page0()
1500                  * to include another argument to collect this. And then what
1501                  * do we do with it?
1502                  * XXX For now print a message..
1503                  */
1504                 mptsas_log(mpt, CE_CONT,
1505                     "!Found MPI25_SAS_DEVICE0_FLAGS_FAST_"
1506                     "PATH_CAPABLE for %llx", (long long)*sas_wwn);
1507         }

1509         return (rval);
1510 }
______unchanged_portion_omitted_
```

```
**********************************************************
   19531 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/io/scsi/adapters/mpt_sas/mptsas_init.c
NEX-1889 upstream
**********************************************************
```

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

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright (c) 2014, Tegile Systems Inc. All rights reserved.
 */

/*
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 * mptsas_init - This file contains all the functions used to initialize
 * MPT2.0 based hardware.
 */

#if defined(lint) || defined(DEBUG)
#define MPTSAS_DEBUG
```



```
#endif

/*
 * standard header files
 */
#include <sys/note.h>
#include <sys/scsi/scsi.h>

#pragma pack(1)
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_type.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_cnfg.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_init.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_ioc.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_tool.h>
#pragma pack()
/*
 * private header files.
 */
#include <sys/scsi/adapters/mpt_sas/mptsas_var.h>

static int mptsas_ioc_do_get_facts(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_ioc_do_get_facts_reply(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_ioc_do_get_port_facts(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_ioc_do_get_port_facts_reply(mptsas_t *mpt, caddr_t memp,
    int var, ddi_acc_handle_t accessp);
static int mptsas_ioc_do_enable_port(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_ioc_do_enable_port_reply(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_ioc_do_enable_event_notification(mptsas_t *mpt, caddr_t memp,
        int var, ddi_acc_handle_t accessp);
static int mptsas_ioc_do_enable_event_notification_reply(mptsas_t *mpt,
    caddr_t memp, int var, ddi_acc_handle_t accessp);
static int mptsas_do_ioc_init(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);
static int mptsas_do_ioc_init_reply(mptsas_t *mpt, caddr_t memp, int var,
        ddi_acc_handle_t accessp);

static const char *
mptsas_product_type_string(mptsas_t *mpt)
{
        switch (mpt->m_productid & MPI2_FW_HEADER_PID_PROD_MASK) {

        case MPI2_FW_HEADER_PID_PROD_A:
                return ("A");
        default:
                return ("?");
        }
}
```

_____*unchanged_portion_omitted_*

```
static int
mptsas_ioc_do_get_facts_reply(mptsas_t *mpt, caddr_t memp, int var,
                ddi_acc_handle_t accessp)
{
#ifndef __lock_lint
        _NOTE(ARGUNUSED(var))
#endif

        pMpi2IOCFactsReply_t    factsreply;
        int                     numbytes;
        uint_t                  iocstatus;
```

```
174         char                    buf[32];
175         uint16_t                numReplyFrames;
176         uint16_t                queueSize, queueDiff;
177         int                     simple_sge_main;
178         int                     simple_sge_next;
179         uint32_t                capabilities;
180         uint16_t                msgversion;

182         bzero(memp, sizeof (*factsreply));
183         factsreply = (void *)memp;
184         numbytes = sizeof (*factsreply);

186         /*
187          * get ioc facts reply message
188          */
189         if (mptsas_get_handshake_msg(mpt, memp, numbytes, accessp)) {
190                 return (DDI_FAILURE);
191         }

193         if (iocstatus = ddi_get16(accessp, &factsreply->IOCStatus)) {
194                 mptsas_log(mpt, CE_WARN, "mptsas_ioc_do_get_facts_reply: "
195                     "IOCStatus=0x%x, IOCLogInfo=0x%x", iocstatus,
196                     ddi_get32(accessp, &factsreply->IOCLogInfo));
197                 return (DDI_FAILURE);
198         }

200         /*
201          * store key values from reply to mpt structure
202          */
203         mpt->m_fwversion = ddi_get32(accessp, &factsreply->FWVersion.Word);
204         mpt->m_productid = ddi_get16(accessp, &factsreply->ProductID);


207         (void) sprintf(buf, "%u.%u.%u.%u",
208             ddi_get8(accessp, &factsreply->FWVersion.Struct.Major),
209             ddi_get8(accessp, &factsreply->FWVersion.Struct.Minor),
210             ddi_get8(accessp, &factsreply->FWVersion.Struct.Unit),
211             ddi_get8(accessp, &factsreply->FWVersion.Struct.Dev));
212         mptsas_log(mpt, CE_NOTE, "?mpt%d Firmware version v%s (%s)\n",
213             mpt->m_instance, buf, mptsas_product_type_string(mpt));
214         (void) ddi_prop_update_string(DDI_DEV_T_NONE, mpt->m_dip,
215             "firmware-version", buf);

217         /*
218          * Set up request info.
219          */
220         mpt->m_max_requests = ddi_get16(accessp,
221             &factsreply->RequestCredit) - 1;
222         mpt->m_req_frame_size = ddi_get16(accessp,
223             &factsreply->IOCRequestFrameSize) * 4;

225         /*
226          * Size of reply free queue should be the number of requests
227          * plus some additional for events (32).  Make sure number of
228          * reply frames is not a multiple of 16 so that the queue sizes
229          * are calculated correctly later to be a multiple of 16.
230          */
231         mpt->m_reply_frame_size = ddi_get8(accessp,
232             &factsreply->ReplyFrameSize) * 4;
233         numReplyFrames = mpt->m_max_requests + 32;
234         if (!(numReplyFrames % 16)) {
235                 numReplyFrames--;
236         }
237         mpt->m_max_replies = numReplyFrames;
238         queueSize = numReplyFrames;
239         queueSize += 16 - (queueSize % 16);
```



```
240         mpt->m_free_queue_depth = queueSize;

242         /*
243          * Size of reply descriptor post queue should be the number of
244          * request frames + the number of reply frames + 1 and needs to
245          * be a multiple of 16.  This size can be no larger than
246          * MaxReplyDescriptorPostQueueDepth from IOCFacts.  If the
247          * calculated queue size is larger than allowed, subtract a
248          * multiple of 16 from m_max_requests, m_max_replies, and
249          * m_reply_free_depth.
250          */
251         queueSize = mpt->m_max_requests + numReplyFrames + 1;
252         if (queueSize % 16) {
253                 queueSize += 16 - (queueSize % 16);
254         }
255         mpt->m_post_queue_depth = ddi_get16(accessp,
256             &factsreply->MaxReplyDescriptorPostQueueDepth);
257         if (queueSize > mpt->m_post_queue_depth) {
258                 queueDiff = queueSize - mpt->m_post_queue_depth;
259                 if (queueDiff % 16) {
260                         queueDiff += 16 - (queueDiff % 16);
261                 }
262                 mpt->m_max_requests -= queueDiff;
263                 mpt->m_max_replies -= queueDiff;
264                 mpt->m_free_queue_depth -= queueDiff;
265                 queueSize -= queueDiff;
266         }
267         mpt->m_post_queue_depth = queueSize;

269         /*
270          * Set up max chain depth.
271          */
272         mpt->m_max_chain_depth = ddi_get8(accessp,
273             &factsreply->MaxChainDepth);
274         mpt->m_ioc_capabilities = ddi_get32(accessp,
275             &factsreply->IOCCapabilities);

277         /*
278          * Set flag to check for SAS3 support.
279          */
280         msgversion = ddi_get16(accessp, &factsreply->MsgVersion);
281         if (msgversion == MPI2_VERSION_02_05) {
282                 mptsas_log(mpt, CE_NOTE, "?mpt_sas%d SAS 3 Supported\n",
283                     mpt->m_instance);
284                 mpt->m_MPI25 = TRUE;
285         } else {
286                 mptsas_log(mpt, CE_NOTE, "?mpt%d MPI Version 0x%x\n",
287                     mpt->m_instance, msgversion);
288         }

290         /*
291          * Calculate max frames per request based on DMA S/G length.
292          */
293         simple_sge_main = MPTSAS_MAX_FRAME_SGES64(mpt) - 1;
294         simple_sge_next = mpt->m_req_frame_size /
295             (mpt->m_MPI25 ? sizeof (MPI2_IEEE_SGE_SIMPLE64) :
296             sizeof (MPI2_SGE_SIMPLE64)) - 1;
280             sizeof (MPI2_SGE_SIMPLE64) - 1;

298         mpt->m_max_request_frames = (MPTSAS_MAX_DMA_SEGS -
299             simple_sge_main) / simple_sge_next + 1;
300         if (((MPTSAS_MAX_DMA_SEGS - simple_sge_main) %
301             simple_sge_next) > 1) {
302                 mpt->m_max_request_frames++;
303         }
```

```
305         /*
306          * Check if controller supports FW diag buffers and set flag to enable
307          * each type.
308          */
309         capabilities = ddi_get32(accessp, &factsreply->IOCCapabilities);
310         if (capabilities & MPI2_IOCFACTS_CAPABILITY_DIAG_TRACE_BUFFER) {
311                 mpt->m_fw_diag_buffer_list[MPI2_DIAG_BUF_TYPE_TRACE].enabled =
312                     TRUE;
313         }
314         if (capabilities & MPI2_IOCFACTS_CAPABILITY_SNAPSHOT_BUFFER) {
315                 mpt->m_fw_diag_buffer_list[MPI2_DIAG_BUF_TYPE_SNAPSHOT].
316                     enabled = TRUE;
317         }
318         if (capabilities & MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER) {
319                 mpt->m_fw_diag_buffer_list[MPI2_DIAG_BUF_TYPE_EXTENDED].
320                     enabled = TRUE;
321         }

323         /*
324          * Check if controller supports replaying events when issuing Message
325          * Unit Reset and set flag to enable MUR.
326          */
327         if (capabilities & MPI2_IOCFACTS_CAPABILITY_EVENT_REPLAY) {
328                 mpt->m_event_replay = TRUE;
329         }

331         /*
332          * Check if controller supports IR.
333          */
334         if (capabilities & MPI2_IOCFACTS_CAPABILITY_INTEGRATED_RAID) {
335                 mpt->m_ir_capable = TRUE;
336         }

338         return (DDI_SUCCESS);
339 }
```

unchanged_portion_omitted_

****
1466 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/THIRDPARTYLICENSE
NEX-1888 upstream
****

```
**********
      42 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/THIRDPARTYLICENSE.descrip
NEX-1888 upstream
**********
   1 LSI Fusion-MPT MPI 2.0 / 2.5 Header Files
```

```
**********************************************************
   54260 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2.h
 *         Title:  MPI Message independent structures and definitions
 *                 including System Interface Register Set and
 *                 scatter/gather formats.
 *  Creation Date:  June 21, 2006
 *
 *  mpi2.h Version:  02.00.33
 *  mpi2.h Version:  02.00.13
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-04-07  02.00.01  Bumped MPI2_HEADER_VERSION_UNIT.
 *  06-26-07  02.00.02  Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-31-07  02.00.03  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Moved ReplyPostHostIndex register to offset 0x6C of the
 *                      MPI2_SYSTEM_INTERFACE_REGS and modified the define for
 *                      MPI2_REPLY_POST_HOST_INDEX_OFFSET.
 *                      Added union of request descriptors.
 *                      Added union of reply descriptors.
 *  10-31-07  02.00.04  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added define for MPI2_VERSION_02_00.
 *                      Fixed the size of the FunctionDependent5 field in the
 *                      MPI2_DEFAULT_REPLY structure.
 *  12-18-07  02.00.05  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Removed the MPI-defined Fault Codes and extended the
 *                      product specific codes up to 0xEFFF.
 *                      Added a sixth key value for the WriteSequence register
 *                      and changed the flush value to 0x0.
 *                      Added message function codes for Diagnostic Buffer Post
 *                      and Diagnsotic Release.
 *                      New IOCStatus define: MPI2_IOCSTATUS_DIAGNOSTIC_RELEASED
 *                      Moved MPI2_VERSION_UNION from mpi2_ioc.h.
 *  02-29-08  02.00.06  Bumped MPI2_HEADER_VERSION_UNIT.
 *  03-03-08  02.00.07  Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-21-08  02.00.08  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added #defines for marking a reply descriptor as unused.
 *  06-27-08  02.00.09  Bumped MPI2_HEADER_VERSION_UNIT.
 *  10-02-08  02.00.10  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Moved LUN field defines from mpi2_init.h.
 *  01-19-09  02.00.11  Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-06-09  02.00.12  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      In all request and reply descriptors, replaced VF_ID
 *                      field with MSIxIndex field.
```

```
 *                     Removed DevHandle field from
 *                     MPI2_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR and made those
 *                     bytes reserved.
 *                     Added RAID Accelerator functionality.
 *  07-30-09  02.00.13 Bumped MPI2_HEADER_VERSION_UNIT.
 *  10-28-09  02.00.14 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added MSI-x index mask and shift for Reply Post Host
 *                     Index register.
 *                     Added function code for Host Based Discovery Action.
 *  02-10-10  02.00.15 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added define for MPI2_FUNCTION_PWR_MGMT_CONTROL.
 *                     Added defines for product-specific range of message
 *                     function codes, 0xF0 to 0xFF.
 *  05-12-10  02.00.16 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added alternative defines for the SGE Direction bit.
 *  08-11-10  02.00.17 Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-10-10  02.00.18 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added MPI2_IEEE_SGE_FLAGS_SYSTEMPLBCPI_ADDR define.
 *  02-23-11  02.00.19 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added MPI2_FUNCTION_SEND_HOST_MESSAGE.
 *  03-09-11  02.00.20 Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-25-11  02.00.21 Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-24-11  02.00.22 Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-18-11  02.00.23 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Incorporating additions for MPI v2.5.
 *  02-06-12  02.00.24 Bumped MPI2_HEADER_VERSION_UNIT.
 *  03-29-12  02.00.25 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added Hard Reset delay timings.
 *  07-10-12  02.00.26 Bumped MPI2_HEADER_VERSION_UNIT.
 *  07-26-12  02.00.27 Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-27-12  02.00.28 Bumped MPI2_HEADER_VERSION_UNIT.
 *  12-20-12  02.00.29 Bumped MPI2_HEADER_VERSION_UNIT.
 *                     Added MPI25_SUP_REPLY_POST_HOST_INDEX_OFFSET.
 *  04-09-13  02.00.30 Bumped MPI2_HEADER_VERSION_UNIT.
 *  04-17-13  02.00.31 Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-19-13  02.00.32 Bumped MPI2_HEADER_VERSION_UNIT.
 *  12-05-13  02.00.33 Bumped MPI2_HEADER_VERSION_UNIT.
 *  ------------------------------------------------------------------------
 */

#ifndef MPI2_H
#define MPI2_H


/*****************************************************************************
*
*        MPI Version Definitions
*
*****************************************************************************/

#define MPI2_VERSION_MAJOR                  (0x02)
#define MPI2_VERSION_MINOR                  (0x00)
#define MPI2_VERSION_MAJOR_MASK             (0xFF00)
#define MPI2_VERSION_MAJOR_SHIFT            (8)
#define MPI2_VERSION_MINOR_MASK             (0x00FF)
#define MPI2_VERSION_MINOR_SHIFT            (0)

/* major version for all MPI v2.x */
#define MPI2_VERSION_MAJOR                  (0x02)

/* minor version for MPI v2.0 compatible products */
#define MPI2_VERSION_MINOR                  (0x00)
#define MPI2_VERSION ((MPI2_VERSION_MAJOR << MPI2_VERSION_MAJOR_SHIFT) |   \
                                    MPI2_VERSION_MINOR)

#define MPI2_VERSION_02_00                  (0x0200)
```



```
/* minor version for MPI v2.5 compatible products */
#define MPI25_VERSION_MINOR                 (0x05)
#define MPI25_VERSION ((MPI2_VERSION_MAJOR << MPI2_VERSION_MAJOR_SHIFT) |   \
                                          MPI25_VERSION_MINOR)
#define MPI2_VERSION_02_05                  (0x0205)


/* Unit and Dev versioning for this MPI header set */
#define MPI2_HEADER_VERSION_UNIT            (0x21)
/* versioning for this MPI header set */
#define MPI2_HEADER_VERSION_UNIT            (0x0D)
#define MPI2_HEADER_VERSION_DEV             (0x00)
#define MPI2_HEADER_VERSION_UNIT_MASK       (0xFF00)
#define MPI2_HEADER_VERSION_UNIT_SHIFT      (8)
#define MPI2_HEADER_VERSION_DEV_MASK        (0x00FF)
#define MPI2_HEADER_VERSION_DEV_SHIFT       (0)
#define MPI2_HEADER_VERSION ((MPI2_HEADER_VERSION_UNIT << 8) | MPI2_HEADER_VERSI


/*****************************************************************************
*
*        IOC State Definitions
*
*****************************************************************************/

#define MPI2_IOC_STATE_RESET                (0x00000000)
#define MPI2_IOC_STATE_READY                (0x10000000)
#define MPI2_IOC_STATE_OPERATIONAL          (0x20000000)
#define MPI2_IOC_STATE_FAULT                (0x40000000)

#define MPI2_IOC_STATE_MASK                 (0xF0000000)
#define MPI2_IOC_STATE_SHIFT                (28)

/* Fault state range for prodcut specific codes */
#define MPI2_FAULT_PRODUCT_SPECIFIC_MIN                 (0x0000)
#define MPI2_FAULT_PRODUCT_SPECIFIC_MAX                 (0xEFFF)


/*****************************************************************************
*
*        System Interface Register Definitions
*
*****************************************************************************/

typedef volatile struct _MPI2_SYSTEM_INTERFACE_REGS
{
    U32         Doorbell;                   /* 0x00 */
    U32         WriteSequence;              /* 0x04 */
    U32         HostDiagnostic;             /* 0x08 */
    U32         Reserved1;                  /* 0x0C */
    U32         DiagRWData;                 /* 0x10 */
    U32         DiagRWAddressLow;           /* 0x14 */
    U32         DiagRWAddressHigh;          /* 0x18 */
    U32         Reserved2[5];               /* 0x1C */
    U32         HostInterruptStatus;        /* 0x30 */
    U32         HostInterruptMask;          /* 0x34 */
    U32         DCRData;                    /* 0x38 */
    U32         DCRAddress;                 /* 0x3C */
    U32         Reserved3[2];               /* 0x40 */
    U32         ReplyFreeHostIndex;         /* 0x48 */
    U32         Reserved4[8];               /* 0x4C */
    U32         ReplyPostHostIndex;         /* 0x6C */
    U32         Reserved5;                  /* 0x70 */
    U32         HCBSize;                    /* 0x74 */
```

```
212     U32         HCBAddressLow;              /* 0x78 */
213     U32         HCBAddressHigh;             /* 0x7C */
214     U32         Reserved6[16];              /* 0x80 */
215     U32         RequestDescriptorPostLow;   /* 0xC0 */
216     U32         RequestDescriptorPostHigh;  /* 0xC4 */
217     U32         Reserved7[14];              /* 0xC8 */
218 } MPI2_SYSTEM_INTERFACE_REGS, MPI2_POINTER PTR_MPI2_SYSTEM_INTERFACE_REGS,
219   Mpi2SystemInterfaceRegs_t, MPI2_POINTER pMpi2SystemInterfaceRegs_t;

221 /*
222  * Defines for working with the Doorbell register.
223  */
224 #define MPI2_DOORBELL_OFFSET                    (0x00000000)

226 /* IOC --> System values */
227 #define MPI2_DOORBELL_USED                      (0x08000000)
228 #define MPI2_DOORBELL_WHO_INIT_MASK             (0x07000000)
229 #define MPI2_DOORBELL_WHO_INIT_SHIFT            (24)
230 #define MPI2_DOORBELL_FAULT_CODE_MASK           (0x0000FFFF)
231 #define MPI2_DOORBELL_DATA_MASK                 (0x0000FFFF)

233 /* System --> IOC values */
234 #define MPI2_DOORBELL_FUNCTION_MASK             (0xFF000000)
235 #define MPI2_DOORBELL_FUNCTION_SHIFT            (24)
236 #define MPI2_DOORBELL_ADD_DWORDS_MASK           (0x00FF0000)
237 #define MPI2_DOORBELL_ADD_DWORDS_SHIFT          (16)


240 /*
241  * Defines for the WriteSequence register
242  */
243 #define MPI2_WRITE_SEQUENCE_OFFSET              (0x00000004)
244 #define MPI2_WRSEQ_KEY_VALUE_MASK               (0x0000000F)
245 #define MPI2_WRSEQ_FLUSH_KEY_VALUE              (0x0)
246 #define MPI2_WRSEQ_1ST_KEY_VALUE                (0xF)
247 #define MPI2_WRSEQ_2ND_KEY_VALUE                (0x4)
248 #define MPI2_WRSEQ_3RD_KEY_VALUE                (0xB)
249 #define MPI2_WRSEQ_4TH_KEY_VALUE                (0x2)
250 #define MPI2_WRSEQ_5TH_KEY_VALUE                (0x7)
251 #define MPI2_WRSEQ_6TH_KEY_VALUE                (0xD)

253 /*
254  * Defines for the HostDiagnostic register
255  */
256 #define MPI2_HOST_DIAGNOSTIC_OFFSET             (0x00000008)

258 #define MPI2_DIAG_BOOT_DEVICE_SELECT_MASK       (0x00001800)
259 #define MPI2_DIAG_BOOT_DEVICE_SELECT_DEFAULT    (0x00000000)
260 #define MPI2_DIAG_BOOT_DEVICE_SELECT_HCDW       (0x00000800)

262 #define MPI2_DIAG_CLEAR_FLASH_BAD_SIG           (0x00000400)
263 #define MPI2_DIAG_FORCE_HCB_ON_RESET            (0x00000200)
264 #define MPI2_DIAG_HCB_MODE                      (0x00000100)
265 #define MPI2_DIAG_DIAG_WRITE_ENABLE             (0x00000080)
266 #define MPI2_DIAG_FLASH_BAD_SIG                 (0x00000040)
267 #define MPI2_DIAG_RESET_HISTORY                 (0x00000020)
268 #define MPI2_DIAG_DIAG_RW_ENABLE                (0x00000010)
269 #define MPI2_DIAG_RESET_ADAPTER                 (0x00000004)
270 #define MPI2_DIAG_HOLD_IOC_RESET                (0x00000002)

272 /*
273  * Offsets for DiagRWData and address
274  */
275 #define MPI2_DIAG_RW_DATA_OFFSET                (0x00000010)
276 #define MPI2_DIAG_RW_ADDRESS_LOW_OFFSET         (0x00000014)
277 #define MPI2_DIAG_RW_ADDRESS_HIGH_OFFSET        (0x00000018)
```



```
279 /*
280  * Defines for the HostInterruptStatus register
281  */
282 #define MPI2_HOST_INTERRUPT_STATUS_OFFSET       (0x00000030)
283 #define MPI2_HIS_SYS2IOC_DB_STATUS              (0x80000000)
284 #define MPI2_HIS_IOP_DOORBELL_STATUS            MPI2_HIS_SYS2IOC_DB_STATUS
285 #define MPI2_HIS_RESET_IRQ_STATUS               (0x40000000)
286 #define MPI2_HIS_REPLY_DESCRIPTOR_INTERRUPT     (0x00000008)
287 #define MPI2_HIS_IOC2SYS_DB_STATUS              (0x00000001)
288 #define MPI2_HIS_DOORBELL_INTERRUPT             MPI2_HIS_IOC2SYS_DB_STATUS

290 /*
291  * Defines for the HostInterruptMask register
292  */
293 #define MPI2_HOST_INTERRUPT_MASK_OFFSET         (0x00000034)
294 #define MPI2_HIM_RESET_IRQ_MASK                 (0x40000000)
295 #define MPI2_HIM_REPLY_INT_MASK                 (0x00000008)
296 #define MPI2_HIM_RIM                            MPI2_HIM_REPLY_INT_MASK
297 #define MPI2_HIM_IOC2SYS_DB_MASK                (0x00000001)
298 #define MPI2_HIM_DIM                            MPI2_HIM_IOC2SYS_DB_MASK

300 /*
301  * Offsets for DCRData and address
302  */
303 #define MPI2_DCR_DATA_OFFSET                    (0x00000038)
304 #define MPI2_DCR_ADDRESS_OFFSET                 (0x0000003C)

306 /*
307  * Offset for the Reply Free Queue
308  */
309 #define MPI2_REPLY_FREE_HOST_INDEX_OFFSET       (0x00000048)

311 /*
312  * Defines for the Reply Descriptor Post Queue
280  * Offset for the Reply Descriptor Post Queue
313  */
314 #define MPI2_REPLY_POST_HOST_INDEX_OFFSET       (0x0000006C)
315 #define MPI2_REPLY_POST_HOST_INDEX_MASK         (0x00FFFFFF)
316 #define MPI2_RPHI_MSIX_INDEX_MASK               (0xFF000000)
317 #define MPI2_RPHI_MSIX_INDEX_SHIFT              (24)
318 #define MPI25_SUP_REPLY_POST_HOST_INDEX_OFFSET  (0x0000030C) /* MPI v2.5 only */


321 /*
322  * Defines for the HCBSize and address
323  */
324 #define MPI2_HCB_SIZE_OFFSET                    (0x00000074)
325 #define MPI2_HCB_SIZE_SIZE_MASK                 (0xFFFFF000)
326 #define MPI2_HCB_SIZE_HCB_ENABLE                (0x00000001)

328 #define MPI2_HCB_ADDRESS_LOW_OFFSET             (0x00000078)
329 #define MPI2_HCB_ADDRESS_HIGH_OFFSET            (0x0000007C)

331 /*
332  * Offsets for the Request Queue
333  */
334 #define MPI2_REQUEST_DESCRIPTOR_POST_LOW_OFFSET     (0x000000C0)
335 #define MPI2_REQUEST_DESCRIPTOR_POST_HIGH_OFFSET    (0x000000C4)


338 /* Hard Reset delay timings */
339 #define MPI2_HARD_RESET_PCIE_FIRST_READ_DELAY_MICRO_SEC     (50000)
340 #define MPI2_HARD_RESET_PCIE_RESET_READ_WINDOW_MICRO_SEC    (255000)
341 #define MPI2_HARD_RESET_PCIE_SECOND_READ_DELAY_MICRO_SEC    (256000)
```

```
343 /*****************************************************************************
344 *
345 *        Message Descriptors
346 *
347 *****************************************************************************/

349 /* Request Descriptors */

351 /* Default Request Descriptor */
352 typedef struct _MPI2_DEFAULT_REQUEST_DESCRIPTOR
353 {
354     U8              RequestFlags;               /* 0x00 */
355     U8              MSIxIndex;                  /* 0x01 */
356     U16             SMID;                       /* 0x02 */
357     U16             LMID;                       /* 0x04 */
358     U16             DescriptorTypeDependent;    /* 0x06 */
359 } MPI2_DEFAULT_REQUEST_DESCRIPTOR,
360   MPI2_POINTER PTR_MPI2_DEFAULT_REQUEST_DESCRIPTOR,
361   Mpi2DefaultRequestDescriptor_t, MPI2_POINTER pMpi2DefaultRequestDescriptor_t;

363 /* defines for the RequestFlags field */
364 #define MPI2_REQ_DESCRIPT_FLAGS_TYPE_MASK               (0x0E)
365 #define MPI2_REQ_DESCRIPT_FLAGS_SCSI_IO                 (0x00)
366 #define MPI2_REQ_DESCRIPT_FLAGS_SCSI_TARGET             (0x02)
367 #define MPI2_REQ_DESCRIPT_FLAGS_HIGH_PRIORITY           (0x06)
368 #define MPI2_REQ_DESCRIPT_FLAGS_DEFAULT_TYPE            (0x08)
369 #define MPI2_REQ_DESCRIPT_FLAGS_RAID_ACCELERATOR        (0x0A)
370 #define MPI25_REQ_DESCRIPT_FLAGS_FAST_PATH_SCSI_IO      (0x0C)

372 #define MPI2_REQ_DESCRIPT_FLAGS_IOC_FIFO_MARKER (0x01)


375 /* High Priority Request Descriptor */
376 typedef struct _MPI2_HIGH_PRIORITY_REQUEST_DESCRIPTOR
377 {
378     U8              RequestFlags;               /* 0x00 */
379     U8              MSIxIndex;                  /* 0x01 */
380     U16             SMID;                       /* 0x02 */
381     U16             LMID;                       /* 0x04 */
382     U16             Reserved1;                  /* 0x06 */
383 } MPI2_HIGH_PRIORITY_REQUEST_DESCRIPTOR,
______unchanged_portion_omitted_
425   MPI2_POINTER PTR_MPI2_RAID_ACCEL_REQUEST_DESCRIPTOR,
426   Mpi2RAIDAcceleratorRequestDescriptor_t,
427   MPI2_POINTER pMpi2RAIDAcceleratorRequestDescriptor_t;


430 /* Fast Path SCSI IO Request Descriptor */
431 typedef MPI2_SCSI_IO_REQUEST_DESCRIPTOR
432     MPI25_FP_SCSI_IO_REQUEST_DESCRIPTOR,
433     MPI2_POINTER PTR_MPI25_FP_SCSI_IO_REQUEST_DESCRIPTOR,
434     Mpi25FastPathSCSIIORequestDescriptor_t,
435     MPI2_POINTER pMpi25FastPathSCSIIORequestDescriptor_t;


438 /* union of Request Descriptors */
439 typedef union _MPI2_REQUEST_DESCRIPTOR_UNION
440 {
441     MPI2_DEFAULT_REQUEST_DESCRIPTOR             Default;
442     MPI2_HIGH_PRIORITY_REQUEST_DESCRIPTOR       HighPriority;
443     MPI2_SCSI_IO_REQUEST_DESCRIPTOR             SCSIIO;
444     MPI2_SCSI_TARGET_REQUEST_DESCRIPTOR         SCSITarget;
445     MPI2_RAID_ACCEL_REQUEST_DESCRIPTOR          RAIDAccelerator;
446     MPI25_FP_SCSI_IO_REQUEST_DESCRIPTOR         FastPathSCSIIO;
447     U64                                         Words;
448 } MPI2_REQUEST_DESCRIPTOR_UNION, MPI2_POINTER PTR_MPI2_REQUEST_DESCRIPTOR_UNION,
______unchanged_portion_omitted_
```



```
462   Mpi2DefaultReplyDescriptor_t, MPI2_POINTER pMpi2DefaultReplyDescriptor_t;

464 /* defines for the ReplyFlags field */
465 #define MPI2_RPY_DESCRIPT_FLAGS_TYPE_MASK                   (0x0F)
466 #define MPI2_RPY_DESCRIPT_FLAGS_SCSI_IO_SUCCESS             (0x00)
467 #define MPI2_RPY_DESCRIPT_FLAGS_ADDRESS_REPLY               (0x01)
468 #define MPI2_RPY_DESCRIPT_FLAGS_TARGETASSIST_SUCCESS        (0x02)
469 #define MPI2_RPY_DESCRIPT_FLAGS_TARGET_COMMAND_BUFFER       (0x03)
470 #define MPI2_RPY_DESCRIPT_FLAGS_RAID_ACCELERATOR_SUCCESS    (0x05)
471 #define MPI25_RPY_DESCRIPT_FLAGS_FAST_PATH_SCSI_IO_SUCCESS  (0x06)
472 #define MPI2_RPY_DESCRIPT_FLAGS_UNUSED                      (0x0F)

474 /* values for marking a reply descriptor as unused */
475 #define MPI2_RPY_DESCRIPT_UNUSED_WORD0_MARK             (0xFFFFFFFF)
476 #define MPI2_RPY_DESCRIPT_UNUSED_WORD1_MARK             (0xFFFFFFFF)

478 /* Address Reply Descriptor */
479 typedef struct _MPI2_ADDRESS_REPLY_DESCRIPTOR
480 {
481     U8              ReplyFlags;                 /* 0x00 */
482     U8              MSIxIndex;                  /* 0x01 */
483     U16             SMID;                       /* 0x02 */
484     U32             ReplyFrameAddress;          /* 0x04 */
485 } MPI2_ADDRESS_REPLY_DESCRIPTOR, MPI2_POINTER PTR_MPI2_ADDRESS_REPLY_DESCRIPTOR,
______unchanged_portion_omitted_
546   MPI2_POINTER PTR_MPI2_RAID_ACCELERATOR_SUCCESS_REPLY_DESCRIPTOR,
547   Mpi2RAIDAcceleratorSuccessReplyDescriptor_t,
548   MPI2_POINTER pMpi2RAIDAcceleratorSuccessReplyDescriptor_t;


551 /* Fast Path SCSI IO Success Reply Descriptor */
552 typedef MPI2_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR
553     MPI25_FP_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR,
554     MPI2_POINTER PTR_MPI25_FP_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR,
555     Mpi25FastPathSCSIIOSuccessReplyDescriptor_t,
556     MPI2_POINTER pMpi25FastPathSCSIIOSuccessReplyDescriptor_t;


559 /* union of Reply Descriptors */
560 typedef union _MPI2_REPLY_DESCRIPTORS_UNION
561 {
562     MPI2_DEFAULT_REPLY_DESCRIPTOR                   Default;
563     MPI2_ADDRESS_REPLY_DESCRIPTOR                   AddressReply;
564     MPI2_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR           SCSIIOSuccess;
565     MPI2_TARGETASSIST_SUCCESS_REPLY_DESCRIPTOR      TargetAssistSuccess;
566     MPI2_TARGET_COMMAND_BUFFER_REPLY_DESCRIPTOR     TargetCommandBuffer;
567     MPI2_RAID_ACCELERATOR_SUCCESS_REPLY_DESCRIPTOR  RAIDAcceleratorSuccess;
568     MPI25_FP_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR       FastPathSCSIIOSuccess;
569     U64                                             Words;
570 } MPI2_REPLY_DESCRIPTORS_UNION, MPI2_POINTER PTR_MPI2_REPLY_DESCRIPTORS_UNION,
571   Mpi2ReplyDescriptorsUnion_t, MPI2_POINTER pMpi2ReplyDescriptorsUnion_t;



575 /*****************************************************************************
576 *
577 *        Message Functions
516 *             0x80 -> 0x8F reserved for private message use per product
578 *
518 *
579 *****************************************************************************/

581 #define MPI2_FUNCTION_SCSI_IO_REQUEST               (0x00) /* SCSI IO */
582 #define MPI2_FUNCTION_SCSI_TASK_MGMT                (0x01) /* SCSI Task Manageme
583 #define MPI2_FUNCTION_IOC_INIT                      (0x02) /* IOC Init */
584 #define MPI2_FUNCTION_IOC_FACTS                     (0x03) /* IOC Facts */
```

```
585 #define MPI2_FUNCTION_CONFIG                        (0x04) /* Configuration */
586 #define MPI2_FUNCTION_PORT_FACTS                    (0x05) /* Port Facts */
587 #define MPI2_FUNCTION_PORT_ENABLE                   (0x06) /* Port Enable */
588 #define MPI2_FUNCTION_EVENT_NOTIFICATION            (0x07) /* Event Notification
589 #define MPI2_FUNCTION_EVENT_ACK                     (0x08) /* Event Acknowledge
590 #define MPI2_FUNCTION_FW_DOWNLOAD                   (0x09) /* FW Download */
591 #define MPI2_FUNCTION_TARGET_ASSIST                 (0x0B) /* Target Assist */
592 #define MPI2_FUNCTION_TARGET_STATUS_SEND            (0x0C) /* Target Status Send
593 #define MPI2_FUNCTION_TARGET_MODE_ABORT             (0x0D) /* Target Mode Abort
594 #define MPI2_FUNCTION_FW_UPLOAD                     (0x12) /* FW Upload */
595 #define MPI2_FUNCTION_RAID_ACTION                   (0x15) /* RAID Action */
596 #define MPI2_FUNCTION_RAID_SCSI_IO_PASSTHROUGH      (0x16) /* SCSI IO RAID Passt
597 #define MPI2_FUNCTION_TOOLBOX                       (0x17) /* Toolbox */
598 #define MPI2_FUNCTION_SCSI_ENCLOSURE_PROCESSOR      (0x18) /* SCSI Enclosure Pro
599 #define MPI2_FUNCTION_SMP_PASSTHROUGH               (0x1A) /* SMP Passthrough */
600 #define MPI2_FUNCTION_SAS_IO_UNIT_CONTROL           (0x1B) /* SAS IO Unit Contro
601 #define MPI2_FUNCTION_SATA_PASSTHROUGH              (0x1C) /* SATA Passthrough *
602 #define MPI2_FUNCTION_DIAG_BUFFER_POST              (0x1D) /* Diagnostic Buffer
603 #define MPI2_FUNCTION_DIAG_RELEASE                  (0x1E) /* Diagnostic Release
604 #define MPI2_FUNCTION_TARGET_CMD_BUF_BASE_POST      (0x24) /* Target Command Buf
605 #define MPI2_FUNCTION_TARGET_CMD_BUF_LIST_POST      (0x25) /* Target Command Buf
606 #define MPI2_FUNCTION_RAID_ACCELERATOR              (0x2C) /* RAID Accelerator *
607 #define MPI2_FUNCTION_HOST_BASED_DISCOVERY_ACTION   (0x2F) /* Host Based Discove
608 #define MPI2_FUNCTION_PWR_MGMT_CONTROL              (0x30) /* Power Management C
609 #define MPI2_FUNCTION_SEND_HOST_MESSAGE             (0x31) /* Send Host Message
610 #define MPI2_FUNCTION_MIN_PRODUCT_SPECIFIC          (0xF0) /* beginning of produ
611 #define MPI2_FUNCTION_MAX_PRODUCT_SPECIFIC          (0xFF) /* end of product-spe


615 /* Doorbell functions */
616 #define MPI2_FUNCTION_IOC_MESSAGE_UNIT_RESET        (0x40)
552 /* #define MPI2_FUNCTION_IO_UNIT_RESET                 (0x41) */
617 #define MPI2_FUNCTION_HANDSHAKE                     (0x42)


620 /****************************************************************************
621 *
622 *        IOC Status Values
623 *
624 ****************************************************************************/

626 /* mask for IOCStatus status value */
627 #define MPI2_IOCSTATUS_MASK                     (0x7FFF)

629 /****************************************************************************
630 *  Common IOCStatus values for all replies
631 ****************************************************************************/

633 #define MPI2_IOCSTATUS_SUCCESS                      (0x0000)
634 #define MPI2_IOCSTATUS_INVALID_FUNCTION             (0x0001)
635 #define MPI2_IOCSTATUS_BUSY                         (0x0002)
636 #define MPI2_IOCSTATUS_INVALID_SGL                  (0x0003)
637 #define MPI2_IOCSTATUS_INTERNAL_ERROR               (0x0004)
638 #define MPI2_IOCSTATUS_INVALID_VPID                 (0x0005)
639 #define MPI2_IOCSTATUS_INSUFFICIENT_RESOURCES       (0x0006)
640 #define MPI2_IOCSTATUS_INVALID_FIELD                (0x0007)
641 #define MPI2_IOCSTATUS_INVALID_STATE                (0x0008)
642 #define MPI2_IOCSTATUS_OP_STATE_NOT_SUPPORTED       (0x0009)

644 /****************************************************************************
645 *  Config IOCStatus values
646 ****************************************************************************/

648 #define MPI2_IOCSTATUS_CONFIG_INVALID_ACTION        (0x0020)
649 #define MPI2_IOCSTATUS_CONFIG_INVALID_TYPE          (0x0021)
```



```
650 #define MPI2_IOCSTATUS_CONFIG_INVALID_PAGE          (0x0022)
651 #define MPI2_IOCSTATUS_CONFIG_INVALID_DATA          (0x0023)
652 #define MPI2_IOCSTATUS_CONFIG_NO_DEFAULTS           (0x0024)
653 #define MPI2_IOCSTATUS_CONFIG_CANT_COMMIT           (0x0025)

655 /****************************************************************************
656 *  SCSI IO Reply
657 ****************************************************************************/

659 #define MPI2_IOCSTATUS_SCSI_RECOVERED_ERROR         (0x0040)
660 #define MPI2_IOCSTATUS_SCSI_INVALID_DEVHANDLE       (0x0042)
661 #define MPI2_IOCSTATUS_SCSI_DEVICE_NOT_THERE        (0x0043)
662 #define MPI2_IOCSTATUS_SCSI_DATA_OVERRUN            (0x0044)
663 #define MPI2_IOCSTATUS_SCSI_DATA_UNDERRUN           (0x0045)
664 #define MPI2_IOCSTATUS_SCSI_IO_DATA_ERROR           (0x0046)
665 #define MPI2_IOCSTATUS_SCSI_PROTOCOL_ERROR          (0x0047)
666 #define MPI2_IOCSTATUS_SCSI_TASK_TERMINATED         (0x0048)
667 #define MPI2_IOCSTATUS_SCSI_RESIDUAL_MISMATCH       (0x0049)
668 #define MPI2_IOCSTATUS_SCSI_TASK_MGMT_FAILED        (0x004A)
669 #define MPI2_IOCSTATUS_SCSI_IOC_TERMINATED          (0x004B)
670 #define MPI2_IOCSTATUS_SCSI_EXT_TERMINATED          (0x004C)

672 /****************************************************************************
673 *  For use by SCSI Initiator and SCSI Target end-to-end data protection
674 ****************************************************************************/

676 #define MPI2_IOCSTATUS_EEDP_GUARD_ERROR             (0x004D)
677 #define MPI2_IOCSTATUS_EEDP_REF_TAG_ERROR           (0x004E)
678 #define MPI2_IOCSTATUS_EEDP_APP_TAG_ERROR           (0x004F)

680 /****************************************************************************
681 *  SCSI Target values
682 ****************************************************************************/

684 #define MPI2_IOCSTATUS_TARGET_INVALID_IO_INDEX      (0x0062)
685 #define MPI2_IOCSTATUS_TARGET_ABORTED               (0x0063)
686 #define MPI2_IOCSTATUS_TARGET_NO_CONN_RETRYABLE     (0x0064)
687 #define MPI2_IOCSTATUS_TARGET_NO_CONNECTION         (0x0065)
688 #define MPI2_IOCSTATUS_TARGET_XFER_COUNT_MISMATCH   (0x006A)
689 #define MPI2_IOCSTATUS_TARGET_DATA_OFFSET_ERROR     (0x006D)
690 #define MPI2_IOCSTATUS_TARGET_TOO_MUCH_WRITE_DATA   (0x006E)
691 #define MPI2_IOCSTATUS_TARGET_IU_TOO_SHORT          (0x006F)
692 #define MPI2_IOCSTATUS_TARGET_ACK_NAK_TIMEOUT       (0x0070)
693 #define MPI2_IOCSTATUS_TARGET_NAK_RECEIVED          (0x0071)

695 /****************************************************************************
696 *  Serial Attached SCSI values
697 ****************************************************************************/

699 #define MPI2_IOCSTATUS_SAS_SMP_REQUEST_FAILED       (0x0090)
700 #define MPI2_IOCSTATUS_SAS_SMP_DATA_OVERRUN         (0x0091)

702 /****************************************************************************
703 *  Diagnostic Buffer Post / Diagnostic Release values
704 ****************************************************************************/

706 #define MPI2_IOCSTATUS_DIAGNOSTIC_RELEASED          (0x00A0)

708 /****************************************************************************
709 *  RAID Accelerator values
710 ****************************************************************************/

712 #define MPI2_IOCSTATUS_RAID_ACCEL_ERROR             (0x00B0)

714 /****************************************************************************
715 *  IOCStatus flag to indicate that log info is available
```

```
 716 *****************************************************************************/

 718 #define MPI2_IOCSTATUS_FLAG_LOG_INFO_AVAILABLE      (0x8000)

 720 /****************************************************************************
 721 *  IOCLogInfo Types
 722 ****************************************************************************/

 724 #define MPI2_IOCLOGINFO_TYPE_MASK               (0xF0000000)
 725 #define MPI2_IOCLOGINFO_TYPE_SHIFT              (28)
 726 #define MPI2_IOCLOGINFO_TYPE_NONE               (0x0)
 727 #define MPI2_IOCLOGINFO_TYPE_SCSI               (0x1)
 728 #define MPI2_IOCLOGINFO_TYPE_FC                 (0x2)
 729 #define MPI2_IOCLOGINFO_TYPE_SAS                (0x3)
 730 #define MPI2_IOCLOGINFO_TYPE_ISCSI              (0x4)
 731 #define MPI2_IOCLOGINFO_LOG_DATA_MASK           (0x0FFFFFFF)


 734 /****************************************************************************
 735 *
 736 *        Standard Message Structures
 737 *
 738 ****************************************************************************/

 740 /****************************************************************************
 741 * Request Message Header for all request messages
 742 ****************************************************************************/

 744 typedef struct _MPI2_REQUEST_HEADER
 745 {
 746     U16             FunctionDependent1;         /* 0x00 */
 747     U8              ChainOffset;                /* 0x02 */
 748     U8              Function;                   /* 0x03 */
 749     U16             FunctionDependent2;         /* 0x04 */
 750     U8              FunctionDependent3;         /* 0x06 */
 751     U8              MsgFlags;                   /* 0x07 */
 752     U8              VP_ID;                      /* 0x08 */
 753     U8              VF_ID;                      /* 0x09 */
 754     U16             Reserved1;                  /* 0x0A */
 755 } MPI2_REQUEST_HEADER, MPI2_POINTER PTR_MPI2_REQUEST_HEADER,
_____unchanged_portion_omitted_
 840   Mpi2SGESimpleUnion_t, MPI2_POINTER pMpi2SGESimpleUnion_t;


 843 /****************************************************************************
 844 *  MPI Chain Element structures - for MPI v2.0 products only
 780 *  MPI Chain Element structures
 845 ****************************************************************************/

 847 typedef struct _MPI2_SGE_CHAIN32
 848 {
 849     U16                     Length;
 850     U8                      NextChainOffset;
 851     U8                      Flags;
 852     U32                     Address;
 853 } MPI2_SGE_CHAIN32, MPI2_POINTER PTR_MPI2_SGE_CHAIN32,
_____unchanged_portion_omitted_
 876   Mpi2SGEChainUnion_t, MPI2_POINTER pMpi2SGEChainUnion_t;


 879 /****************************************************************************
 880 *  MPI Transaction Context Element structures - for MPI v2.0 products only
 816 *  MPI Transaction Context Element structures
 881 ****************************************************************************/

 883 typedef struct _MPI2_SGE_TRANSACTION32
```



```
 884 {
 885     U8                      Reserved;
 886     U8                      ContextSize;
 887     U8                      DetailsLength;
 888     U8                      Flags;
 889     U32                     TransactionContext[1];
 890     U32                     TransactionDetails[1];
 891 } MPI2_SGE_TRANSACTION32, MPI2_POINTER PTR_MPI2_SGE_TRANSACTION32,
_____unchanged_portion_omitted_
 942   Mpi2SGETransactionUnion_t, MPI2_POINTER pMpi2SGETransactionUnion_t;


 945 /****************************************************************************
 946 *  MPI SGE union for IO SGL's - for MPI v2.0 products only
 882 *  MPI SGE union for IO SGL's
 947 ****************************************************************************/

 949 typedef struct _MPI2_MPI_SGE_IO_UNION
 950 {
 951     union
 952     {
 953         MPI2_SGE_SIMPLE_UNION   Simple;
 954         MPI2_SGE_CHAIN_UNION    Chain;
 955     } u;
 956 } MPI2_MPI_SGE_IO_UNION, MPI2_POINTER PTR_MPI2_MPI_SGE_IO_UNION,
 957   Mpi2MpiSGEIOUnion_t, MPI2_POINTER pMpi2MpiSGEIOUnion_t;


 960 /****************************************************************************
 961 *  MPI SGE union for SGL's with Simple and Transaction elements - for MPI v2.0 p
 897 *  MPI SGE union for SGL's with Simple and Transaction elements
 962 ****************************************************************************/

 964 typedef struct _MPI2_SGE_TRANS_SIMPLE_UNION
 965 {
 966     union
 967     {
 968         MPI2_SGE_SIMPLE_UNION       Simple;
 969         MPI2_SGE_TRANSACTION_UNION  Transaction;
 970     } u;
 971 } MPI2_SGE_TRANS_SIMPLE_UNION, MPI2_POINTER PTR_MPI2_SGE_TRANS_SIMPLE_UNION,
_____unchanged_portion_omitted_
 988   Mpi2MpiSgeUnion_t, MPI2_POINTER pMpi2MpiSgeUnion_t;


 991 /****************************************************************************
 992 *  MPI SGE field definition and masks
 993 ****************************************************************************/

 995 /* Flags field bit definitions */

 997 #define MPI2_SGE_FLAGS_LAST_ELEMENT             (0x80)
 998 #define MPI2_SGE_FLAGS_END_OF_BUFFER            (0x40)
 999 #define MPI2_SGE_FLAGS_ELEMENT_TYPE_MASK        (0x30)
1000 #define MPI2_SGE_FLAGS_LOCAL_ADDRESS            (0x08)
1001 #define MPI2_SGE_FLAGS_DIRECTION                (0x04)
1002 #define MPI2_SGE_FLAGS_ADDRESS_SIZE             (0x02)
1003 #define MPI2_SGE_FLAGS_END_OF_LIST              (0x01)

1005 #define MPI2_SGE_FLAGS_SHIFT                    (24)

1007 #define MPI2_SGE_LENGTH_MASK                    (0x00FFFFFF)
1008 #define MPI2_SGE_CHAIN_LENGTH_MASK              (0x0000FFFF)

1010 /* Element Type */
```

```
1012 #define MPI2_SGE_FLAGS_TRANSACTION_ELEMENT      (0x00) /* for MPI v2.0 products
 948 #define MPI2_SGE_FLAGS_TRANSACTION_ELEMENT      (0x00)
1013 #define MPI2_SGE_FLAGS_SIMPLE_ELEMENT           (0x10)
1014 #define MPI2_SGE_FLAGS_CHAIN_ELEMENT           (0x30) /* for MPI v2.0 products
 950 #define MPI2_SGE_FLAGS_CHAIN_ELEMENT           (0x30)
1015 #define MPI2_SGE_FLAGS_ELEMENT_MASK            (0x30)

1017 /* Address location */

1019 #define MPI2_SGE_FLAGS_SYSTEM_ADDRESS          (0x00)

1021 /* Direction */

1023 #define MPI2_SGE_FLAGS_IOC_TO_HOST             (0x00)
1024 #define MPI2_SGE_FLAGS_HOST_TO_IOC             (0x04)

1026 #define MPI2_SGE_FLAGS_DEST                    (MPI2_SGE_FLAGS_IOC_TO_HOST)
1027 #define MPI2_SGE_FLAGS_SOURCE                  (MPI2_SGE_FLAGS_HOST_TO_IOC)

1029 /* Address Size */

1031 #define MPI2_SGE_FLAGS_32_BIT_ADDRESSING       (0x00)
1032 #define MPI2_SGE_FLAGS_64_BIT_ADDRESSING       (0x02)

1034 /* Context Size */

1036 #define MPI2_SGE_FLAGS_32_BIT_CONTEXT          (0x00)
1037 #define MPI2_SGE_FLAGS_64_BIT_CONTEXT          (0x02)
1038 #define MPI2_SGE_FLAGS_96_BIT_CONTEXT          (0x04)
1039 #define MPI2_SGE_FLAGS_128_BIT_CONTEXT         (0x06)

1041 #define MPI2_SGE_CHAIN_OFFSET_MASK             (0x00FF0000)
1042 #define MPI2_SGE_CHAIN_OFFSET_SHIFT            (16)

1044 /****************************************************************************
1045 *  MPI SGE operation Macros
1046 ****************************************************************************/

1048 /* SIMPLE FlagsLength manipulations... */
1049 #define MPI2_SGE_SET_FLAGS(f)          ((U32)(f) << MPI2_SGE_FLAGS_SHIFT)
1050 #define MPI2_SGE_GET_FLAGS(f)          (((f) & ~MPI2_SGE_LENGTH_MASK) >> MPI2_SG
1051 #define MPI2_SGE_LENGTH(f)             ((f) & MPI2_SGE_LENGTH_MASK)
1052 #define MPI2_SGE_CHAIN_LENGTH(f)       ((f) & MPI2_SGE_CHAIN_LENGTH_MASK)

1054 #define MPI2_SGE_SET_FLAGS_LENGTH(f,l) (MPI2_SGE_SET_FLAGS(f) | MPI2_SGE_LENGTH(

1056 #define MPI2_pSGE_GET_FLAGS(psg)           MPI2_SGE_GET_FLAGS((psg)->FlagsLengt
1057 #define MPI2_pSGE_GET_LENGTH(psg)          MPI2_SGE_LENGTH((psg)->FlagsLength)
1058 #define MPI2_pSGE_SET_FLAGS_LENGTH(psg,f,l) (psg)->FlagsLength = MPI2_SGE_SET_FL

1060 /* CAUTION - The following are READ-MODIFY-WRITE! */
1061 #define MPI2_pSGE_SET_FLAGS(psg,f)     (psg)->FlagsLength |= MPI2_SGE_SET_FLAGS
1062 #define MPI2_pSGE_SET_LENGTH(psg,l)    (psg)->FlagsLength |= MPI2_SGE_LENGTH(l)

1064 #define MPI2_GET_CHAIN_OFFSET(x)    ((x & MPI2_SGE_CHAIN_OFFSET_MASK) >> MPI2_SG


1067 /****************************************************************************
1068 *
1069 *        Fusion-MPT IEEE Scatter Gather Elements
1070 *
1071 ****************************************************************************/

1073 /****************************************************************************
1074 *  IEEE Simple Element structures
1075 ****************************************************************************/
```



```
1077 /* MPI2_IEEE_SGE_SIMPLE32 is for MPI v2.0 products only */
1078 typedef struct _MPI2_IEEE_SGE_SIMPLE32
1079 {
1080     U32                     Address;
1081     U32                     FlagsLength;
1082 } MPI2_IEEE_SGE_SIMPLE32, MPI2_POINTER PTR_MPI2_IEEE_SGE_SIMPLE32,
_____unchanged_portion_omitted_
1100   Mpi2IeeeSgeSimpleUnion_t, MPI2_POINTER pMpi2IeeeSgeSimpleUnion_t;


1103 /****************************************************************************
1104 *  IEEE Chain Element structures
1105 ****************************************************************************/

1107 /* MPI2_IEEE_SGE_CHAIN32 is for MPI v2.0 products only */
1108 typedef MPI2_IEEE_SGE_SIMPLE32  MPI2_IEEE_SGE_CHAIN32;

1110 /* MPI2_IEEE_SGE_CHAIN64 is for MPI v2.0 products only */
1111 typedef MPI2_IEEE_SGE_SIMPLE64  MPI2_IEEE_SGE_CHAIN64;

1113 typedef union _MPI2_IEEE_SGE_CHAIN_UNION
1114 {
1115     MPI2_IEEE_SGE_CHAIN32   Chain32;
1116     MPI2_IEEE_SGE_CHAIN64   Chain64;
1117 } MPI2_IEEE_SGE_CHAIN_UNION, MPI2_POINTER PTR_MPI2_IEEE_SGE_CHAIN_UNION,
1118   Mpi2IeeeSgeChainUnion_t, MPI2_POINTER pMpi2IeeeSgeChainUnion_t;

1120 /* MPI25_IEEE_SGE_CHAIN64 is for MPI v2.5 products only */
1121 typedef struct _MPI25_IEEE_SGE_CHAIN64
1122 {
1123     U64                     Address;
1124     U32                     Length;
1125     U16                     Reserved1;
1126     U8                      NextChainOffset;
1127     U8                      Flags;
1128 } MPI25_IEEE_SGE_CHAIN64, MPI2_POINTER PTR_MPI25_IEEE_SGE_CHAIN64,
1129   Mpi25IeeeSgeChain64_t, MPI2_POINTER pMpi25IeeeSgeChain64_t;


1132 /****************************************************************************
1133 *  All IEEE SGE types union
1134 ****************************************************************************/

1136 /* MPI2_IEEE_SGE_UNION is for MPI v2.0 products only */
1137 typedef struct _MPI2_IEEE_SGE_UNION
1138 {
1139     union
1140     {
1141         MPI2_IEEE_SGE_SIMPLE_UNION  Simple;
1142         MPI2_IEEE_SGE_CHAIN_UNION   Chain;
1143     } u;
1144 } MPI2_IEEE_SGE_UNION, MPI2_POINTER PTR_MPI2_IEEE_SGE_UNION,
1145   Mpi2IeeeSgeUnion_t, MPI2_POINTER pMpi2IeeeSgeUnion_t;


1148 /****************************************************************************
1149 *  IEEE SGE union for IO SGL's
1150 ****************************************************************************/

1152 typedef union _MPI25_SGE_IO_UNION
1153 {
1154     MPI2_IEEE_SGE_SIMPLE64      IeeeSimple;
1155     MPI25_IEEE_SGE_CHAIN64      IeeeChain;
1156 } MPI25_SGE_IO_UNION, MPI2_POINTER PTR_MPI25_SGE_IO_UNION,
1157   Mpi25SGEIOUnion_t, MPI2_POINTER pMpi25SGEIOUnion_t;
```

```
1160 /*****************************************************************************
1161 *  IEEE SGE field definitions and masks
1162 *****************************************************************************/

1164 /* Flags field bit definitions */

1166 #define MPI2_IEEE_SGE_FLAGS_ELEMENT_TYPE_MASK   (0x80)
1167 #define MPI25_IEEE_SGE_FLAGS_END_OF_LIST        (0x40)

1169 #define MPI2_IEEE32_SGE_FLAGS_SHIFT             (24)

1171 #define MPI2_IEEE32_SGE_LENGTH_MASK             (0x00FFFFFF)

1173 /* Element Type */

1175 #define MPI2_IEEE_SGE_FLAGS_SIMPLE_ELEMENT      (0x00)
1176 #define MPI2_IEEE_SGE_FLAGS_CHAIN_ELEMENT       (0x80)

1178 /* Data Location Address Space */

1180 #define MPI2_IEEE_SGE_FLAGS_ADDR_MASK           (0x03)
1181 #define MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR         (0x00) /* for MPI v2.0, use in I
1182 #define MPI2_IEEE_SGE_FLAGS_IOCDDR_ADDR         (0x01) /* use in IEEE Simple Ele
1086 #define MPI2_IEEE_SGE_FLAGS_SYSTEM_ADDR         (0x00)
1087 #define MPI2_IEEE_SGE_FLAGS_IOCDDR_ADDR         (0x01)
1183 #define MPI2_IEEE_SGE_FLAGS_IOCPLB_ADDR         (0x02)
1184 #define MPI2_IEEE_SGE_FLAGS_IOCPLBNTA_ADDR      (0x03) /* for MPI v2.0, use in I
1185 #define MPI2_IEEE_SGE_FLAGS_SYSTEMPLBPCI_ADDR   (0x03) /* use in MPI v2.0 IEEE C
1186 #define MPI2_IEEE_SGE_FLAGS_SYSTEMPLBCPI_ADDR   (MPI2_IEEE_SGE_FLAGS_SYSTEMPLBPC
1089 #define MPI2_IEEE_SGE_FLAGS_IOCPLBNTA_ADDR      (0x03)


1188 /*****************************************************************************
1189 *  IEEE SGE operation Macros
1190 *****************************************************************************/

1192 /* SIMPLE FlagsLength manipulations... */
1193 #define MPI2_IEEE32_SGE_SET_FLAGS(f)     ((U32)(f) << MPI2_IEEE32_SGE_FLAGS_SHIF
1194 #define MPI2_IEEE32_SGE_GET_FLAGS(f)     (((f) & ~MPI2_IEEE32_SGE_LENGTH_MASK) >
1195 #define MPI2_IEEE32_SGE_LENGTH(f)        ((f) & MPI2_IEEE32_SGE_LENGTH_MASK)

1197 #define MPI2_IEEE32_SGE_SET_FLAGS_LENGTH(f, l)      (MPI2_IEEE32_SGE_SET_FLAGS(f

1199 #define MPI2_IEEE32_pSGE_GET_FLAGS(psg)             MPI2_IEEE32_SGE_GET_FLAGS((p
1200 #define MPI2_IEEE32_pSGE_GET_LENGTH(psg)            MPI2_IEEE32_SGE_LENGTH((psg)
1201 #define MPI2_IEEE32_pSGE_SET_FLAGS_LENGTH(psg,f,l)  (psg)->FlagsLength = MPI2_IE

1203 /* CAUTION - The following are READ-MODIFY-WRITE! */
1204 #define MPI2_IEEE32_pSGE_SET_FLAGS(psg,f)    (psg)->FlagsLength |= MPI2_IEEE32_S
1205 #define MPI2_IEEE32_pSGE_SET_LENGTH(psg,l)   (psg)->FlagsLength |= MPI2_IEEE32_S



1209 /*****************************************************************************
1210 *
1211 *        Fusion-MPT MPI/IEEE Scatter Gather Unions
1212 *
1213 *****************************************************************************/

1215 typedef union _MPI2_SIMPLE_SGE_UNION
1216 {
1217     MPI2_SGE_SIMPLE_UNION       MpiSimple;
1218     MPI2_IEEE_SGE_SIMPLE_UNION  IeeeSimple;
```



```
1219 } MPI2_SIMPLE_SGE_UNION, MPI2_POINTER PTR_MPI2_SIMPLE_SGE_UNION,
_____unchanged_portion_omitted_
1230   Mpi2SGEIOUnion_t, MPI2_POINTER pMpi2SGEIOUnion_t;


1233 /*****************************************************************************
1234 *
1235 *  Values for SGLFlags field, used in many request messages with an SGL
1236 *
1237 *****************************************************************************/

1239 /* values for MPI SGL Data Location Address Space subfield */
1240 #define MPI2_SGLFLAGS_ADDRESS_SPACE_MASK            (0x0C)
1241 #define MPI2_SGLFLAGS_SYSTEM_ADDRESS_SPACE          (0x00)
1242 #define MPI2_SGLFLAGS_IOCDDR_ADDRESS_SPACE          (0x04)
1243 #define MPI2_SGLFLAGS_IOCPLB_ADDRESS_SPACE          (0x08)
1244 #define MPI2_SGLFLAGS_IOCPLBNTA_ADDRESS_SPACE       (0x0C)
1245 /* values for SGL Type subfield */
1246 #define MPI2_SGLFLAGS_SGL_TYPE_MASK                 (0x03)
1247 #define MPI2_SGLFLAGS_SGL_TYPE_MPI                  (0x00)
1248 #define MPI2_SGLFLAGS_SGL_TYPE_IEEE32               (0x01) /* MPI v2.0 products
1153 #define MPI2_SGLFLAGS_SGL_TYPE_IEEE32               (0x01)
1249 #define MPI2_SGLFLAGS_SGL_TYPE_IEEE64               (0x02)


1252 #endif
```

```
**********************************************************
  158147 Mon Jun 16 21:18:06 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_cnfg.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 */
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_cnfg.h
 *         Title:  MPI Configuration messages and pages
 *  Creation Date:  November 10, 2006
 *
 *    mpi2_cnfg.h Version:  02.00.27
 *    mpi2_cnfg.h Version:  02.00.12
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-04-07  02.00.01  Added defines for SAS IO Unit Page 2 PhyFlags.
 *                      Added Manufacturing Page 11.
 *                      Added MPI2_SAS_EXPANDER0_FLAGS_CONNECTOR_END_DEVICE
 *                      define.
 *  06-26-07  02.00.02  Adding generic structure for product-specific
 *                      Manufacturing pages: MPI2_CONFIG_PAGE_MANUFACTURING_PS.
 *                      Rework of BIOS Page 2 configuration page.
 *                      Fixed MPI2_BIOSPAGE2_BOOT_DEVICE to be a union of the
 *                      forms.
 *                      Added configuration pages IOC Page 8 and Driver
 *                      Persistent Mapping Page 0.
 *  08-31-07  02.00.03  Modified configuration pages dealing with Integrated
 *                      RAID (Manufacturing Page 4, RAID Volume Pages 0 and 1,
 *                      RAID Physical Disk Pages 0 and 1, RAID Configuration
 *                      Page 0).
 *                      Added new value for AccessStatus field of SAS Device
 *                      Page 0 (_SATA_NEEDS_INITIALIZATION).
 *  10-31-07  02.00.04  Added missing SEPDevHandle field to
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *  12-18-07  02.00.05  Modified IO Unit Page 0 to use 32-bit version fields for
 *                      NVDATA.
 *                      Modified IOC Page 7 to use masks and added field for
 *                      SASBroadcastPrimitiveMasks.
 *                      Added MPI2_CONFIG_PAGE_BIOS_4.
 *                      Added MPI2_CONFIG_PAGE_LOG_0.
 *  02-29-08  02.00.06  Modified various names to make them 32-character unique.
 *                      Added SAS Device IDs.
 *                      Updated Integrated RAID configuration pages including
 *                      Manufacturing Page 4, IOC Page 6, and RAID Configuration
 *                      Page 0.
 *  05-21-08  02.00.07  Added define MPI2_MANPAGE4_MIX_SSD_SAS_SATA.
 *                      Added define MPI2_MANPAGE4_PHYSDISK_128MB_COERCION.
 *                      Fixed define MPI2_IOCPAGE8_FLAGS_ENCLOSURE_SLOT_MAPPING.
 *                      Added missing MaxNumRoutedSasAddresses field to
```

```
 *                      MPI2_CONFIG_PAGE_EXPANDER_0.
 *                      Added SAS Port Page 0.
 *                      Modified structure layout for
 *                      MPI2_CONFIG_PAGE_DRIVER_MAPPING_0.
 *  06-27-08  02.00.08  Changed MPI2_CONFIG_PAGE_RD_PDISK_1 to use
 *                      MPI2_RAID_PHYS_DISK1_PATH_MAX to size the array.
 *  10-02-08  02.00.09  Changed MPI2_RAID_PGAD_CONFIGNUM_MASK from 0x0000FFFF
 *                      to 0x000000FF.
 *                      Added two new values for the Physical Disk Coercion Size
 *                      bits in the Flags field of Manufacturing Page 4.
 *                      Added product-specific Manufacturing pages 16 to 31.
 *                      Modified Flags bits for controlling write cache on SATA
 *                      drives in IO Unit Page 1.
 *                      Added new bit to AdditionalControlFlags of SAS IO Unit
 *                      Page 1 to control Invalid Topology Correction.
 *                      Added additional defines for RAID Volume Page 0
 *                      VolumeStatusFlags field.
 *                      Modified meaning of RAID Volume Page 0 VolumeSettings
 *                      define for auto-configure of hot-swap drives.
 *                      Added SupportedPhysDisks field to RAID Volume Page 1 and
 *                      added related defines.
 *                      Added PhysDiskAttributes field (and related defines) to
 *                      RAID Physical Disk Page 0.
 *                      Added MPI2_SAS_PHYINFO_PHY_VACANT define.
 *                      Added three new DiscoveryStatus bits for SAS IO Unit
 *                      Page 0 and SAS Expander Page 0.
 *                      Removed multiplexing information from SAS IO Unit pages.
 *                      Added BootDeviceWaitTime field to SAS IO Unit Page 4.
 *                      Removed Zone Address Resolved bit from PhyInfo and from
 *                      Expander Page 0 Flags field.
 *                      Added two new AccessStatus values to SAS Device Page 0
 *                      for indicating routing problems. Added 3 reserved words
 *                      to this page.
 *  01-19-09  02.00.10  Fixed defines for GPIOVal field of IO Unit Page 3.
 *                      Inserted missing reserved field into structure for IOC
 *                      Page 6.
 *                      Added more pending task bits to RAID Volume Page 0
 *                      VolumeStatusFlags defines.
 *                      Added MPI2_PHYSDISK0_STATUS_FLAG_NOT_CERTIFIED define.
 *                      Added a new DiscoveryStatus bit for SAS IO Unit Page 0
 *                      and SAS Expander Page 0 to flag a downstream initiator
 *                      when in simplified routing mode.
 *                      Removed SATA Init Failure defines for DiscoveryStatus
 *                      fields of SAS IO Unit Page 0 and SAS Expander Page 0.
 *                      Added MPI2_SAS_DEVICE0_ASTATUS_DEVICE_BLOCKED define.
 *                      Added PortGroups, DmaGroup, and ControlGroup fields to
 *                      SAS Device Page 0.
 *  05-06-09  02.00.11  Added structures and defines for IO Unit Page 5 and IO
 *                      Unit Page 6.
 *                      Added expander reduced functionality data to SAS
 *                      Expander Page 0.
 *                      Added SAS PHY Page 2 and SAS PHY Page 3.
 *  07-30-09  02.00.12  Added IO Unit Page 7.
 *                      Added new device ids.
 *                      Added SAS IO Unit Page 5.
 *                      Added partial and slumber power management capable flags
 *                      to SAS Device Page 0 Flags field.
 *                      Added PhyInfo defines for power condition.
 *                      Added Ethernet configuration pages.
 *  10-28-09  02.00.13  Added MPI2_IOUNITPAGE1_ENABLE_HOST_BASED_DISCOVERY.
 *                      Added SAS PHY Page 4 structure and defines.
 *  02-10-10  02.00.14  Modified the comments for the configuration page
 *                      structures that contain an array of data. The host
 *                      should use the "count" field in the page data (e.g. the
 *                      NumPhys field) to determine the number of valid elements
 *                      in the array.
```



```
 *                      Added/modified some MPI2_MFGPAGE_DEVID_SAS defines.
 *                      Added PowerManagementCapabilities to IO Unit Page 7.
 *                      Added PortWidthModGroup field to
 *                      MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_6 and related defines.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_7 and related defines.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_8 and related defines.
 *  05-12-10  02.00.15  Added MPI2_RAIDVOL0_STATUS_FLAG_VOL_NOT_CONSISTENT
 *                      define.
 *                      Added MPI2_PHYSDISK0_INCOMPATIBLE_MEDIA_TYPE define.
 *                      Added MPI2_SAS_NEG_LINK_RATE_UNSUPPORTED_PHY define.
 *  08-11-10  02.00.16  Removed IO Unit Page 1 device path (multi-pathing)
 *                      defines.
 *  11-10-10  02.00.17  Added ReceptacleID field (replacing Reserved1) to
 *                      MPI2_MANPAGE7_CONNECTOR_INFO and reworked defines for
 *                      the Pinout field.
 *                      Added BoardTemperature and BoardTemperatureUnits fields
 *                      to MPI2_CONFIG_PAGE_IO_UNIT_7.
 *                      Added MPI2_CONFIG_EXTPAGETYPE_EXT_MANUFACTURING define
 *                      and MPI2_CONFIG_PAGE_EXT_MAN_PS structure.
 *  02-23-11  02.00.18  Added ProxyVF_ID field to MPI2_CONFIG_REQUEST.
 *                      Added IO Unit Page 8, IO Unit Page 9,
 *                      and IO Unit Page 10.
 *                      Added SASNotifyPrimitiveMasks field to
 *                      MPI2_CONFIG_PAGE_IOC_7.
 *  03-09-11  02.00.19  Fixed IO Unit Page 10 (to match the spec).
 *  05-25-11  02.00.20  Cleaned up a few comments.
 *  08-24-11  02.00.21  Marked the IO Unit Page 7 PowerManagementCapabilities
 *                      for PCIe link as obsolete.
 *                      Added SpinupFlags field containing a Disable Spin-up bit
 *                      to the MPI2_SAS_IOUNIT4_SPINUP_GROUP fields of SAS IO
 *                      Unit Page 4.
 *  11-18-11  02.00.22  Added define MPI2_IOCPAGE6_CAP_FLAGS_4K_SECTORS_SUPPORT.
 *                      Added UEFIVersion field to BIOS Page 1 and defined new
 *                      BiosOptions bits.
 *                      Incorporating additions for MPI v2.5.
 *  11-27-12  02.00.23  Added MPI2_MANPAGE7_FLAG_EVENTREPLAY_SLOT_ORDER.
 *                      Added MPI2_BIOSPAGE1_OPTIONS_MASK_OEM_ID.
 *  12-20-12  02.00.24  Marked MPI2_SASIOUNIT1_CONTROL_CLEAR_AFFILIATION as
 *                      obsolete for MPI v2.5 and later.
 *                      Added some defines for 12G SAS speeds.
 *  04-09-13  02.00.25  Added MPI2_IOUNITPAGE1_ATA_SECURITY_FREEZE_LOCK.
 *                      Fixed MPI2_IOUNITPAGE5_DMA_CAP_MASK_MAX_REQUESTS to
 *                      match the specification.
 *  08-19-13  02.00.26  Added reserved words to MPI2_CONFIG_PAGE_IO_UNIT_7 for
 *                      future use.
 *  12-05-13  02.00.27  Added MPI2_MANPAGE7_FLAG_BASE_ENCLOSURE_LEVEL for
 *                      MPI2_CONFIG_PAGE_MAN_7.
 *                      Added EnclosureLevel and ConnectorName fields to
 *                      MPI2_CONFIG_PAGE_SAS_DEV_0.
 *                      Added MPI2_SAS_DEVICE0_FLAGS_ENCL_LEVEL_VALID for
 *                      MPI2_CONFIG_PAGE_SAS_DEV_0.
 *                      Added EnclosureLevel field to
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *                      Added MPI2_SAS_ENCLS0_FLAGS_ENCL_LEVEL_VALID for
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_CNFG_H
#define MPI2_CNFG_H

/*****************************************************************************
*   Configuration Page Header and defines
*****************************************************************************/
```

```
 217 /* Config Page Header */
 218 typedef struct _MPI2_CONFIG_PAGE_HEADER
 219 {
 220     U8                 PageVersion;               /* 0x00 */
 221     U8                 PageLength;                /* 0x01 */
 222     U8                 PageNumber;                /* 0x02 */
 223     U8                 PageType;                  /* 0x03 */
 224 } MPI2_CONFIG_PAGE_HEADER, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_HEADER,
_____unchanged_portion_omitted_
 258   Mpi2ConfigPageExtendedHeaderUnion, MPI2_POINTER pMpi2ConfigPageExtendedHeaderU


 261 /* PageType field values */
 262 #define MPI2_CONFIG_PAGEATTR_READ_ONLY             (0x00)
 263 #define MPI2_CONFIG_PAGEATTR_CHANGEABLE            (0x10)
 264 #define MPI2_CONFIG_PAGEATTR_PERSISTENT            (0x20)
 265 #define MPI2_CONFIG_PAGEATTR_MASK                  (0xF0)

 267 #define MPI2_CONFIG_PAGETYPE_IO_UNIT               (0x00)
 268 #define MPI2_CONFIG_PAGETYPE_IOC                   (0x01)
 269 #define MPI2_CONFIG_PAGETYPE_BIOS                  (0x02)
 270 #define MPI2_CONFIG_PAGETYPE_RAID_VOLUME           (0x08)
 271 #define MPI2_CONFIG_PAGETYPE_MANUFACTURING         (0x09)
 272 #define MPI2_CONFIG_PAGETYPE_RAID_PHYSDISK         (0x0A)
 273 #define MPI2_CONFIG_PAGETYPE_EXTENDED              (0x0F)
 274 #define MPI2_CONFIG_PAGETYPE_MASK                  (0x0F)

 276 #define MPI2_CONFIG_TYPENUM_MASK                   (0x0FFF)


 279 /* ExtPageType field values */
 280 #define MPI2_CONFIG_EXTPAGETYPE_SAS_IO_UNIT        (0x10)
 281 #define MPI2_CONFIG_EXTPAGETYPE_SAS_EXPANDER       (0x11)
 282 #define MPI2_CONFIG_EXTPAGETYPE_SAS_DEVICE         (0x12)
 283 #define MPI2_CONFIG_EXTPAGETYPE_SAS_PHY            (0x13)
 284 #define MPI2_CONFIG_EXTPAGETYPE_LOG                (0x14)
 285 #define MPI2_CONFIG_EXTPAGETYPE_ENCLOSURE          (0x15)
 286 #define MPI2_CONFIG_EXTPAGETYPE_RAID_CONFIG        (0x16)
 287 #define MPI2_CONFIG_EXTPAGETYPE_DRIVER_MAPPING     (0x17)
 288 #define MPI2_CONFIG_EXTPAGETYPE_SAS_PORT           (0x18)
 289 #define MPI2_CONFIG_EXTPAGETYPE_ETHERNET           (0x19)
 290 #define MPI2_CONFIG_EXTPAGETYPE_EXT_MANUFACTURING  (0x1A)


 293 /*****************************************************************************
 294 *   PageAddress defines
 295 *****************************************************************************/

 297 /* RAID Volume PageAddress format */
 298 #define MPI2_RAID_VOLUME_PGAD_FORM_MASK            (0xF0000000)
 299 #define MPI2_RAID_VOLUME_PGAD_FORM_GET_NEXT_HANDLE (0x00000000)
 300 #define MPI2_RAID_VOLUME_PGAD_FORM_HANDLE          (0x10000000)

 302 #define MPI2_RAID_VOLUME_PGAD_HANDLE_MASK          (0x0000FFFF)


 305 /* RAID Physical Disk PageAddress format */
 306 #define MPI2_PHYSDISK_PGAD_FORM_MASK                  (0xF0000000)
 307 #define MPI2_PHYSDISK_PGAD_FORM_GET_NEXT_PHYSDISKNUM  (0x00000000)
 308 #define MPI2_PHYSDISK_PGAD_FORM_PHYSDISKNUM           (0x10000000)
 309 #define MPI2_PHYSDISK_PGAD_FORM_DEVHANDLE             (0x20000000)

 311 #define MPI2_PHYSDISK_PGAD_PHYSDISKNUM_MASK           (0x000000FF)
 312 #define MPI2_PHYSDISK_PGAD_DEVHANDLE_MASK             (0x0000FFFF)
```



```
 315 /* SAS Expander PageAddress format */
 316 #define MPI2_SAS_EXPAND_PGAD_FORM_MASK             (0xF0000000)
 317 #define MPI2_SAS_EXPAND_PGAD_FORM_GET_NEXT_HNDL    (0x00000000)
 318 #define MPI2_SAS_EXPAND_PGAD_FORM_HNDL_PHY_NUM     (0x10000000)
 319 #define MPI2_SAS_EXPAND_PGAD_FORM_HNDL             (0x20000000)

 321 #define MPI2_SAS_EXPAND_PGAD_HANDLE_MASK           (0x0000FFFF)
 322 #define MPI2_SAS_EXPAND_PGAD_PHYNUM_MASK           (0x00FF0000)
 323 #define MPI2_SAS_EXPAND_PGAD_PHYNUM_SHIFT          (16)


 326 /* SAS Device PageAddress format */
 327 #define MPI2_SAS_DEVICE_PGAD_FORM_MASK             (0xF0000000)
 328 #define MPI2_SAS_DEVICE_PGAD_FORM_GET_NEXT_HANDLE  (0x00000000)
 329 #define MPI2_SAS_DEVICE_PGAD_FORM_HANDLE           (0x20000000)

 331 #define MPI2_SAS_DEVICE_PGAD_HANDLE_MASK           (0x0000FFFF)


 334 /* SAS PHY PageAddress format */
 335 #define MPI2_SAS_PHY_PGAD_FORM_MASK                (0xF0000000)
 336 #define MPI2_SAS_PHY_PGAD_FORM_PHY_NUMBER          (0x00000000)
 337 #define MPI2_SAS_PHY_PGAD_FORM_PHY_TBL_INDEX       (0x10000000)

 339 #define MPI2_SAS_PHY_PGAD_PHY_NUMBER_MASK          (0x000000FF)
 340 #define MPI2_SAS_PHY_PGAD_PHY_TBL_INDEX_MASK       (0x0000FFFF)


 343 /* SAS Port PageAddress format */
 344 #define MPI2_SASPORT_PGAD_FORM_MASK                (0xF0000000)
 345 #define MPI2_SASPORT_PGAD_FORM_GET_NEXT_PORT       (0x00000000)
 346 #define MPI2_SASPORT_PGAD_FORM_PORT_NUM            (0x10000000)

 348 #define MPI2_SASPORT_PGAD_PORTNUMBER_MASK          (0x00000FFF)


 351 /* SAS Enclosure PageAddress format */
 352 #define MPI2_SAS_ENCLOS_PGAD_FORM_MASK             (0xF0000000)
 353 #define MPI2_SAS_ENCLOS_PGAD_FORM_GET_NEXT_HANDLE  (0x00000000)
 354 #define MPI2_SAS_ENCLOS_PGAD_FORM_HANDLE           (0x10000000)

 356 #define MPI2_SAS_ENCLOS_PGAD_HANDLE_MASK           (0x0000FFFF)


 359 /* RAID Configuration PageAddress format */
 360 #define MPI2_RAID_PGAD_FORM_MASK                   (0xF0000000)
 361 #define MPI2_RAID_PGAD_FORM_GET_NEXT_CONFIGNUM     (0x00000000)
 362 #define MPI2_RAID_PGAD_FORM_CONFIGNUM              (0x10000000)
 363 #define MPI2_RAID_PGAD_FORM_ACTIVE_CONFIG          (0x20000000)

 365 #define MPI2_RAID_PGAD_CONFIGNUM_MASK              (0x000000FF)


 368 /* Driver Persistent Mapping PageAddress format */
 369 #define MPI2_DPM_PGAD_FORM_MASK                    (0xF0000000)
 370 #define MPI2_DPM_PGAD_FORM_ENTRY_RANGE             (0x00000000)

 372 #define MPI2_DPM_PGAD_ENTRY_COUNT_MASK             (0x0FFF0000)
 373 #define MPI2_DPM_PGAD_ENTRY_COUNT_SHIFT            (16)
 374 #define MPI2_DPM_PGAD_START_ENTRY_MASK             (0x0000FFFF)


 377 /* Ethernet PageAddress format */
 378 #define MPI2_ETHERNET_PGAD_FORM_MASK               (0xF0000000)
 379 #define MPI2_ETHERNET_PGAD_FORM_IF_NUM             (0x00000000)
```

```
#define MPI2_ETHERNET_PGAD_IF_NUMBER_MASK           (0x000000FF)



/*****************************************************************************
*   Configuration messages
*****************************************************************************/

/* Configuration Request Message */
typedef struct _MPI2_CONFIG_REQUEST
{
    U8                      Action;                 /* 0x00 */
    U8                      SGLFlags;               /* 0x01 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     ExtPageLength;          /* 0x04 */
    U8                      ExtPageType;            /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved1;              /* 0x0A */
    U8                      Reserved2;              /* 0x0C */
    U8                      ProxyVF_ID;             /* 0x0D */
    U16                     Reserved4;              /* 0x0E */
    U32                     Reserved2;              /* 0x0C */
    U32                     Reserved3;              /* 0x10 */
    MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x14 */
    U32                     PageAddress;            /* 0x18 */
    MPI2_SGE_IO_UNION       PageBufferSGE;          /* 0x1C */
} MPI2_CONFIG_REQUEST, MPI2_POINTER PTR_MPI2_CONFIG_REQUEST,
  Mpi2ConfigRequest_t, MPI2_POINTER pMpi2ConfigRequest_t;

/* values for the Action field */
#define MPI2_CONFIG_ACTION_PAGE_HEADER              (0x00)
#define MPI2_CONFIG_ACTION_PAGE_READ_CURRENT        (0x01)
#define MPI2_CONFIG_ACTION_PAGE_WRITE_CURRENT       (0x02)
#define MPI2_CONFIG_ACTION_PAGE_DEFAULT             (0x03)
#define MPI2_CONFIG_ACTION_PAGE_WRITE_NVRAM         (0x04)
#define MPI2_CONFIG_ACTION_PAGE_READ_DEFAULT        (0x05)
#define MPI2_CONFIG_ACTION_PAGE_READ_NVRAM          (0x06)
#define MPI2_CONFIG_ACTION_PAGE_GET_CHANGEABLE      (0x07)

/* use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */
/* values for SGLFlags field are in the SGL section of mpi2.h */


/* Config Reply Message */
typedef struct _MPI2_CONFIG_REPLY
{
    U8                      Action;                 /* 0x00 */
    U8                      SGLFlags;               /* 0x01 */
    U8                      MsgLength;              /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     ExtPageLength;          /* 0x04 */
    U8                      ExtPageType;            /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved1;              /* 0x0A */
    U16                     Reserved2;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
    MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x14 */
} MPI2_CONFIG_REPLY, MPI2_POINTER PTR_MPI2_CONFIG_REPLY,
  Mpi2ConfigReply_t, MPI2_POINTER pMpi2ConfigReply_t;
```



```
/*****************************************************************************
*
*               C o n f i g u r a t i o n    P a g e s
*
*****************************************************************************/

/*****************************************************************************
*   Manufacturing Config pages
*****************************************************************************/

#define MPI2_MFGPAGE_VENDORID_LSI                   (0x1000)

/* MPI v2.0 SAS products */
/* SAS */
#define MPI2_MFGPAGE_DEVID_SAS2004                  (0x0070)
#define MPI2_MFGPAGE_DEVID_SAS2008                  (0x0072)
#define MPI2_MFGPAGE_DEVID_SAS2108_1                (0x0074)
#define MPI2_MFGPAGE_DEVID_SAS2108_2                (0x0076)
#define MPI2_MFGPAGE_DEVID_SAS2108_3                (0x0077)
#define MPI2_MFGPAGE_DEVID_SAS2116_1                (0x0064)
#define MPI2_MFGPAGE_DEVID_SAS2116_2                (0x0065)

#define MPI2_MFGPAGE_DEVID_SSS6200                  (0x007E)

#define MPI2_MFGPAGE_DEVID_SAS2208_1                (0x0080)
#define MPI2_MFGPAGE_DEVID_SAS2208_2                (0x0081)
#define MPI2_MFGPAGE_DEVID_SAS2208_3                (0x0082)
#define MPI2_MFGPAGE_DEVID_SAS2208_4                (0x0083)
#define MPI2_MFGPAGE_DEVID_SAS2208_5                (0x0084)
#define MPI2_MFGPAGE_DEVID_SAS2208_6                (0x0085)
#define MPI2_MFGPAGE_DEVID_SAS2308_1                (0x0086)
#define MPI2_MFGPAGE_DEVID_SAS2308_2                (0x0087)
#define MPI2_MFGPAGE_DEVID_SAS2308_3                (0x006E)
#define MPI2_MFGPAGE_DEVID_SAS2208_7                (0x0086)
#define MPI2_MFGPAGE_DEVID_SAS2208_8                (0x0087)

/* MPI v2.5 SAS products */
#define MPI25_MFGPAGE_DEVID_SAS3004                 (0x0096)
#define MPI25_MFGPAGE_DEVID_SAS3008                 (0x0097)
#define MPI25_MFGPAGE_DEVID_SAS3108_1               (0x0090)
#define MPI25_MFGPAGE_DEVID_SAS3108_2               (0x0091)
#define MPI25_MFGPAGE_DEVID_SAS3108_5               (0x0094)
#define MPI25_MFGPAGE_DEVID_SAS3108_6               (0x0095)



/* Manufacturing Page 0 */

typedef struct _MPI2_CONFIG_PAGE_MAN_0
{
    MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x00 */
    U8                      ChipName[16];           /* 0x04 */
    U8                      ChipRevision[8];        /* 0x14 */
    U8                      BoardName[16];          /* 0x1C */
    U8                      BoardAssembly[16];      /* 0x2C */
    U8                      BoardTracerNumber[16];  /* 0x3C */
} MPI2_CONFIG_PAGE_MAN_0,
      unchanged_portion_omitted
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_MAN_4,
  Mpi2ManufacturingPage4_t, MPI2_POINTER pMpi2ManufacturingPage4_t;

#define MPI2_MANUFACTURING4_PAGEVERSION             (0x0A)
```

```
/* Manufacturing Page 4 Flags field */
#define MPI2_MANPAGE4_METADATA_SIZE_MASK                (0x00030000)
#define MPI2_MANPAGE4_METADATA_512MB                    (0x00000000)

#define MPI2_MANPAGE4_MIX_SSD_SAS_SATA                  (0x00008000)
#define MPI2_MANPAGE4_MIX_SSD_AND_NON_SSD               (0x00004000)
#define MPI2_MANPAGE4_HIDE_PHYSDISK_NON_IR              (0x00002000)

#define MPI2_MANPAGE4_MASK_PHYSDISK_COERCION            (0x00001C00)
#define MPI2_MANPAGE4_PHYSDISK_COERCION_1GB             (0x00000000)
#define MPI2_MANPAGE4_PHYSDISK_128MB_COERCION           (0x00000400)
#define MPI2_MANPAGE4_PHYSDISK_ADAPTIVE_COERCION        (0x00000800)
#define MPI2_MANPAGE4_PHYSDISK_ZERO_COERCION            (0x00000C00)

#define MPI2_MANPAGE4_MASK_BAD_BLOCK_MARKING            (0x00000300)
#define MPI2_MANPAGE4_DEFAULT_BAD_BLOCK_MARKING         (0x00000000)
#define MPI2_MANPAGE4_TABLE_BAD_BLOCK_MARKING           (0x00000100)
#define MPI2_MANPAGE4_WRITE_LONG_BAD_BLOCK_MARKING      (0x00000200)

#define MPI2_MANPAGE4_FORCE_OFFLINE_FAILOVER            (0x00000080)
#define MPI2_MANPAGE4_RAID10_DISABLE                    (0x00000040)
#define MPI2_MANPAGE4_RAID1E_DISABLE                    (0x00000020)
#define MPI2_MANPAGE4_RAID1_DISABLE                     (0x00000010)
#define MPI2_MANPAGE4_RAID0_DISABLE                     (0x00000008)
#define MPI2_MANPAGE4_IR_MODEPAGE8_DISABLE              (0x00000004)
#define MPI2_MANPAGE4_IM_RESYNC_CACHE_ENABLE            (0x00000002)
#define MPI2_MANPAGE4_IR_NO_MIX_SAS_SATA                (0x00000001)


/* Manufacturing Page 5 */

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhys at runtime.
 * one and check Header.PageLength or NumPhys at runtime.
 */
#ifndef MPI2_MAN_PAGE_5_PHY_ENTRIES
#define MPI2_MAN_PAGE_5_PHY_ENTRIES        (1)
#endif

typedef struct _MPI2_MANUFACTURING5_ENTRY
{
    U64                                 WWID;           /* 0x00 */
    U64                                 DeviceName;     /* 0x08 */
} MPI2_MANUFACTURING5_ENTRY, MPI2_POINTER PTR_MPI2_MANUFACTURING5_ENTRY,
_____unchanged_portion_omitted_
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_MAN_6,
  Mpi2ManufacturingPage6_t, MPI2_POINTER pMpi2ManufacturingPage6_t;

#define MPI2_MANUFACTURING6_PAGEVERSION                 (0x00)


/* Manufacturing Page 7 */

typedef struct _MPI2_MANPAGE7_CONNECTOR_INFO
{
    U32                         Pinout;                 /* 0x00 */
    U8                          Connector[16];          /* 0x04 */
    U8                          Location;               /* 0x14 */
    U8                          ReceptacleID;           /* 0x15 */
    U8                          Reserved1;              /* 0x15 */
    U16                         Slot;                   /* 0x16 */
    U32                         Reserved2;              /* 0x18 */
} MPI2_MANPAGE7_CONNECTOR_INFO, MPI2_POINTER PTR_MPI2_MANPAGE7_CONNECTOR_INFO,
  Mpi2ManPage7ConnectorInfo_t, MPI2_POINTER pMpi2ManPage7ConnectorInfo_t;
```



```
/* defines for the Pinout field */
#define MPI2_MANPAGE7_PINOUT_LANE_MASK                  (0x0000FF00)
#define MPI2_MANPAGE7_PINOUT_LANE_SHIFT                 (8)
#define MPI2_MANPAGE7_PINOUT_SFF_8484_L4                (0x00080000)
#define MPI2_MANPAGE7_PINOUT_SFF_8484_L3                (0x00040000)
#define MPI2_MANPAGE7_PINOUT_SFF_8484_L2                (0x00020000)
#define MPI2_MANPAGE7_PINOUT_SFF_8484_L1                (0x00010000)
#define MPI2_MANPAGE7_PINOUT_SFF_8470_L4                (0x00000800)
#define MPI2_MANPAGE7_PINOUT_SFF_8470_L3                (0x00000400)
#define MPI2_MANPAGE7_PINOUT_SFF_8470_L2                (0x00000200)
#define MPI2_MANPAGE7_PINOUT_SFF_8470_L1                (0x00000100)
#define MPI2_MANPAGE7_PINOUT_SFF_8482                   (0x00000002)
#define MPI2_MANPAGE7_PINOUT_CONNECTION_UNKNOWN         (0x00000001)

#define MPI2_MANPAGE7_PINOUT_TYPE_MASK                  (0x000000FF)
#define MPI2_MANPAGE7_PINOUT_TYPE_UNKNOWN               (0x00)
#define MPI2_MANPAGE7_PINOUT_SATA_SINGLE                (0x01)
#define MPI2_MANPAGE7_PINOUT_SFF_8482                   (0x02)
#define MPI2_MANPAGE7_PINOUT_SFF_8486                   (0x03)
#define MPI2_MANPAGE7_PINOUT_SFF_8484                   (0x04)
#define MPI2_MANPAGE7_PINOUT_SFF_8087                   (0x05)
#define MPI2_MANPAGE7_PINOUT_SFF_8643_4I                (0x06)
#define MPI2_MANPAGE7_PINOUT_SFF_8643_8I                (0x07)
#define MPI2_MANPAGE7_PINOUT_SFF_8470                   (0x08)
#define MPI2_MANPAGE7_PINOUT_SFF_8088                   (0x09)
#define MPI2_MANPAGE7_PINOUT_SFF_8644_4X                (0x0A)
#define MPI2_MANPAGE7_PINOUT_SFF_8644_8X                (0x0B)
#define MPI2_MANPAGE7_PINOUT_SFF_8644_16X               (0x0C)
#define MPI2_MANPAGE7_PINOUT_SFF_8436                   (0x0D)

/* defines for the Location field */
#define MPI2_MANPAGE7_LOCATION_UNKNOWN                  (0x01)
#define MPI2_MANPAGE7_LOCATION_INTERNAL                 (0x02)
#define MPI2_MANPAGE7_LOCATION_EXTERNAL                 (0x04)
#define MPI2_MANPAGE7_LOCATION_SWITCHABLE               (0x08)
#define MPI2_MANPAGE7_LOCATION_AUTO                     (0x10)
#define MPI2_MANPAGE7_LOCATION_NOT_PRESENT              (0x20)
#define MPI2_MANPAGE7_LOCATION_NOT_CONNECTED            (0x80)

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhys at runtime.
 * one and check NumPhys at runtime.
 */
#ifndef MPI2_MANPAGE7_CONNECTOR_INFO_MAX
#define MPI2_MANPAGE7_CONNECTOR_INFO_MAX  (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_MAN_7
{
    MPI2_CONFIG_PAGE_HEADER         Header;             /* 0x00 */
    U32                             Reserved1;          /* 0x04 */
    U32                             Reserved2;          /* 0x08 */
    U32                             Flags;              /* 0x0C */
    U8                              EnclosureName[16];  /* 0x10 */
    U8                              NumPhys;            /* 0x20 */
    U8                              Reserved3;          /* 0x21 */
    U16                             Reserved4;          /* 0x22 */
    MPI2_MANPAGE7_CONNECTOR_INFO    ConnectorInfo[MPI2_MANPAGE7_CONNECTOR_INFO_M
} MPI2_CONFIG_PAGE_MAN_7,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_MAN_7,
  Mpi2ManufacturingPage7_t, MPI2_POINTER pMpi2ManufacturingPage7_t;

#define MPI2_MANUFACTURING7_PAGEVERSION                 (0x01)
#define MPI2_MANUFACTURING7_PAGEVERSION                 (0x00)
```

```
763 /* defines for the Flags field */
764 #define MPI2_MANPAGE7_FLAG_BASE_ENCLOSURE_LEVEL         (0x00000008)
765 #define MPI2_MANPAGE7_FLAG_EVENTREPLAY_SLOT_ORDER       (0x00000002)
766 #define MPI2_MANPAGE7_FLAG_USE_SLOT_INFO                (0x00000001)


769 /*
770  * Generic structure to use for product-specific manufacturing pages
771  * (currently Manufacturing Page 8 through Manufacturing Page 31).
772  */

774 typedef struct _MPI2_CONFIG_PAGE_MAN_PS
775 {
776     MPI2_CONFIG_PAGE_HEADER         Header;             /* 0x00 */
777     U32                             ProductSpecificInfo;/* 0x04 */
778 } MPI2_CONFIG_PAGE_MAN_PS,
      unchanged_portion_omitted_
833   Mpi2IOUnitPage1_t, MPI2_POINTER pMpi2IOUnitPage1_t;

835 #define MPI2_IOUNITPAGE1_PAGEVERSION                    (0x04)

837 /* IO Unit Page 1 Flags defines */
838 #define MPI2_IOUNITPAGE1_ATA_SECURITY_FREEZE_LOCK       (0x00004000)
839 #define MPI25_IOUNITPAGE1_NEW_DEVICE_FAST_PATH_DISABLE  (0x00002000)
840 #define MPI25_IOUNITPAGE1_DISABLE_FAST_PATH             (0x00001000)
841 #define MPI2_IOUNITPAGE1_ENABLE_HOST_BASED_DISCOVERY    (0x00000800)
842 #define MPI2_IOUNITPAGE1_MASK_SATA_WRITE_CACHE          (0x00000600)
843 #define MPI2_IOUNITPAGE1_SATA_WRITE_CACHE_SHIFT         (9)
844 #define MPI2_IOUNITPAGE1_ENABLE_SATA_WRITE_CACHE        (0x00000000)
845 #define MPI2_IOUNITPAGE1_DISABLE_SATA_WRITE_CACHE       (0x00000200)
846 #define MPI2_IOUNITPAGE1_UNCHANGED_SATA_WRITE_CACHE     (0x00000400)
847 #define MPI2_IOUNITPAGE1_NATIVE_COMMAND_Q_DISABLE       (0x00000100)
848 #define MPI2_IOUNITPAGE1_DISABLE_IR                     (0x00000040)
849 #define MPI2_IOUNITPAGE1_DISABLE_TASK_SET_FULL_HANDLING (0x00000020)
850 #define MPI2_IOUNITPAGE1_IR_USE_STATIC_VOLUME_ID        (0x00000004)
768 #define MPI2_IOUNITPAGE1_MULTI_PATHING                  (0x00000002)
769 #define MPI2_IOUNITPAGE1_SINGLE_PATHING                 (0x00000000)


853 /* IO Unit Page 3 */

855 /*
856  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
857  * one and check the value returned for GPIOCount at runtime.
776  * one and check Header.PageLength at runtime.
858  */
859 #ifndef MPI2_IO_UNIT_PAGE_3_GPIO_VAL_MAX
860 #define MPI2_IO_UNIT_PAGE_3_GPIO_VAL_MAX    (1)
861 #endif

863 typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_3
864 {
865     MPI2_CONFIG_PAGE_HEADER Header;                             /* 0x00 */
866     U8                      GPIOCount;                          /* 0x04 */
867     U8                      Reserved1;                          /* 0x05 */
868     U16                     Reserved2;                          /* 0x06 */
869     U16                     GPIOVal[MPI2_IO_UNIT_PAGE_3_GPIO_VAL_MAX];/* 0x08 */
870 } MPI2_CONFIG_PAGE_IO_UNIT_3, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_3,
871   Mpi2IOUnitPage3_t, MPI2_POINTER pMpi2IOUnitPage3_t;

873 #define MPI2_IOUNITPAGE3_PAGEVERSION                    (0x01)

875 /* defines for IO Unit Page 3 GPIOVal field */
876 #define MPI2_IOUNITPAGE3_GPIO_FUNCTION_MASK             (0xFFFC)
877 #define MPI2_IOUNITPAGE3_GPIO_FUNCTION_SHIFT            (2)
878 #define MPI2_IOUNITPAGE3_GPIO_SETTING_OFF               (0x0000)
```



```
879 #define MPI2_IOUNITPAGE3_GPIO_SETTING_ON                (0x0001)


882 /* IO Unit Page 5 */

884 /*
885  * Upper layer code (drivers, utilities, etc.) should leave this define set to
886  * one and check the value returned for NumDmaEngines at runtime.
805  * one and check Header.PageLength or NumDmaEngines at runtime.
887  */
888 #ifndef MPI2_IOUNITPAGE5_DMAENGINE_ENTRIES
889 #define MPI2_IOUNITPAGE5_DMAENGINE_ENTRIES      (1)
890 #endif

892 typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_5
893 {
894     MPI2_CONFIG_PAGE_HEADER Header;                                 /* 0x00
895     U64                     RaidAcceleratorBufferBaseAddress;       /* 0x04
896     U64                     RaidAcceleratorBufferSize;              /* 0x0C
897     U64                     RaidAcceleratorControlBaseAddress;      /* 0x14
898     U8                      RAControlSize;                          /* 0x1C
899     U8                      NumDmaEngines;                          /* 0x1D
900     U8                      RAMinControlSize;                       /* 0x1E
901     U8                      RAMaxControlSize;                       /* 0x1F
902     U32                     Reserved1;                              /* 0x20
903     U32                     Reserved2;                              /* 0x24
904     U32                     Reserved3;                              /* 0x28
905     U32                     DmaEngineCapabilities[MPI2_IOUNITPAGE5_DMAENGINE_ENT
906 } MPI2_CONFIG_PAGE_IO_UNIT_5, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_5,
907   Mpi2IOUnitPage5_t, MPI2_POINTER pMpi2IOUnitPage5_t;

909 #define MPI2_IOUNITPAGE5_PAGEVERSION                    (0x00)

911 /* defines for IO Unit Page 5 DmaEngineCapabilities field */
912 #define MPI2_IOUNITPAGE5_DMA_CAP_MASK_MAX_REQUESTS      (0xFFFF0000)
831 #define MPI2_IOUNITPAGE5_DMA_CAP_MASK_MAX_REQUESTS      (0xFF00)
913 #define MPI2_IOUNITPAGE5_DMA_CAP_SHIFT_MAX_REQUESTS     (16)

915 #define MPI2_IOUNITPAGE5_DMA_CAP_EEDP                   (0x0008)
916 #define MPI2_IOUNITPAGE5_DMA_CAP_PARITY_GENERATION      (0x0004)
917 #define MPI2_IOUNITPAGE5_DMA_CAP_HASHING                (0x0002)
918 #define MPI2_IOUNITPAGE5_DMA_CAP_ENCRYPTION             (0x0001)


921 /* IO Unit Page 6 */

923 typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_6
924 {
925     MPI2_CONFIG_PAGE_HEADER Header;                                 /* 0x00 */
926     U16                     Flags;                                  /* 0x04 */
927     U8                      RAHostControlSize;                      /* 0x06 */
928     U8                      Reserved0;                              /* 0x07 */
929     U64                     RaidAcceleratorHostControlBaseAddress;  /* 0x08 */
930     U32                     Reserved1;                              /* 0x10 */
931     U32                     Reserved2;                              /* 0x14 */
932     U32                     Reserved3;                              /* 0x18 */
933 } MPI2_CONFIG_PAGE_IO_UNIT_6, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_6,
934   Mpi2IOUnitPage6_t, MPI2_POINTER pMpi2IOUnitPage6_t;

936 #define MPI2_IOUNITPAGE6_PAGEVERSION                    (0x00)

938 /* defines for IO Unit Page 6 Flags field */
939 #define MPI2_IOUNITPAGE6_FLAGS_ENABLE_RAID_ACCELERATOR  (0x0001)


942 /* IO Unit Page 7 */
```

```
 944 typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_7
 945 {
 946     MPI2_CONFIG_PAGE_HEADER Header;                         /* 0x00 */
 947     U8                      CurrentPowerMode;               /* 0x04 */ /
 948     U8                      PreviousPowerMode;              /* 0x05 */ /
 866     U16                     Reserved1;                      /* 0x04 */
 949     U8                      PCIeWidth;                      /* 0x06 */
 950     U8                      PCIeSpeed;                      /* 0x07 */
 951     U32                     ProcessorState;                 /* 0x08 */
 952     U32                     PowerManagementCapabilities;    /* 0x0C */
 870     U32                     Reserved2;                      /* 0x0C */
 953     U16                     IOCTemperature;                 /* 0x10 */
 954     U8                      IOCTemperatureUnits;            /* 0x12 */
 955     U8                      IOCSpeed;                       /* 0x13 */
 956     U16                     BoardTemperature;               /* 0x14 */
 957     U8                      BoardTemperatureUnits;          /* 0x16 */
 958     U8                      Reserved3;                      /* 0x17 */
 959     U32                     Reserved4;                      /* 0x18 */
 960     U32                     Reserved5;                      /* 0x1C */
 961     U32                     Reserved6;                      /* 0x20 */
 962     U32                     Reserved7;                      /* 0x24 */
 874     U32                     Reserved3;                      /* 0x14 */
 963 } MPI2_CONFIG_PAGE_IO_UNIT_7, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_7,
 964   Mpi2IOUnitPage7_t, MPI2_POINTER pMpi2IOUnitPage7_t;

 966 #define MPI2_IOUNITPAGE7_PAGEVERSION                (0x04)
 878 #define MPI2_IOUNITPAGE7_PAGEVERSION                (0x00)

 968 /* defines for IO Unit Page 7 CurrentPowerMode and PreviousPowerMode fields */
 969 #define MPI25_IOUNITPAGE7_PM_INIT_MASK              (0xC0)
 970 #define MPI25_IOUNITPAGE7_PM_INIT_UNAVAILABLE       (0x00)
 971 #define MPI25_IOUNITPAGE7_PM_INIT_HOST              (0x40)
 972 #define MPI25_IOUNITPAGE7_PM_INIT_IO_UNIT           (0x80)
 973 #define MPI25_IOUNITPAGE7_PM_INIT_PCIE_DPA          (0xC0)

 975 #define MPI25_IOUNITPAGE7_PM_MODE_MASK              (0x07)
 976 #define MPI25_IOUNITPAGE7_PM_MODE_UNAVAILABLE       (0x00)
 977 #define MPI25_IOUNITPAGE7_PM_MODE_UNKNOWN           (0x01)
 978 #define MPI25_IOUNITPAGE7_PM_MODE_FULL_POWER        (0x04)
 979 #define MPI25_IOUNITPAGE7_PM_MODE_REDUCED_POWER     (0x05)
 980 #define MPI25_IOUNITPAGE7_PM_MODE_STANDBY           (0x06)


 983 /* defines for IO Unit Page 7 PCIeWidth field */
 984 #define MPI2_IOUNITPAGE7_PCIE_WIDTH_X1              (0x01)
 985 #define MPI2_IOUNITPAGE7_PCIE_WIDTH_X2              (0x02)
 986 #define MPI2_IOUNITPAGE7_PCIE_WIDTH_X4              (0x04)
 987 #define MPI2_IOUNITPAGE7_PCIE_WIDTH_X8              (0x08)

 989 /* defines for IO Unit Page 7 PCIeSpeed field */
 990 #define MPI2_IOUNITPAGE7_PCIE_SPEED_2_5_GBPS        (0x00)
 991 #define MPI2_IOUNITPAGE7_PCIE_SPEED_5_0_GBPS        (0x01)
 992 #define MPI2_IOUNITPAGE7_PCIE_SPEED_8_0_GBPS        (0x02)

 994 /* defines for IO Unit Page 7 ProcessorState field */
 995 #define MPI2_IOUNITPAGE7_PSTATE_MASK_SECOND         (0x0000000F)
 996 #define MPI2_IOUNITPAGE7_PSTATE_SHIFT_SECOND        (0)

 998 #define MPI2_IOUNITPAGE7_PSTATE_NOT_PRESENT         (0x00)
 999 #define MPI2_IOUNITPAGE7_PSTATE_DISABLED            (0x01)
1000 #define MPI2_IOUNITPAGE7_PSTATE_ENABLED             (0x02)

1002 /* defines for IO Unit Page 7 PowerManagementCapabilities field */
1003 #define MPI25_IOUNITPAGE7_PMCAP_DPA_FULL_PWR_MODE       (0x00400000)
1004 #define MPI25_IOUNITPAGE7_PMCAP_DPA_REDUCED_PWR_MODE    (0x00200000)
```



```
1005 #define MPI25_IOUNITPAGE7_PMCAP_DPA_STANDBY_MODE        (0x00100000)
1006 #define MPI25_IOUNITPAGE7_PMCAP_HOST_FULL_PWR_MODE      (0x00040000)
1007 #define MPI25_IOUNITPAGE7_PMCAP_HOST_REDUCED_PWR_MODE   (0x00020000)
1008 #define MPI25_IOUNITPAGE7_PMCAP_HOST_STANDBY_MODE       (0x00010000)
1009 #define MPI25_IOUNITPAGE7_PMCAP_IO_FULL_PWR_MODE        (0x00004000)
1010 #define MPI25_IOUNITPAGE7_PMCAP_IO_REDUCED_PWR_MODE     (0x00002000)
1011 #define MPI25_IOUNITPAGE7_PMCAP_IO_STANDBY_MODE         (0x00001000)
1012 #define MPI2_IOUNITPAGE7_PMCAP_HOST_12_5_PCT_IOCSPEED   (0x00000400)
1013 #define MPI2_IOUNITPAGE7_PMCAP_HOST_25_0_PCT_IOCSPEED   (0x00000200)
1014 #define MPI2_IOUNITPAGE7_PMCAP_HOST_50_0_PCT_IOCSPEED   (0x00000100)
1015 #define MPI25_IOUNITPAGE7_PMCAP_IO_12_5_PCT_IOCSPEED    (0x00000040)
1016 #define MPI25_IOUNITPAGE7_PMCAP_IO_25_0_PCT_IOCSPEED    (0x00000020)
1017 #define MPI25_IOUNITPAGE7_PMCAP_IO_50_0_PCT_IOCSPEED    (0x00000010)
1018 #define MPI2_IOUNITPAGE7_PMCAP_HOST_WIDTH_CHANGE_PCIE   (0x00000008) /* obsolete
1019 #define MPI2_IOUNITPAGE7_PMCAP_HOST_SPEED_CHANGE_PCIE   (0x00000004) /* obsolete
1020 #define MPI25_IOUNITPAGE7_PMCAP_IO_WIDTH_CHANGE_PCIE    (0x00000002) /* obsolete
1021 #define MPI25_IOUNITPAGE7_PMCAP_IO_SPEED_CHANGE_PCIE    (0x00000001) /* obsolete

1023 /* obsolete names for the PowerManagementCapabilities bits (above) */
1024 #define MPI2_IOUNITPAGE7_PMCAP_12_5_PCT_IOCSPEED    (0x00000400)
1025 #define MPI2_IOUNITPAGE7_PMCAP_25_0_PCT_IOCSPEED    (0x00000200)
1026 #define MPI2_IOUNITPAGE7_PMCAP_50_0_PCT_IOCSPEED    (0x00000100)
1027 #define MPI2_IOUNITPAGE7_PMCAP_PCIE_WIDTH_CHANGE    (0x00000008) /* obsolete */
1028 #define MPI2_IOUNITPAGE7_PMCAP_PCIE_SPEED_CHANGE    (0x00000004) /* obsolete */


1031 /* defines for IO Unit Page 7 IOCTemperatureUnits field */
1032 #define MPI2_IOUNITPAGE7_IOC_TEMP_NOT_PRESENT       (0x00)
1033 #define MPI2_IOUNITPAGE7_IOC_TEMP_FAHRENHEIT        (0x01)
1034 #define MPI2_IOUNITPAGE7_IOC_TEMP_CELSIUS           (0x02)

1036 /* defines for IO Unit Page 7 IOCSpeed field */
1037 #define MPI2_IOUNITPAGE7_IOC_SPEED_FULL             (0x01)
1038 #define MPI2_IOUNITPAGE7_IOC_SPEED_HALF             (0x02)
1039 #define MPI2_IOUNITPAGE7_IOC_SPEED_QUARTER          (0x04)
1040 #define MPI2_IOUNITPAGE7_IOC_SPEED_EIGHTH           (0x08)

1042 /* defines for IO Unit Page 7 BoardTemperatureUnits field */
1043 #define MPI2_IOUNITPAGE7_BOARD_TEMP_NOT_PRESENT     (0x00)
1044 #define MPI2_IOUNITPAGE7_BOARD_TEMP_FAHRENHEIT      (0x01)
1045 #define MPI2_IOUNITPAGE7_BOARD_TEMP_CELSIUS         (0x02)


1048 /* IO Unit Page 8 */

1050 #define MPI2_IOUNIT8_NUM_THRESHOLDS     (4)

1052 typedef struct _MPI2_IOUNIT8_SENSOR
1053 {
1054     U16                     Flags;                              /* 0x00 */
1055     U16                     Reserved1;                          /* 0x02 */
1056     U16                     Threshold[MPI2_IOUNIT8_NUM_THRESHOLDS]; /* 0x04 */
1057     U32                     Reserved2;                          /* 0x0C */
1058     U32                     Reserved3;                          /* 0x10 */
1059     U32                     Reserved4;                          /* 0x14 */
1060 } MPI2_IOUNIT8_SENSOR, MPI2_POINTER PTR_MPI2_IOUNIT8_SENSOR,
1061   Mpi2IOUnit8Sensor_t, MPI2_POINTER pMpi2IOUnit8Sensor_t;

1063 /* defines for IO Unit Page 8 Sensor Flags field */
1064 #define MPI2_IOUNIT8_SENSOR_FLAGS_T3_ENABLE         (0x0008)
1065 #define MPI2_IOUNIT8_SENSOR_FLAGS_T2_ENABLE         (0x0004)
1066 #define MPI2_IOUNIT8_SENSOR_FLAGS_T1_ENABLE         (0x0002)
1067 #define MPI2_IOUNIT8_SENSOR_FLAGS_T0_ENABLE         (0x0001)

1069 /*
1070  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
```

```
 * one and check the value returned for NumSensors at runtime.
 */
#ifndef MPI2_IOUNITPAGE8_SENSOR_ENTRIES
#define MPI2_IOUNITPAGE8_SENSOR_ENTRIES     (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_8
{
    MPI2_CONFIG_PAGE_HEADER Header;                             /* 0x00 */
    U32                     Reserved1;                          /* 0x04 */
    U32                     Reserved2;                          /* 0x08 */
    U8                      NumSensors;                         /* 0x0C */
    U8                      PollingInterval;                    /* 0x0D */
    U16                     Reserved3;                          /* 0x0E */
    MPI2_IOUNIT8_SENSOR     Sensor[MPI2_IOUNITPAGE8_SENSOR_ENTRIES];/* 0x10 */
} MPI2_CONFIG_PAGE_IO_UNIT_8, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_8,
  Mpi2IOUnitPage8_t, MPI2_POINTER pMpi2IOUnitPage8_t;

#define MPI2_IOUNITPAGE8_PAGEVERSION                    (0x00)


/* IO Unit Page 9 */

typedef struct _MPI2_IOUNIT9_SENSOR
{
    U16                     CurrentTemperature;                 /* 0x00 */
    U16                     Reserved1;                          /* 0x02 */
    U8                      Flags;                              /* 0x04 */
    U8                      Reserved2;                          /* 0x05 */
    U16                     Reserved3;                          /* 0x06 */
    U32                     Reserved4;                          /* 0x08 */
    U32                     Reserved5;                          /* 0x0C */
} MPI2_IOUNIT9_SENSOR, MPI2_POINTER PTR_MPI2_IOUNIT9_SENSOR,
  Mpi2IOUnit9Sensor_t, MPI2_POINTER pMpi2IOUnit9Sensor_t;

/* defines for IO Unit Page 9 Sensor Flags field */
#define MPI2_IOUNIT9_SENSOR_FLAGS_TEMP_VALID        (0x01)

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumSensors at runtime.
 */
#ifndef MPI2_IOUNITPAGE9_SENSOR_ENTRIES
#define MPI2_IOUNITPAGE9_SENSOR_ENTRIES     (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_9
{
    MPI2_CONFIG_PAGE_HEADER Header;                             /* 0x00 */
    U32                     Reserved1;                          /* 0x04 */
    U32                     Reserved2;                          /* 0x08 */
    U8                      NumSensors;                         /* 0x0C */
    U8                      Reserved4;                          /* 0x0D */
    U16                     Reserved3;                          /* 0x0E */
    MPI2_IOUNIT9_SENSOR     Sensor[MPI2_IOUNITPAGE9_SENSOR_ENTRIES];/* 0x10 */
} MPI2_CONFIG_PAGE_IO_UNIT_9, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_9,
  Mpi2IOUnitPage9_t, MPI2_POINTER pMpi2IOUnitPage9_t;

#define MPI2_IOUNITPAGE9_PAGEVERSION                    (0x00)


/* IO Unit Page 10 */

typedef struct _MPI2_IOUNIT10_FUNCTION
{
    U8                      CreditPercent;      /* 0x00 */
```



```
    U8                      Reserved1;          /* 0x01 */
    U16                     Reserved2;          /* 0x02 */
} MPI2_IOUNIT10_FUNCTION, MPI2_POINTER PTR_MPI2_IOUNIT10_FUNCTION,
  Mpi2IOUnit10Function_t, MPI2_POINTER pMpi2IOUnit10Function_t;

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumFunctions at runtime.
 */
#ifndef MPI2_IOUNITPAGE10_FUNCTION_ENTRIES
#define MPI2_IOUNITPAGE10_FUNCTION_ENTRIES      (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_IO_UNIT_10
{
    MPI2_CONFIG_PAGE_HEADER Header;                                         /* 0
    U8                      NumFunctions;                                   /* 0
    U8                      Reserved1;                                      /* 0
    U16                     Reserved2;                                      /* 0
    U32                     Reserved3;                                      /* 0
    U32                     Reserved4;                                      /* 0
    MPI2_IOUNIT10_FUNCTION  Function[MPI2_IOUNITPAGE10_FUNCTION_ENTRIES];   /* 0
} MPI2_CONFIG_PAGE_IO_UNIT_10, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IO_UNIT_10,
  Mpi2IOUnitPage10_t, MPI2_POINTER pMpi2IOUnitPage10_t;

#define MPI2_IOUNITPAGE10_PAGEVERSION                   (0x01)




/*****************************************************************************
*   IOC Config Pages
*****************************************************************************/

/* IOC Page 0 */

typedef struct _MPI2_CONFIG_PAGE_IOC_0
{
    MPI2_CONFIG_PAGE_HEADER Header;                     /* 0x00 */
    U32                     Reserved1;                  /* 0x04 */
    U32                     Reserved2;                  /* 0x08 */
    U16                     VendorID;                   /* 0x0C */
    U16                     DeviceID;                   /* 0x0E */
    U8                      RevisionID;                 /* 0x10 */
    U8                      Reserved3;                  /* 0x11 */
    U16                     Reserved4;                  /* 0x12 */
    U32                     ClassCode;                  /* 0x14 */
    U16                     SubsystemVendorID;          /* 0x18 */
    U16                     SubsystemID;                /* 0x1A */
} MPI2_CONFIG_PAGE_IOC_0, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IOC_0,
_____unchanged_portion_omitted_
  Mpi2IOCPage6_t, MPI2_POINTER pMpi2IOCPage6_t;

#define MPI2_IOCPAGE6_PAGEVERSION                       (0x05)
 997 #define MPI2_IOCPAGE6_PAGEVERSION                  (0x04)

/* defines for IOC Page 6 CapabilitiesFlags */
#define MPI2_IOCPAGE6_CAP_FLAGS_4K_SECTORS_SUPPORT      (0x00000020)
#define MPI2_IOCPAGE6_CAP_FLAGS_RAID10_SUPPORT          (0x00000010)
#define MPI2_IOCPAGE6_CAP_FLAGS_RAID1_SUPPORT           (0x00000008)
#define MPI2_IOCPAGE6_CAP_FLAGS_RAID1E_SUPPORT          (0x00000004)
#define MPI2_IOCPAGE6_CAP_FLAGS_RAID0_SUPPORT           (0x00000002)
#define MPI2_IOCPAGE6_CAP_FLAGS_GLOBAL_HOT_SPARE        (0x00000001)


/* IOC Page 7 */
```

```
1264 #define MPI2_IOCPAGE7_EVENTMASK_WORDS       (4)

1266 typedef struct _MPI2_CONFIG_PAGE_IOC_7
1267 {
1268     MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x00 */
1269     U32                     Reserved1;              /* 0x04 */
1270     U32                     EventMasks[MPI2_IOCPAGE7_EVENTMASK_WORDS];/* 0x08 */
1271     U16                     SASBroadcastPrimitiveMasks; /* 0x18 */
1272     U16                     SASNotifyPrimitiveMasks;    /* 0x1A */
1017     U16                     Reserved2;              /* 0x1A */
1273     U32                     Reserved3;              /* 0x1C */
1274 } MPI2_CONFIG_PAGE_IOC_7, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IOC_7,
1275   Mpi2IOCPage7_t, MPI2_POINTER pMpi2IOCPage7_t;

1277 #define MPI2_IOCPAGE7_PAGEVERSION                   (0x02)
1022 #define MPI2_IOCPAGE7_PAGEVERSION                   (0x01)


1280 /* IOC Page 8 */

1282 typedef struct _MPI2_CONFIG_PAGE_IOC_8
1283 {
1284     MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x00 */
1285     U8                      NumDevsPerEnclosure;    /* 0x04 */
1286     U8                      Reserved1;              /* 0x05 */
1287     U16                     Reserved2;              /* 0x06 */
1288     U16                     MaxPersistentEntries;   /* 0x08 */
1289     U16                     MaxNumPhysicalMappedIDs;/* 0x0A */
1290     U16                     Flags;                  /* 0x0C */
1291     U16                     Reserved3;              /* 0x0E */
1292     U16                     IRVolumeMappingFlags;   /* 0x10 */
1293     U16                     Reserved4;              /* 0x12 */
1294     U32                     Reserved5;              /* 0x14 */
1295 } MPI2_CONFIG_PAGE_IOC_8, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_IOC_8,
1296   Mpi2IOCPage8_t, MPI2_POINTER pMpi2IOCPage8_t;

1298 #define MPI2_IOCPAGE8_PAGEVERSION                   (0x00)

1300 /* defines for IOC Page 8 Flags field */
1301 #define MPI2_IOCPAGE8_FLAGS_DA_START_SLOT_1         (0x00000020)
1302 #define MPI2_IOCPAGE8_FLAGS_RESERVED_TARGETID_0     (0x00000010)

1304 #define MPI2_IOCPAGE8_FLAGS_MASK_MAPPING_MODE       (0x0000000E)
1305 #define MPI2_IOCPAGE8_FLAGS_DEVICE_PERSISTENCE_MAPPING  (0x00000000)
1306 #define MPI2_IOCPAGE8_FLAGS_ENCLOSURE_SLOT_MAPPING  (0x00000002)

1308 #define MPI2_IOCPAGE8_FLAGS_DISABLE_PERSISTENT_MAPPING  (0x00000001)
1309 #define MPI2_IOCPAGE8_FLAGS_ENABLE_PERSISTENT_MAPPING   (0x00000000)

1311 /* defines for IOC Page 8 IRVolumeMappingFlags */
1312 #define MPI2_IOCPAGE8_IRFLAGS_MASK_VOLUME_MAPPING_MODE  (0x00000003)
1313 #define MPI2_IOCPAGE8_IRFLAGS_LOW_VOLUME_MAPPING    (0x00000000)
1314 #define MPI2_IOCPAGE8_IRFLAGS_HIGH_VOLUME_MAPPING   (0x00000001)


1317 /****************************************************************************
1318 *   BIOS Config Pages
1319 ****************************************************************************/

1321 /* BIOS Page 1 */

1323 typedef struct _MPI2_CONFIG_PAGE_BIOS_1
1324 {
1325     MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x00 */
1326     U32                     BiosOptions;            /* 0x04 */
1327     U32                     IOCSettings;            /* 0x08 */
```



```
1328     U32                     Reserved1;              /* 0x0C */
1329     U32                     DeviceSettings;         /* 0x10 */
1330     U16                     NumberOfDevices;        /* 0x14 */
1331     U16                     UEFIVersion;            /* 0x16 */
1076     U16                     Reserved2;              /* 0x16 */
1332     U16                     IOTimeoutBlockDevicesNonRM; /* 0x18 */
1333     U16                     IOTimeoutSequential;    /* 0x1A */
1334     U16                     IOTimeoutOther;         /* 0x1C */
1335     U16                     IOTimeoutBlockDevicesRM;    /* 0x1E */
1336 } MPI2_CONFIG_PAGE_BIOS_1, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_BIOS_1,
1337   Mpi2BiosPage1_t, MPI2_POINTER pMpi2BiosPage1_t;

1339 #define MPI2_BIOSPAGE1_PAGEVERSION                  (0x05)
1084 #define MPI2_BIOSPAGE1_PAGEVERSION                  (0x04)

1341 /* values for BIOS Page 1 BiosOptions field */
1342 #define MPI2_BIOSPAGE1_OPTIONS_MASK_OEM_ID                  (0x000000F0)
1343 #define MPI2_BIOSPAGE1_OPTIONS_LSI_OEM_ID                   (0x00000000)

1345 #define MPI2_BIOSPAGE1_OPTIONS_MASK_UEFI_HII_REGISTRATION   (0x00000006)
1346 #define MPI2_BIOSPAGE1_OPTIONS_ENABLE_UEFI_HII              (0x00000000)
1347 #define MPI2_BIOSPAGE1_OPTIONS_DISABLE_UEFI_HII             (0x00000002)
1348 #define MPI2_BIOSPAGE1_OPTIONS_VERSION_CHECK_UEFI_HII       (0x00000004)

1350 #define MPI2_BIOSPAGE1_OPTIONS_DISABLE_BIOS                 (0x00000001)

1352 /* values for BIOS Page 1 IOCSettings field */
1353 #define MPI2_BIOSPAGE1_IOCSET_MASK_BOOT_PREFERENCE      (0x00030000)
1354 #define MPI2_BIOSPAGE1_IOCSET_ENCLOSURE_SLOT_BOOT       (0x00000000)
1355 #define MPI2_BIOSPAGE1_IOCSET_SAS_ADDRESS_BOOT          (0x00010000)

1357 #define MPI2_BIOSPAGE1_IOCSET_MASK_RM_SETTING           (0x000000C0)
1358 #define MPI2_BIOSPAGE1_IOCSET_NONE_RM_SETTING           (0x00000000)
1359 #define MPI2_BIOSPAGE1_IOCSET_BOOT_RM_SETTING           (0x00000040)
1360 #define MPI2_BIOSPAGE1_IOCSET_MEDIA_RM_SETTING          (0x00000080)

1362 #define MPI2_BIOSPAGE1_IOCSET_MASK_ADAPTER_SUPPORT      (0x00000030)
1363 #define MPI2_BIOSPAGE1_IOCSET_NO_SUPPORT                (0x00000000)
1364 #define MPI2_BIOSPAGE1_IOCSET_BIOS_SUPPORT              (0x00000010)
1365 #define MPI2_BIOSPAGE1_IOCSET_OS_SUPPORT                (0x00000020)
1366 #define MPI2_BIOSPAGE1_IOCSET_ALL_SUPPORT               (0x00000030)

1368 #define MPI2_BIOSPAGE1_IOCSET_ALTERNATE_CHS             (0x00000008)

1370 /* values for BIOS Page 1 DeviceSettings field */
1371 #define MPI2_BIOSPAGE1_DEVSET_DISABLE_SMART_POLLING     (0x00000010)
1372 #define MPI2_BIOSPAGE1_DEVSET_DISABLE_SEQ_LUN           (0x00000008)
1373 #define MPI2_BIOSPAGE1_DEVSET_DISABLE_RM_LUN            (0x00000004)
1374 #define MPI2_BIOSPAGE1_DEVSET_DISABLE_NON_RM_LUN        (0x00000002)
1375 #define MPI2_BIOSPAGE1_DEVSET_DISABLE_OTHER_LUN         (0x00000001)

1377 /* defines for BIOS Page 1 UEFIVersion field */
1378 #define MPI2_BIOSPAGE1_UEFI_VER_MAJOR_MASK              (0xFF00)
1379 #define MPI2_BIOSPAGE1_UEFI_VER_MAJOR_SHIFT             (8)
1380 #define MPI2_BIOSPAGE1_UEFI_VER_MINOR_MASK              (0x00FF)
1381 #define MPI2_BIOSPAGE1_UEFI_VER_MINOR_SHIFT             (0)


1385 /* BIOS Page 2 */

1387 typedef struct _MPI2_BOOT_DEVICE_ADAPTER_ORDER
1388 {
1389     U32         Reserved1;                          /* 0x00 */
1390     U32         Reserved2;                          /* 0x04 */
1391     U32         Reserved3;                          /* 0x08 */
```

```
1392     U32         Reserved4;                              /* 0x0C */
1393     U32         Reserved5;                              /* 0x10 */
1394     U32         Reserved6;                              /* 0x14 */
1395 } MPI2_BOOT_DEVICE_ADAPTER_ORDER,
_____unchanged_portion_omitted_
1493   Mpi2BiosPage3_t, MPI2_POINTER pMpi2BiosPage3_t;

1495 #define MPI2_BIOSPAGE3_PAGEVERSION                      (0x00)

1497 /* values for BIOS Page 3 GlobalFlags */
1498 #define MPI2_BIOSPAGE3_FLAGS_PAUSE_ON_ERROR             (0x00000002)
1499 #define MPI2_BIOSPAGE3_FLAGS_VERBOSE_ENABLE             (0x00000004)
1500 #define MPI2_BIOSPAGE3_FLAGS_HOOK_INT_40_DISABLE        (0x00000010)

1502 #define MPI2_BIOSPAGE3_FLAGS_DEV_LIST_DISPLAY_MASK      (0x000000E0)
1503 #define MPI2_BIOSPAGE3_FLAGS_INSTALLED_DEV_DISPLAY      (0x00000000)
1504 #define MPI2_BIOSPAGE3_FLAGS_ADAPTER_DISPLAY            (0x00000020)
1505 #define MPI2_BIOSPAGE3_FLAGS_ADAPTER_DEV_DISPLAY        (0x00000040)


1508 /* BIOS Page 4 */

1510 /*
1511  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
1512  * one and check the value returned for NumPhys at runtime.
1242  * one and check Header.PageLength or NumPhys at runtime.
1513  */
1514 #ifndef MPI2_BIOS_PAGE_4_PHY_ENTRIES
1515 #define MPI2_BIOS_PAGE_4_PHY_ENTRIES        (1)
1516 #endif

1518 typedef struct _MPI2_BIOS4_ENTRY
1519 {
1520     U64                     ReassignmentWWID;       /* 0x00 */
1521     U64                     ReassignmentDeviceName; /* 0x08 */
1522 } MPI2_BIOS4_ENTRY, MPI2_POINTER PTR_MPI2_BIOS4_ENTRY,
_____unchanged_portion_omitted_
1563   Mpi2RaidVol0Settings_t, MPI2_POINTER pMpi2RaidVol0Settings_t;

1565 /* RAID Volume Page 0 HotSparePool defines, also used in RAID Physical Disk */
1566 #define MPI2_RAID_HOT_SPARE_POOL_0                      (0x01)
1567 #define MPI2_RAID_HOT_SPARE_POOL_1                      (0x02)
1568 #define MPI2_RAID_HOT_SPARE_POOL_2                      (0x04)
1569 #define MPI2_RAID_HOT_SPARE_POOL_3                      (0x08)
1570 #define MPI2_RAID_HOT_SPARE_POOL_4                      (0x10)
1571 #define MPI2_RAID_HOT_SPARE_POOL_5                      (0x20)
1572 #define MPI2_RAID_HOT_SPARE_POOL_6                      (0x40)
1573 #define MPI2_RAID_HOT_SPARE_POOL_7                      (0x80)

1575 /* RAID Volume Page 0 VolumeSettings defines */
1576 #define MPI2_RAIDVOL0_SETTING_USE_PRODUCT_ID_SUFFIX     (0x0008)
1577 #define MPI2_RAIDVOL0_SETTING_AUTO_CONFIG_HSWAP_DISABLE (0x0004)

1579 #define MPI2_RAIDVOL0_SETTING_MASK_WRITE_CACHING        (0x0003)
1580 #define MPI2_RAIDVOL0_SETTING_UNCHANGED                 (0x0000)
1581 #define MPI2_RAIDVOL0_SETTING_DISABLE_WRITE_CACHING     (0x0001)
1582 #define MPI2_RAIDVOL0_SETTING_ENABLE_WRITE_CACHING      (0x0002)

1584 /*
1585  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
1586  * one and check the value returned for NumPhysDisks at runtime.
1316  * one and check Header.PageLength at runtime.
1587  */
1588 #ifndef MPI2_RAID_VOL_PAGE_0_PHYSDISK_MAX
1589 #define MPI2_RAID_VOL_PAGE_0_PHYSDISK_MAX       (1)
1590 #endif
```



```
1592 typedef struct _MPI2_CONFIG_PAGE_RAID_VOL_0
1593 {
1594     MPI2_CONFIG_PAGE_HEADER Header;                     /* 0x00 */
1595     U16                     DevHandle;                  /* 0x04 */
1596     U8                      VolumeState;                /* 0x06 */
1597     U8                      VolumeType;                 /* 0x07 */
1598     U32                     VolumeStatusFlags;          /* 0x08 */
1599     MPI2_RAIDVOL0_SETTINGS  VolumeSettings;             /* 0x0C */
1600     U64                     MaxLBA;                     /* 0x10 */
1601     U32                     StripeSize;                 /* 0x18 */
1602     U16                     BlockSize;                  /* 0x1C */
1603     U16                     Reserved1;                  /* 0x1E */
1604     U8                      SupportedPhysDisks;         /* 0x20 */
1605     U8                      ResyncRate;                 /* 0x21 */
1606     U16                     DataScrubDuration;          /* 0x22 */
1607     U8                      NumPhysDisks;               /* 0x24 */
1608     U8                      Reserved2;                  /* 0x25 */
1609     U8                      Reserved3;                  /* 0x26 */
1610     U8                      InactiveStatus;             /* 0x27 */
1611     MPI2_RAIDVOL0_PHYS_DISK PhysDisk[MPI2_RAID_VOL_PAGE_0_PHYSDISK_MAX]; /* 0x28
1612 } MPI2_CONFIG_PAGE_RAID_VOL_0, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_RAID_VOL_0,
1613   Mpi2RaidVolPage0_t, MPI2_POINTER pMpi2RaidVolPage0_t;

1615 #define MPI2_RAIDVOLPAGE0_PAGEVERSION           (0x0A)

1617 /* values for RAID VolumeState */
1618 #define MPI2_RAID_VOL_STATE_MISSING                         (0x00)
1619 #define MPI2_RAID_VOL_STATE_FAILED                          (0x01)
1620 #define MPI2_RAID_VOL_STATE_INITIALIZING                    (0x02)
1621 #define MPI2_RAID_VOL_STATE_ONLINE                          (0x03)
1622 #define MPI2_RAID_VOL_STATE_DEGRADED                        (0x04)
1623 #define MPI2_RAID_VOL_STATE_OPTIMAL                         (0x05)

1625 /* values for RAID VolumeType */
1626 #define MPI2_RAID_VOL_TYPE_RAID0                            (0x00)
1627 #define MPI2_RAID_VOL_TYPE_RAID1E                           (0x01)
1628 #define MPI2_RAID_VOL_TYPE_RAID1                            (0x02)
1629 #define MPI2_RAID_VOL_TYPE_RAID10                           (0x05)
1630 #define MPI2_RAID_VOL_TYPE_UNKNOWN                          (0xFF)

1632 /* values for RAID Volume Page 0 VolumeStatusFlags field */
1633 #define MPI2_RAIDVOL0_STATUS_FLAG_PENDING_RESYNC            (0x02000000)
1634 #define MPI2_RAIDVOL0_STATUS_FLAG_BACKG_INIT_PENDING        (0x01000000)
1635 #define MPI2_RAIDVOL0_STATUS_FLAG_MDC_PENDING               (0x00800000)
1636 #define MPI2_RAIDVOL0_STATUS_FLAG_USER_CONSIST_PENDING      (0x00400000)
1637 #define MPI2_RAIDVOL0_STATUS_FLAG_MAKE_DATA_CONSISTENT      (0x00200000)
1638 #define MPI2_RAIDVOL0_STATUS_FLAG_DATA_SCRUB                (0x00100000)
1639 #define MPI2_RAIDVOL0_STATUS_FLAG_CONSISTENCY_CHECK         (0x00080000)
1640 #define MPI2_RAIDVOL0_STATUS_FLAG_CAPACITY_EXPANSION        (0x00040000)
1641 #define MPI2_RAIDVOL0_STATUS_FLAG_BACKGROUND_INIT           (0x00020000)
1642 #define MPI2_RAIDVOL0_STATUS_FLAG_RESYNC_IN_PROGRESS        (0x00010000)
1643 #define MPI2_RAIDVOL0_STATUS_FLAG_VOL_NOT_CONSISTENT        (0x00000080)
1644 #define MPI2_RAIDVOL0_STATUS_FLAG_OCE_ALLOWED               (0x00000040)
1645 #define MPI2_RAIDVOL0_STATUS_FLAG_BGI_COMPLETE              (0x00000020)
1646 #define MPI2_RAIDVOL0_STATUS_FLAG_1E_OFFSET_MIRROR          (0x00000000)
1647 #define MPI2_RAIDVOL0_STATUS_FLAG_1E_ADJACENT_MIRROR        (0x00000010)
1648 #define MPI2_RAIDVOL0_STATUS_FLAG_BAD_BLOCK_TABLE_FULL      (0x00000008)
1649 #define MPI2_RAIDVOL0_STATUS_FLAG_VOLUME_INACTIVE           (0x00000004)
1650 #define MPI2_RAIDVOL0_STATUS_FLAG_QUIESCED                  (0x00000002)
1651 #define MPI2_RAIDVOL0_STATUS_FLAG_ENABLED                   (0x00000001)

1653 /* values for RAID Volume Page 0 SupportedPhysDisks field */
1654 #define MPI2_RAIDVOL0_SUPPORT_SOLID_STATE_DISKS             (0x08)
1655 #define MPI2_RAIDVOL0_SUPPORT_HARD_DISKS                    (0x04)
1656 #define MPI2_RAIDVOL0_SUPPORT_SAS_PROTOCOL                  (0x02)
```

```
#define MPI2_RAIDVOL0_SUPPORT_SATA_PROTOCOL               (0x01)

/* values for RAID Volume Page 0 InactiveStatus field */
#define MPI2_RAIDVOLPAGE0_UNKNOWN_INACTIVE                (0x00)
#define MPI2_RAIDVOLPAGE0_STALE_METADATA_INACTIVE         (0x01)
#define MPI2_RAIDVOLPAGE0_FOREIGN_VOLUME_INACTIVE         (0x02)
#define MPI2_RAIDVOLPAGE0_INSUFFICIENT_RESOURCE_INACTIVE  (0x03)
#define MPI2_RAIDVOLPAGE0_CLONE_VOLUME_INACTIVE           (0x04)
#define MPI2_RAIDVOLPAGE0_INSUFFICIENT_METADATA_INACTIVE  (0x05)
#define MPI2_RAIDVOLPAGE0_PREVIOUSLY_DELETED              (0x06)


/* RAID Volume Page 1 */

typedef struct _MPI2_CONFIG_PAGE_RAID_VOL_1
{
    MPI2_CONFIG_PAGE_HEADER Header;                 /* 0x00 */
    U16                     DevHandle;              /* 0x04 */
    U16                     Reserved0;              /* 0x06 */
    U8                      GUID[24];               /* 0x08 */
    U8                      Name[16];               /* 0x20 */
    U64                     WWID;                   /* 0x30 */
    U32                     Reserved1;              /* 0x38 */
    U32                     Reserved2;              /* 0x3C */
} MPI2_CONFIG_PAGE_RAID_VOL_1, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_RAID_VOL_1,
_____unchanged_portion_omitted_
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_RD_PDISK_0,
  Mpi2RaidPhysDiskPage0_t, MPI2_POINTER pMpi2RaidPhysDiskPage0_t;

#define MPI2_RAIDPHYSDISKPAGE0_PAGEVERSION          (0x05)

/* PhysDiskState defines */
#define MPI2_RAID_PD_STATE_NOT_CONFIGURED               (0x00)
#define MPI2_RAID_PD_STATE_NOT_COMPATIBLE               (0x01)
#define MPI2_RAID_PD_STATE_OFFLINE                      (0x02)
#define MPI2_RAID_PD_STATE_ONLINE                       (0x03)
#define MPI2_RAID_PD_STATE_HOT_SPARE                    (0x04)
#define MPI2_RAID_PD_STATE_DEGRADED                     (0x05)
#define MPI2_RAID_PD_STATE_REBUILDING                   (0x06)
#define MPI2_RAID_PD_STATE_OPTIMAL                      (0x07)

/* OfflineReason defines */
#define MPI2_PHYSDISK0_ONLINE                           (0x00)
#define MPI2_PHYSDISK0_OFFLINE_MISSING                  (0x01)
#define MPI2_PHYSDISK0_OFFLINE_FAILED                   (0x03)
#define MPI2_PHYSDISK0_OFFLINE_INITIALIZING             (0x04)
#define MPI2_PHYSDISK0_OFFLINE_REQUESTED                (0x05)
#define MPI2_PHYSDISK0_OFFLINE_FAILED_REQUESTED         (0x06)
#define MPI2_PHYSDISK0_OFFLINE_OTHER                    (0xFF)

/* IncompatibleReason defines */
#define MPI2_PHYSDISK0_COMPATIBLE                       (0x00)
#define MPI2_PHYSDISK0_INCOMPATIBLE_PROTOCOL            (0x01)
#define MPI2_PHYSDISK0_INCOMPATIBLE_BLOCKSIZE           (0x02)
#define MPI2_PHYSDISK0_INCOMPATIBLE_MAX_LBA             (0x03)
#define MPI2_PHYSDISK0_INCOMPATIBLE_SATA_EXTENDED_CMD   (0x04)
#define MPI2_PHYSDISK0_INCOMPATIBLE_REMOVEABLE_MEDIA    (0x05)
#define MPI2_PHYSDISK0_INCOMPATIBLE_MEDIA_TYPE          (0x06)
#define MPI2_PHYSDISK0_INCOMPATIBLE_UNKNOWN             (0xFF)

/* PhysDiskAttributes defines */
#define MPI2_PHYSDISK0_ATTRIB_MEDIA_MASK                (0x0C)
#define MPI2_PHYSDISK0_ATTRIB_SOLID_STATE_DRIVE         (0x08)
#define MPI2_PHYSDISK0_ATTRIB_HARD_DISK_DRIVE           (0x04)

#define MPI2_PHYSDISK0_ATTRIB_PROTOCOL_MASK             (0x03)
```



```
#define MPI2_PHYSDISK0_ATTRIB_SAS_PROTOCOL              (0x02)
#define MPI2_PHYSDISK0_ATTRIB_SATA_PROTOCOL             (0x01)

/* PhysDiskStatusFlags defines */
#define MPI2_PHYSDISK0_STATUS_FLAG_NOT_CERTIFIED        (0x00000040)
#define MPI2_PHYSDISK0_STATUS_FLAG_OCE_TARGET           (0x00000020)
#define MPI2_PHYSDISK0_STATUS_FLAG_WRITE_CACHE_ENABLED  (0x00000010)
#define MPI2_PHYSDISK0_STATUS_FLAG_OPTIMAL_PREVIOUS     (0x00000000)
#define MPI2_PHYSDISK0_STATUS_FLAG_NOT_OPTIMAL_PREVIOUS (0x00000008)
#define MPI2_PHYSDISK0_STATUS_FLAG_INACTIVE_VOLUME      (0x00000004)
#define MPI2_PHYSDISK0_STATUS_FLAG_QUIESCED             (0x00000002)
#define MPI2_PHYSDISK0_STATUS_FLAG_OUT_OF_SYNC          (0x00000001)


/* RAID Physical Disk Page 1 */

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhysDiskPaths at runtime.
 * one and check Header.PageLength or NumPhysDiskPaths at runtime.
 */
#ifndef MPI2_RAID_PHYS_DISK1_PATH_MAX
#define MPI2_RAID_PHYS_DISK1_PATH_MAX   (1)
#endif

typedef struct _MPI2_RAIDPHYSDISK1_PATH
{
    U16             DevHandle;          /* 0x00 */
    U16             Reserved1;          /* 0x02 */
    U64             WWID;               /* 0x04 */
    U64             OwnerWWID;          /* 0x0C */
    U8              OwnerIdentifier;    /* 0x14 */
    U8              Reserved2;          /* 0x15 */
    U16             Flags;              /* 0x16 */
} MPI2_RAIDPHYSDISK1_PATH, MPI2_POINTER PTR_MPI2_RAIDPHYSDISK1_PATH,
_____unchanged_portion_omitted_
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_RD_PDISK_1,
  Mpi2RaidPhysDiskPage1_t, MPI2_POINTER pMpi2RaidPhysDiskPage1_t;

#define MPI2_RAIDPHYSDISKPAGE1_PAGEVERSION          (0x02)


/*****************************************************************************
*   values for fields used by several types of SAS Config Pages
*****************************************************************************/

/* values for NegotiatedLinkRates fields */
#define MPI2_SAS_NEG_LINK_RATE_MASK_LOGICAL             (0xF0)
#define MPI2_SAS_NEG_LINK_RATE_SHIFT_LOGICAL            (4)
#define MPI2_SAS_NEG_LINK_RATE_MASK_PHYSICAL            (0x0F)
/* link rates used for Negotiated Physical and Logical Link Rate */
#define MPI2_SAS_NEG_LINK_RATE_UNKNOWN_LINK_RATE        (0x00)
#define MPI2_SAS_NEG_LINK_RATE_PHY_DISABLED             (0x01)
#define MPI2_SAS_NEG_LINK_RATE_NEGOTIATION_FAILED       (0x02)
#define MPI2_SAS_NEG_LINK_RATE_SATA_OOB_COMPLETE        (0x03)
#define MPI2_SAS_NEG_LINK_RATE_PORT_SELECTOR            (0x04)
#define MPI2_SAS_NEG_LINK_RATE_SMP_RESET_IN_PROGRESS    (0x05)
#define MPI2_SAS_NEG_LINK_RATE_UNSUPPORTED_PHY          (0x06)
#define MPI2_SAS_NEG_LINK_RATE_1_5                      (0x08)
#define MPI2_SAS_NEG_LINK_RATE_3_0                      (0x09)
#define MPI2_SAS_NEG_LINK_RATE_6_0                      (0x0A)
#define MPI25_SAS_NEG_LINK_RATE_12_0                    (0x0B)


/* values for AttachedPhyInfo fields */
#define MPI2_SAS_APHYINFO_INSIDE_ZPSDS_PERSISTENT       (0x00000040)
```

```
#define MPI2_SAS_APHYINFO_REQUESTED_INSIDE_ZPSDS        (0x00000020)
#define MPI2_SAS_APHYINFO_BREAK_REPLY_CAPABLE           (0x00000010)

#define MPI2_SAS_APHYINFO_REASON_MASK                   (0x0000000F)
#define MPI2_SAS_APHYINFO_REASON_UNKNOWN                (0x00000000)
#define MPI2_SAS_APHYINFO_REASON_POWER_ON               (0x00000001)
#define MPI2_SAS_APHYINFO_REASON_HARD_RESET             (0x00000002)
#define MPI2_SAS_APHYINFO_REASON_SMP_PHY_CONTROL        (0x00000003)
#define MPI2_SAS_APHYINFO_REASON_LOSS_OF_SYNC           (0x00000004)
#define MPI2_SAS_APHYINFO_REASON_MULTIPLEXING_SEQ       (0x00000005)
#define MPI2_SAS_APHYINFO_REASON_IT_NEXUS_LOSS_TIMER    (0x00000006)
#define MPI2_SAS_APHYINFO_REASON_BREAK_TIMEOUT          (0x00000007)
#define MPI2_SAS_APHYINFO_REASON_PHY_TEST_STOPPED       (0x00000008)


/* values for PhyInfo fields */
#define MPI2_SAS_PHYINFO_PHY_VACANT                     (0x80000000)

#define MPI2_SAS_PHYINFO_PHY_POWER_CONDITION_MASK       (0x18000000)
#define MPI2_SAS_PHYINFO_SHIFT_PHY_POWER_CONDITION      (27)
#define MPI2_SAS_PHYINFO_PHY_POWER_ACTIVE               (0x00000000)
#define MPI2_SAS_PHYINFO_PHY_POWER_PARTIAL              (0x08000000)
#define MPI2_SAS_PHYINFO_PHY_POWER_SLUMBER              (0x10000000)

#define MPI2_SAS_PHYINFO_CHANGED_REQ_INSIDE_ZPSDS       (0x04000000)
#define MPI2_SAS_PHYINFO_INSIDE_ZPSDS_PERSISTENT        (0x02000000)
#define MPI2_SAS_PHYINFO_REQ_INSIDE_ZPSDS               (0x01000000)
#define MPI2_SAS_PHYINFO_ZONE_GROUP_PERSISTENT          (0x00400000)
#define MPI2_SAS_PHYINFO_INSIDE_ZPSDS                   (0x00200000)
#define MPI2_SAS_PHYINFO_ZONING_ENABLED                 (0x00100000)

#define MPI2_SAS_PHYINFO_REASON_MASK                    (0x000F0000)
#define MPI2_SAS_PHYINFO_REASON_UNKNOWN                 (0x00000000)
#define MPI2_SAS_PHYINFO_REASON_POWER_ON                (0x00010000)
#define MPI2_SAS_PHYINFO_REASON_HARD_RESET              (0x00020000)
#define MPI2_SAS_PHYINFO_REASON_SMP_PHY_CONTROL         (0x00030000)
#define MPI2_SAS_PHYINFO_REASON_LOSS_OF_SYNC            (0x00040000)
#define MPI2_SAS_PHYINFO_REASON_MULTIPLEXING_SEQ        (0x00050000)
#define MPI2_SAS_PHYINFO_REASON_IT_NEXUS_LOSS_TIMER     (0x00060000)
#define MPI2_SAS_PHYINFO_REASON_BREAK_TIMEOUT           (0x00070000)
#define MPI2_SAS_PHYINFO_REASON_PHY_TEST_STOPPED        (0x00080000)

#define MPI2_SAS_PHYINFO_MULTIPLEXING_SUPPORTED         (0x00008000)
#define MPI2_SAS_PHYINFO_SATA_PORT_ACTIVE               (0x00004000)
#define MPI2_SAS_PHYINFO_SATA_PORT_SELECTOR_PRESENT     (0x00002000)
#define MPI2_SAS_PHYINFO_VIRTUAL_PHY                    (0x00001000)

#define MPI2_SAS_PHYINFO_MASK_PARTIAL_PATHWAY_TIME      (0x00000F00)
#define MPI2_SAS_PHYINFO_SHIFT_PARTIAL_PATHWAY_TIME     (8)

#define MPI2_SAS_PHYINFO_MASK_ROUTING_ATTRIBUTE         (0x000000F0)
#define MPI2_SAS_PHYINFO_DIRECT_ROUTING                 (0x00000000)
#define MPI2_SAS_PHYINFO_SUBTRACTIVE_ROUTING            (0x00000010)
#define MPI2_SAS_PHYINFO_TABLE_ROUTING                  (0x00000020)


/* values for SAS ProgrammedLinkRate fields */
#define MPI2_SAS_PRATE_MAX_RATE_MASK                    (0xF0)
#define MPI2_SAS_PRATE_MAX_RATE_NOT_PROGRAMMABLE        (0x00)
#define MPI2_SAS_PRATE_MAX_RATE_1_5                     (0x80)
#define MPI2_SAS_PRATE_MAX_RATE_3_0                     (0x90)
#define MPI2_SAS_PRATE_MAX_RATE_6_0                     (0xA0)
#define MPI25_SAS_PRATE_MAX_RATE_12_0                   (0xB0)
#define MPI2_SAS_PRATE_MIN_RATE_MASK                    (0x0F)
#define MPI2_SAS_PRATE_MIN_RATE_NOT_PROGRAMMABLE        (0x00)
#define MPI2_SAS_PRATE_MIN_RATE_1_5                     (0x08)
```



```
#define MPI2_SAS_PRATE_MIN_RATE_3_0                     (0x09)
#define MPI2_SAS_PRATE_MIN_RATE_6_0                     (0x0A)
#define MPI25_SAS_PRATE_MIN_RATE_12_0                   (0x0B)


/* values for SAS HwLinkRate fields */
#define MPI2_SAS_HWRATE_MAX_RATE_MASK                   (0xF0)
#define MPI2_SAS_HWRATE_MAX_RATE_1_5                    (0x80)
#define MPI2_SAS_HWRATE_MAX_RATE_3_0                    (0x90)
#define MPI2_SAS_HWRATE_MAX_RATE_6_0                    (0xA0)
#define MPI25_SAS_HWRATE_MAX_RATE_12_0                  (0xB0)
#define MPI2_SAS_HWRATE_MIN_RATE_MASK                   (0x0F)
#define MPI2_SAS_HWRATE_MIN_RATE_1_5                    (0x08)
#define MPI2_SAS_HWRATE_MIN_RATE_3_0                    (0x09)
#define MPI2_SAS_HWRATE_MIN_RATE_6_0                    (0x0A)
#define MPI25_SAS_HWRATE_MIN_RATE_12_0                  (0x0B)



/****************************************************************************
*   SAS IO Unit Config Pages
****************************************************************************/

/* SAS IO Unit Page 0 */

typedef struct _MPI2_SAS_IO_UNIT0_PHY_DATA
{
    U8          Port;                   /* 0x00 */
    U8          PortFlags;              /* 0x01 */
    U8          PhyFlags;               /* 0x02 */
    U8          NegotiatedLinkRate;     /* 0x03 */
    U32         ControllerPhyDeviceInfo;/* 0x04 */
    U16         AttachedDevHandle;      /* 0x08 */
    U16         ControllerDevHandle;    /* 0x0A */
    U32         DiscoveryStatus;        /* 0x0C */
    U32         Reserved;               /* 0x10 */
} MPI2_SAS_IO_UNIT0_PHY_DATA, MPI2_POINTER PTR_MPI2_SAS_IO_UNIT0_PHY_DATA,
  Mpi2SasIOUnit0PhyData_t, MPI2_POINTER pMpi2SasIOUnit0PhyData_t;

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhys at runtime.
 * one and check Header.ExtPageLength or NumPhys at runtime.
 */
#ifndef MPI2_SAS_IOUNIT0_PHY_MAX
#define MPI2_SAS_IOUNIT0_PHY_MAX        (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_0
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                             /* 0
    U32                                 Reserved1;                          /* 0
    U8                                  NumPhys;                            /* 0
    U8                                  Reserved2;                          /* 0
    U16                                 Reserved3;                          /* 0
    MPI2_SAS_IO_UNIT0_PHY_DATA          PhyData[MPI2_SAS_IOUNIT0_PHY_MAX];  /* 0
} MPI2_CONFIG_PAGE_SASIOUNIT_0,
_____unchanged_portion_omitted_
  Mpi2SasIOUnit1PhyData_t, MPI2_POINTER pMpi2SasIOUnit1PhyData_t;

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhys at runtime.
 * one and check Header.ExtPageLength or NumPhys at runtime.
 */
#ifndef MPI2_SAS_IOUNIT1_PHY_MAX
```

```
2036 #define MPI2_SAS_IOUNIT1_PHY_MAX        (1)
2037 #endif

2039 typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_1
2040 {
2041     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                         /* 0
2042     U16                                 ControlFlags;                   /* 0
2043     U16                                 SASNarrowMaxQueueDepth;         /* 0
2044     U16                                 AdditionalControlFlags;         /* 0
2045     U16                                 SASWideMaxQueueDepth;           /* 0
2046     U8                                  NumPhys;                        /* 0
2047     U8                                  SATAMaxQDepth;                  /* 0
2048     U8                                  ReportDeviceMissingDelay;       /* 0
2049     U8                                  IODeviceMissingDelay;           /* 0
2050     MPI2_SAS_IO_UNIT1_PHY_DATA          PhyData[MPI2_SAS_IOUNIT1_PHY_MAX];  /* 0
2051 } MPI2_CONFIG_PAGE_SASIOUNIT_1,
2052   MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_1,
2053   Mpi2SasIOUnitPage1_t, MPI2_POINTER pMpi2SasIOUnitPage1_t;

2055 #define MPI2_SASIOUNITPAGE1_PAGEVERSION     (0x09)

2057 /* values for SAS IO Unit Page 1 ControlFlags */
2058 #define MPI2_SASIOUNIT1_CONTROL_DEVICE_SELF_TEST                (0x8000)
2059 #define MPI2_SASIOUNIT1_CONTROL_SATA_3_0_MAX                    (0x4000)
2060 #define MPI2_SASIOUNIT1_CONTROL_SATA_1_5_MAX                    (0x2000) /*
1778 #define MPI2_SASIOUNIT1_CONTROL_SATA_1_5_MAX                    (0x2000)
2061 #define MPI2_SASIOUNIT1_CONTROL_SATA_SW_PRESERVE                (0x1000)

2063 #define MPI2_SASIOUNIT1_CONTROL_MASK_DEV_SUPPORT                (0x0600)
2064 #define MPI2_SASIOUNIT1_CONTROL_SHIFT_DEV_SUPPORT               (9)
2065 #define MPI2_SASIOUNIT1_CONTROL_DEV_SUPPORT_BOTH                (0x0)
2066 #define MPI2_SASIOUNIT1_CONTROL_DEV_SAS_SUPPORT                 (0x1)
2067 #define MPI2_SASIOUNIT1_CONTROL_DEV_SATA_SUPPORT                (0x2)

2069 #define MPI2_SASIOUNIT1_CONTROL_SATA_48BIT_LBA_REQUIRED         (0x0080)
2070 #define MPI2_SASIOUNIT1_CONTROL_SATA_SMART_REQUIRED             (0x0040)
2071 #define MPI2_SASIOUNIT1_CONTROL_SATA_NCQ_REQUIRED               (0x0020)
2072 #define MPI2_SASIOUNIT1_CONTROL_SATA_FUA_REQUIRED               (0x0010)
2073 #define MPI2_SASIOUNIT1_CONTROL_TABLE_SUBTRACTIVE_ILLEGAL       (0x0008)
2074 #define MPI2_SASIOUNIT1_CONTROL_SUBTRACTIVE_ILLEGAL             (0x0004)
2075 #define MPI2_SASIOUNIT1_CONTROL_FIRST_LVL_DISC_ONLY             (0x0002)
2076 #define MPI2_SASIOUNIT1_CONTROL_CLEAR_AFFILIATION               (0x0001) /*
1794 #define MPI2_SASIOUNIT1_CONTROL_CLEAR_AFFILIATION               (0x0001)

2078 /* values for SAS IO Unit Page 1 AdditionalControlFlags */
2079 #define MPI2_SASIOUNIT1_ACONTROL_MULTI_PORT_DOMAIN_ILLEGAL      (0x0080)
2080 #define MPI2_SASIOUNIT1_ACONTROL_SATA_ASYNCHROUNOUS_NOTIFICATION    (0x0040)
2081 #define MPI2_SASIOUNIT1_ACONTROL_INVALID_TOPOLOGY_CORRECTION    (0x0020)
2082 #define MPI2_SASIOUNIT1_ACONTROL_PORT_ENABLE_ONLY_SATA_LINK_RESET   (0x0010)
2083 #define MPI2_SASIOUNIT1_ACONTROL_OTHER_AFFILIATION_SATA_LINK_RESET  (0x0008)
2084 #define MPI2_SASIOUNIT1_ACONTROL_SELF_AFFILIATION_SATA_LINK_RESET   (0x0004)
2085 #define MPI2_SASIOUNIT1_ACONTROL_NO_AFFILIATION_SATA_LINK_RESET     (0x0002)
2086 #define MPI2_SASIOUNIT1_ACONTROL_ALLOW_TABLE_TO_TABLE           (0x0001)

2088 /* defines for SAS IO Unit Page 1 ReportDeviceMissingDelay */
2089 #define MPI2_SASIOUNIT1_REPORT_MISSING_TIMEOUT_MASK             (0x7F)
2090 #define MPI2_SASIOUNIT1_REPORT_MISSING_UNIT_16                  (0x80)

2092 /* values for SAS IO Unit Page 1 PortFlags */
2093 #define MPI2_SASIOUNIT1_PORT_FLAGS_AUTO_PORT_CONFIG             (0x01)

2095 /* values for SAS IO Unit Page 1 PhyFlags */
2096 #define MPI2_SASIOUNIT1_PHYFLAGS_ZONING_ENABLE                  (0x10)
2097 #define MPI2_SASIOUNIT1_PHYFLAGS_PHY_DISABLE                    (0x08)

2099 /* values for SAS IO Unit Page 1 MaxMinLinkRate */
```



```
2100 #define MPI2_SASIOUNIT1_MAX_RATE_MASK                               (0xF0)
2101 #define MPI2_SASIOUNIT1_MAX_RATE_1_5                                (0x80)
2102 #define MPI2_SASIOUNIT1_MAX_RATE_3_0                                (0x90)
2103 #define MPI2_SASIOUNIT1_MAX_RATE_6_0                                (0xA0)
2104 #define MPI25_SASIOUNIT1_MAX_RATE_12_0                              (0xB0)
2105 #define MPI2_SASIOUNIT1_MIN_RATE_MASK                               (0x0F)
2106 #define MPI2_SASIOUNIT1_MIN_RATE_1_5                                (0x08)
2107 #define MPI2_SASIOUNIT1_MIN_RATE_3_0                                (0x09)
2108 #define MPI2_SASIOUNIT1_MIN_RATE_6_0                                (0x0A)
2109 #define MPI25_SASIOUNIT1_MIN_RATE_12_0                              (0x0B)

2111 /* see mpi2_sas.h for values for SAS IO Unit Page 1 ControllerPhyDeviceInfo valu


2114 /* SAS IO Unit Page 4 */

2116 typedef struct _MPI2_SAS_IOUNIT4_SPINUP_GROUP
2117 {
2118     U8          MaxTargetSpinup;            /* 0x00 */
2119     U8          SpinupDelay;                /* 0x01 */
2120     U8          SpinupFlags;                /* 0x02 */
2121     U8          Reserved1;                  /* 0x03 */
1836     U16         Reserved1;                  /* 0x02 */
2122 } MPI2_SAS_IOUNIT4_SPINUP_GROUP, MPI2_POINTER PTR_MPI2_SAS_IOUNIT4_SPINUP_GROUP,
2123   Mpi2SasIOUnit4SpinupGroup_t, MPI2_POINTER pMpi2SasIOUnit4SpinupGroup_t;

2125 /* defines for SAS IO Unit Page 4 SpinupFlags */
2126 #define MPI2_SASIOUNIT4_SPINUP_DISABLE_FLAG         (0x01)



2129 /*
2130  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
2131  * one and check the value returned for NumPhys at runtime.
1842  * four and check Header.ExtPageLength or NumPhys at runtime.
2132  */
2133 #ifndef MPI2_SAS_IOUNIT4_PHY_MAX
2134 #define MPI2_SAS_IOUNIT4_PHY_MAX        (4)
2135 #endif

2137 typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_4
2138 {
2139     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                         /* 0x00
2140     MPI2_SAS_IOUNIT4_SPINUP_GROUP       SpinupGroupParameters[4];       /* 0x08
2141     U32                                 Reserved1;                      /* 0x18
2142     U32                                 Reserved2;                      /* 0x1C
2143     U32                                 Reserved3;                      /* 0x20
2144     U8                                  BootDeviceWaitTime;             /* 0x24
2145     U8                                  Reserved4;                      /* 0x25
2146     U16                                 Reserved5;                      /* 0x26
2147     U8                                  NumPhys;                        /* 0x28
2148     U8                                  PEInitialSpinupDelay;           /* 0x29
2149     U8                                  PEReplyDelay;                   /* 0x2A
2150     U8                                  Flags;                          /* 0x2B
2151     U8                                  PHY[MPI2_SAS_IOUNIT4_PHY_MAX];  /* 0x2C
2152 } MPI2_CONFIG_PAGE_SASIOUNIT_4,
2153   MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_4,
2154   Mpi2SasIOUnitPage4_t, MPI2_POINTER pMpi2SasIOUnitPage4_t;

2156 #define MPI2_SASIOUNITPAGE4_PAGEVERSION     (0x02)

2158 /* defines for Flags field */
2159 #define MPI2_SASIOUNIT4_FLAGS_AUTO_PORTENABLE               (0x01)

2161 /* defines for PHY field */
2162 #define MPI2_SASIOUNIT4_PHY_SPINUP_GROUP_MASK               (0x03)
```

```
/* SAS IO Unit Page 5 */

typedef struct _MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS
{
    U8          ControlFlags;               /* 0x00 */
    U8          PortWidthModGroup;          /* 0x01 */
    U8          Reserved1;                  /* 0x01 */
    U16         InactivityTimerExponent;    /* 0x02 */
    U8          SATAPartialTimeout;         /* 0x04 */
    U8          Reserved2;                  /* 0x05 */
    U8          SATASlumberTimeout;         /* 0x06 */
    U8          Reserved3;                  /* 0x07 */
    U8          SASPartialTimeout;          /* 0x08 */
    U8          Reserved4;                  /* 0x09 */
    U8          SASSlumberTimeout;          /* 0x0A */
    U8          Reserved5;                  /* 0x0B */
} MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS,
  MPI2_POINTER PTR_MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS,
  Mpi2SasIOUnit5PhyPmSettings_t, MPI2_POINTER pMpi2SasIOUnit5PhyPmSettings_t;

/* defines for ControlFlags field */
#define MPI2_SASIOUNIT5_CONTROL_SAS_SLUMBER_ENABLE      (0x08)
#define MPI2_SASIOUNIT5_CONTROL_SAS_PARTIAL_ENABLE      (0x04)
#define MPI2_SASIOUNIT5_CONTROL_SATA_SLUMBER_ENABLE     (0x02)
#define MPI2_SASIOUNIT5_CONTROL_SATA_PARTIAL_ENABLE     (0x01)

/* defines for PortWidthModeGroup field */
#define MPI2_SASIOUNIT5_PWMG_DISABLE                    (0xFF)

/* defines for InactivityTimerExponent field */
#define MPI2_SASIOUNIT5_ITE_MASK_SAS_SLUMBER            (0x7000)
#define MPI2_SASIOUNIT5_ITE_SHIFT_SAS_SLUMBER           (12)
#define MPI2_SASIOUNIT5_ITE_MASK_SAS_PARTIAL            (0x0700)
#define MPI2_SASIOUNIT5_ITE_SHIFT_SAS_PARTIAL           (8)
#define MPI2_SASIOUNIT5_ITE_MASK_SATA_SLUMBER           (0x0070)
#define MPI2_SASIOUNIT5_ITE_SHIFT_SATA_SLUMBER          (4)
#define MPI2_SASIOUNIT5_ITE_MASK_SATA_PARTIAL           (0x0007)
#define MPI2_SASIOUNIT5_ITE_SHIFT_SATA_PARTIAL          (0)

#define MPI2_SASIOUNIT5_ITE_TEN_SECONDS                 (7)
#define MPI2_SASIOUNIT5_ITE_ONE_SECOND                  (6)
#define MPI2_SASIOUNIT5_ITE_HUNDRED_MILLISECONDS        (5)
#define MPI2_SASIOUNIT5_ITE_TEN_MILLISECONDS            (4)
#define MPI2_SASIOUNIT5_ITE_ONE_MILLISECOND             (3)
#define MPI2_SASIOUNIT5_ITE_HUNDRED_MICROSECONDS        (2)
#define MPI2_SASIOUNIT5_ITE_TEN_MICROSECONDS            (1)
#define MPI2_SASIOUNIT5_ITE_ONE_MICROSECOND             (0)

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhys at runtime.
 * one and check Header.ExtPageLength or NumPhys at runtime.
 */
#ifndef MPI2_SAS_IOUNIT5_PHY_MAX
#define MPI2_SAS_IOUNIT5_PHY_MAX        (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_5
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                             /* 0
    U8                                  NumPhys;                            /* 0
    U8                                  Reserved1;                          /* 0
    U16                                 Reserved2;                          /* 0
    U32                                 Reserved3;                          /* 0
    MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS   SASPhyPowerManagementSettings[MPI2_SAS_I
```



```
} MPI2_CONFIG_PAGE_SASIOUNIT_5,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_5,
  Mpi2SasIOUnitPage5_t, MPI2_POINTER pMpi2SasIOUnitPage5_t;

#define MPI2_SASIOUNITPAGE5_PAGEVERSION     (0x01)
#define MPI2_SASIOUNITPAGE5_PAGEVERSION     (0x00)


/* SAS IO Unit Page 6 */

typedef struct _MPI2_SAS_IO_UNIT6_PORT_WIDTH_MOD_GROUP_STATUS
{
    U8          CurrentStatus;              /* 0x00 */
    U8          CurrentModulation;          /* 0x01 */
    U8          CurrentUtilization;         /* 0x02 */
    U8          Reserved1;                  /* 0x03 */
    U32         Reserved2;                  /* 0x04 */
} MPI2_SAS_IO_UNIT6_PORT_WIDTH_MOD_GROUP_STATUS,
  MPI2_POINTER PTR_MPI2_SAS_IO_UNIT6_PORT_WIDTH_MOD_GROUP_STATUS,
  Mpi2SasIOUnit6PortWidthModGroupStatus_t,
  MPI2_POINTER pMpi2SasIOUnit6PortWidthModGroupStatus_t;

/* defines for CurrentStatus field */
#define MPI2_SASIOUNIT6_STATUS_UNAVAILABLE                      (0x00)
#define MPI2_SASIOUNIT6_STATUS_UNCONFIGURED                     (0x01)
#define MPI2_SASIOUNIT6_STATUS_INVALID_CONFIG                   (0x02)
#define MPI2_SASIOUNIT6_STATUS_LINK_DOWN                        (0x03)
#define MPI2_SASIOUNIT6_STATUS_OBSERVATION_ONLY                 (0x04)
#define MPI2_SASIOUNIT6_STATUS_INACTIVE                         (0x05)
#define MPI2_SASIOUNIT6_STATUS_ACTIVE_IOUNIT                    (0x06)
#define MPI2_SASIOUNIT6_STATUS_ACTIVE_HOST                      (0x07)

/* defines for CurrentModulation field */
#define MPI2_SASIOUNIT6_MODULATION_25_PERCENT                   (0x00)
#define MPI2_SASIOUNIT6_MODULATION_50_PERCENT                   (0x01)
#define MPI2_SASIOUNIT6_MODULATION_75_PERCENT                   (0x02)
#define MPI2_SASIOUNIT6_MODULATION_100_PERCENT                  (0x03)

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumGroups at runtime.
 */
#ifndef MPI2_SAS_IOUNIT6_GROUP_MAX
#define MPI2_SAS_IOUNIT6_GROUP_MAX      (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_6
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                         /* 0x00 */
    U32                                 Reserved1;                      /* 0x08 */
    U32                                 Reserved2;                      /* 0x0C */
    U8                                  NumGroups;                      /* 0x10 */
    U8                                  Reserved3;                      /* 0x11 */
    U16                                 Reserved4;                      /* 0x12 */
    MPI2_SAS_IO_UNIT6_PORT_WIDTH_MOD_GROUP_STATUS
        PortWidthModulationGroupStatus[MPI2_SAS_IOUNIT6_GROUP_MAX]; /* 0x14 */
} MPI2_CONFIG_PAGE_SASIOUNIT_6,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_6,
  Mpi2SasIOUnitPage6_t, MPI2_POINTER pMpi2SasIOUnitPage6_t;

#define MPI2_SASIOUNITPAGE6_PAGEVERSION     (0x00)


/* SAS IO Unit Page 7 */

typedef struct _MPI2_SAS_IO_UNIT7_PORT_WIDTH_MOD_GROUP_SETTINGS
```

```
{
    U8          Flags;                      /* 0x00 */
    U8          Reserved1;                  /* 0x01 */
    U16         Reserved2;                  /* 0x02 */
    U8          Threshold75Pct;             /* 0x04 */
    U8          Threshold50Pct;             /* 0x05 */
    U8          Threshold25Pct;             /* 0x06 */
    U8          Reserved3;                  /* 0x07 */
} MPI2_SAS_IO_UNIT7_PORT_WIDTH_MOD_GROUP_SETTINGS,
  MPI2_POINTER PTR_MPI2_SAS_IO_UNIT7_PORT_WIDTH_MOD_GROUP_SETTINGS,
  Mpi2SasIOUnit7PortWidthModGroupSettings_t,
  MPI2_POINTER pMpi2SasIOUnit7PortWidthModGroupSettings_t;

/* defines for Flags field */
#define MPI2_SASIOUNIT7_FLAGS_ENABLE_PORT_WIDTH_MODULATION  (0x01)


/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumGroups at runtime.
 */
#ifndef MPI2_SAS_IOUNIT7_GROUP_MAX
#define MPI2_SAS_IOUNIT7_GROUP_MAX      (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_7
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER            Header;             /* 0x00 */
    U8                                          SamplingInterval;   /* 0x08 */
    U8                                          WindowLength;       /* 0x09 */
    U16                                         Reserved1;          /* 0x0A */
    U32                                         Reserved2;          /* 0x0C */
    U32                                         Reserved3;          /* 0x10 */
    U8                                          NumGroups;          /* 0x14 */
    U8                                          Reserved4;          /* 0x15 */
    U16                                         Reserved5;          /* 0x16 */
    MPI2_SAS_IO_UNIT7_PORT_WIDTH_MOD_GROUP_SETTINGS
        PortWidthModulationGroupSettings[MPI2_SAS_IOUNIT7_GROUP_MAX]; /* 0x18 */
} MPI2_CONFIG_PAGE_SASIOUNIT_7,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_7,
  Mpi2SasIOUnitPage7_t, MPI2_POINTER pMpi2SasIOUnitPage7_t;

#define MPI2_SASIOUNITPAGE7_PAGEVERSION     (0x00)


/* SAS IO Unit Page 8 */

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT_8
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                         /* 0x00
    U32                                 Reserved1;                      /* 0x08
    U32                                 PowerManagementCapabilities;    /* 0x0C
    U8                                  TxRxSleepStatus;                /* 0x10
    U8                                  Reserved2;                      /* 0x11
    U16                                 Reserved3;                      /* 0x12
} MPI2_CONFIG_PAGE_SASIOUNIT_8,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT_8,
  Mpi2SasIOUnitPage8_t, MPI2_POINTER pMpi2SasIOUnitPage8_t;

#define MPI2_SASIOUNITPAGE8_PAGEVERSION     (0x00)

/* defines for PowerManagementCapabilities field */
#define MPI2_SASIOUNIT8_PM_HOST_PORT_WIDTH_MOD          (0x00001000)
#define MPI2_SASIOUNIT8_PM_HOST_SAS_SLUMBER_MODE        (0x00000800)
#define MPI2_SASIOUNIT8_PM_HOST_SAS_PARTIAL_MODE        (0x00000400)
#define MPI2_SASIOUNIT8_PM_HOST_SATA_SLUMBER_MODE       (0x00000200)
```



```
#define MPI2_SASIOUNIT8_PM_HOST_SATA_PARTIAL_MODE       (0x00000100)
#define MPI2_SASIOUNIT8_PM_IOUNIT_PORT_WIDTH_MOD        (0x00000010)
#define MPI2_SASIOUNIT8_PM_IOUNIT_SAS_SLUMBER_MODE      (0x00000008)
#define MPI2_SASIOUNIT8_PM_IOUNIT_SAS_PARTIAL_MODE      (0x00000004)
#define MPI2_SASIOUNIT8_PM_IOUNIT_SATA_SLUMBER_MODE     (0x00000002)
#define MPI2_SASIOUNIT8_PM_IOUNIT_SATA_PARTIAL_MODE     (0x00000001)

/* defines for TxRxSleepStatus field */
#define MPI25_SASIOUNIT8_TXRXSLEEP_UNSUPPORTED          (0x00)
#define MPI25_SASIOUNIT8_TXRXSLEEP_DISENGAGED           (0x01)
#define MPI25_SASIOUNIT8_TXRXSLEEP_ACTIVE               (0x02)
#define MPI25_SASIOUNIT8_TXRXSLEEP_SHUTDOWN             (0x03)



/* SAS IO Unit Page 16 */

typedef struct _MPI2_CONFIG_PAGE_SASIOUNIT16
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                         /* 0
    U64                                 TimeStamp;                      /* 0
    U32                                 Reserved1;                      /* 0
    U32                                 Reserved2;                      /* 0
    U32                                 FastPathPendedRequests;         /* 0
    U32                                 FastPathUnPendedRequests;       /* 0
    U32                                 FastPathHostRequestStarts;      /* 0
    U32                                 FastPathFirmwareRequestStarts;  /* 0
    U32                                 FastPathHostCompletions;        /* 0
    U32                                 FastPathFirmwareCompletions;    /* 0
    U32                                 NonFastPathRequestStarts;       /* 0
    U32                                 NonFastPathHostCompletions;     /* 0
} MPI2_CONFIG_PAGE_SASIOUNIT16,
  MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SASIOUNIT16,
  Mpi2SasIOUnitPage16_t, MPI2_POINTER pMpi2SasIOUnitPage16_t;

#define MPI2_SASIOUNITPAGE16_PAGEVERSION    (0x00)


/*****************************************************************************
*   SAS Expander Config Pages
*****************************************************************************/

/* SAS Expander Page 0 */

typedef struct _MPI2_CONFIG_PAGE_EXPANDER_0
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
    U8                                  PhysicalPort;               /* 0x08 */
    U8                                  ReportGenLength;            /* 0x09 */
    U16                                 EnclosureHandle;            /* 0x0A */
    U64                                 SASAddress;                 /* 0x0C */
    U32                                 DiscoveryStatus;            /* 0x14 */
    U16                                 DevHandle;                  /* 0x18 */
    U16                                 ParentDevHandle;            /* 0x1A */
    U16                                 ExpanderChangeCount;        /* 0x1C */
    U16                                 ExpanderRouteIndexes;       /* 0x1E */
    U8                                  NumPhys;                    /* 0x20 */
    U8                                  SASLevel;                   /* 0x21 */
    U16                                 Flags;                      /* 0x22 */
    U16                                 STPBusInactivityTimeLimit;  /* 0x24 */
    U16                                 STPMaxConnectTimeLimit;     /* 0x26 */
    U16                                 STP_SMP_NexusLossTime;      /* 0x28 */
    U16                                 MaxNumRoutedSasAddresses;   /* 0x2A */
    U64                                 ActiveZoneManagerSASAddress;/* 0x2C */
    U16                                 ZoneLockInactivityLimit;    /* 0x34 */
    U16                                 Reserved1;                  /* 0x36 */
```

```
2425     U8                                  TimeToReducedFunc;          /* 0x38 */
2426     U8                                  InitialTimeToReducedFunc;   /* 0x39 */
2427     U8                                  MaxReducedFuncTime;         /* 0x3A */
2428     U8                                  Reserved2;                  /* 0x3B */
2429 } MPI2_CONFIG_PAGE_EXPANDER_0, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_EXPANDER_0,
_____unchanged_portion_omitted_
2498   Mpi2ExpanderPage1_t, MPI2_POINTER pMpi2ExpanderPage1_t;

2500 #define MPI2_SASEXPANDER1_PAGEVERSION       (0x02)

2502 /* use MPI2_SAS_PRATE_ defines for the ProgrammedLinkRate field */

2504 /* use MPI2_SAS_HWRATE_ defines for the HwLinkRate field */

2506 /* use MPI2_SAS_PHYINFO_ for the PhyInfo field */

2508 /* see mpi2_sas.h for the MPI2_SAS_DEVICE_INFO_ defines used for the AttachedDev

2510 /* use MPI2_SAS_NEG_LINK_RATE_ defines for the NegotiatedLinkRate field */

2060 /* use MPI2_SAS_APHYINFO_ defines for AttachedPhyInfo field */

2512 /* values for SAS Expander Page 1 DiscoveryInfo field */
2513 #define MPI2_SAS_EXPANDER1_DISCINFO_BAD_PHY_DISABLED    (0x04)
2514 #define MPI2_SAS_EXPANDER1_DISCINFO_LINK_STATUS_CHANGE  (0x02)
2515 #define MPI2_SAS_EXPANDER1_DISCINFO_NO_ROUTING_ENTRIES  (0x01)

2517 /* use MPI2_SAS_APHYINFO_ defines for AttachedPhyInfo field */


2520 /*****************************************************************************
2521 *   SAS Device Config Pages
2522 *****************************************************************************/

2524 /* SAS Device Page 0 */

2526 typedef struct _MPI2_CONFIG_PAGE_SAS_DEV_0
2527 {
2528     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                 /* 0x00 */
2529     U16                                 Slot;                   /* 0x08 */
2530     U16                                 EnclosureHandle;        /* 0x0A */
2531     U64                                 SASAddress;             /* 0x0C */
2532     U16                                 ParentDevHandle;        /* 0x14 */
2533     U8                                  PhyNum;                 /* 0x16 */
2534     U8                                  AccessStatus;           /* 0x17 */
2535     U16                                 DevHandle;              /* 0x18 */
2536     U8                                  AttachedPhyIdentifier;  /* 0x1A */
2537     U8                                  ZoneGroup;              /* 0x1B */
2538     U32                                 DeviceInfo;             /* 0x1C */
2539     U16                                 Flags;                  /* 0x20 */
2540     U8                                  PhysicalPort;           /* 0x22 */
2541     U8                                  MaxPortConnections;     /* 0x23 */
2542     U64                                 DeviceName;             /* 0x24 */
2543     U8                                  PortGroups;             /* 0x2C */
2544     U8                                  DmaGroup;               /* 0x2D */
2545     U8                                  ControlGroup;           /* 0x2E */
2546     U8                                  EnclosureLevel;         /* 0x2F */
2547     U8                                  ConnectorName[4];       /* 0x30 */
2094     U8                                  Reserved1;              /* 0x2F */
2095     U32                                 Reserved2;              /* 0x30 */
2548     U32                                 Reserved3;              /* 0x34 */
2549 } MPI2_CONFIG_PAGE_SAS_DEV_0, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_DEV_0,
2550   Mpi2SasDevicePage0_t, MPI2_POINTER pMpi2SasDevicePage0_t;

2552 #define MPI2_SASDEVICE0_PAGEVERSION         (0x09)
2100 #define MPI2_SASDEVICE0_PAGEVERSION         (0x08)
```



```
2554 /* values for SAS Device Page 0 AccessStatus field */
2555 #define MPI2_SAS_DEVICE0_ASTATUS_NO_ERRORS                  (0x00)
2556 #define MPI2_SAS_DEVICE0_ASTATUS_SATA_INIT_FAILED           (0x01)
2557 #define MPI2_SAS_DEVICE0_ASTATUS_SATA_CAPABILITY_FAILED     (0x02)
2558 #define MPI2_SAS_DEVICE0_ASTATUS_SATA_AFFILIATION_CONFLICT  (0x03)
2559 #define MPI2_SAS_DEVICE0_ASTATUS_SATA_NEEDS_INITIALIZATION  (0x04)
2560 #define MPI2_SAS_DEVICE0_ASTATUS_ROUTE_NOT_ADDRESSABLE      (0x05)
2561 #define MPI2_SAS_DEVICE0_ASTATUS_SMP_ERROR_NOT_ADDRESSABLE  (0x06)
2562 #define MPI2_SAS_DEVICE0_ASTATUS_DEVICE_BLOCKED             (0x07)
2563 /* specific values for SATA Init failures */
2564 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_UNKNOWN                (0x10)
2565 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_AFFILIATION_CONFLICT   (0x11)
2566 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_DIAG                   (0x12)
2567 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_IDENTIFICATION         (0x13)
2568 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_CHECK_POWER            (0x14)
2569 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_PIO_SN                 (0x15)
2570 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_MDMA_SN                (0x16)
2571 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_UDMA_SN                (0x17)
2572 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_ZONING_VIOLATION       (0x18)
2573 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_NOT_ADDRESSABLE        (0x19)
2574 #define MPI2_SAS_DEVICE0_ASTATUS_SIF_MAX                    (0x1F)

2576 /* see mpi2_sas.h for values for SAS Device Page 0 DeviceInfo values */

2578 /* values for SAS Device Page 0 Flags field */
2579 #define MPI2_SAS_DEVICE0_FLAGS_UNAUTHORIZED_DEVICE          (0x8000)
2580 #define MPI25_SAS_DEVICE0_FLAGS_ENABLED_FAST_PATH           (0x4000)
2581 #define MPI25_SAS_DEVICE0_FLAGS_FAST_PATH_CAPABLE           (0x2000)
2582 #define MPI2_SAS_DEVICE0_FLAGS_SLUMBER_PM_CAPABLE           (0x1000)
2583 #define MPI2_SAS_DEVICE0_FLAGS_PARTIAL_PM_CAPABLE           (0x0800)
2584 #define MPI2_SAS_DEVICE0_FLAGS_SATA_ASYNCHRONOUS_NOTIFY     (0x0400)
2585 #define MPI2_SAS_DEVICE0_FLAGS_SATA_SW_PRESERVE             (0x0200)
2586 #define MPI2_SAS_DEVICE0_FLAGS_UNSUPPORTED_DEVICE           (0x0100)
2587 #define MPI2_SAS_DEVICE0_FLAGS_SATA_48BIT_LBA_SUPPORTED     (0x0080)
2588 #define MPI2_SAS_DEVICE0_FLAGS_SATA_SMART_SUPPORTED         (0x0040)
2589 #define MPI2_SAS_DEVICE0_FLAGS_SATA_NCQ_SUPPORTED           (0x0020)
2590 #define MPI2_SAS_DEVICE0_FLAGS_SATA_FUA_SUPPORTED           (0x0010)
2591 #define MPI2_SAS_DEVICE0_FLAGS_PORT_SELECTOR_ATTACH         (0x0008)
2592 #define MPI2_SAS_DEVICE0_FLAGS_ENCL_LEVEL_VALID             (0x0002)
2593 #define MPI2_SAS_DEVICE0_FLAGS_DEVICE_PRESENT               (0x0001)


2596 /* SAS Device Page 1 */

2598 typedef struct _MPI2_CONFIG_PAGE_SAS_DEV_1
2599 {
2600     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                 /* 0x00 */
2601     U32                                 Reserved1;              /* 0x08 */
2602     U64                                 SASAddress;             /* 0x0C */
2603     U32                                 Reserved2;              /* 0x14 */
2604     U16                                 DevHandle;              /* 0x18 */
2605     U16                                 Reserved3;              /* 0x1A */
2606     U8                                  InitialRegDeviceFIS[20];/* 0x1C */
2607 } MPI2_CONFIG_PAGE_SAS_DEV_1, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_DEV_1,
_____unchanged_portion_omitted_
2637   Mpi2SasPhyPage0_t, MPI2_POINTER pMpi2SasPhyPage0_t;

2639 #define MPI2_SASPHY0_PAGEVERSION            (0x03)

2641 /* use MPI2_SAS_APHYINFO_ defines for AttachedPhyInfo field */

2643 /* use MPI2_SAS_PRATE_ defines for the ProgrammedLinkRate field */

2645 /* use MPI2_SAS_HWRATE_ defines for the HwLinkRate field */
```

```
/* values for SAS PHY Page 0 Flags field */
#define MPI2_SAS_PHY0_FLAGS_SGPIO_DIRECT_ATTACH_ENC             (0x01)

/* use MPI2_SAS_PHYINFO_ for the PhyInfo field */
/* use MPI2_SAS_APHYINFO_ defines for AttachedPhyInfo field */

/* use MPI2_SAS_NEG_LINK_RATE_ defines for the NegotiatedLinkRate field */

/* use MPI2_SAS_PHYINFO_ for the PhyInfo field */


/* SAS PHY Page 1 */

typedef struct _MPI2_CONFIG_PAGE_SAS_PHY_1
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
    U32                                 Reserved1;                  /* 0x08 */
    U32                                 InvalidDwordCount;          /* 0x0C */
    U32                                 RunningDisparityErrorCount; /* 0x10 */
    U32                                 LossDwordSynchCount;        /* 0x14 */
    U32                                 PhyResetProblemCount;       /* 0x18 */
} MPI2_CONFIG_PAGE_SAS_PHY_1, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_PHY_1,
  unchanged_portion_omitted_
  Mpi2SasPhy2PhyEvent_t, MPI2_POINTER pMpi2SasPhy2PhyEvent_t;

/* use MPI2_SASPHY3_EVENT_CODE_ for the PhyEventCode field */


/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhyEvents at runtime.
 * one and check Header.ExtPageLength or NumPhyEvents at runtime.
 */
#ifndef MPI2_SASPHY2_PHY_EVENT_MAX
#define MPI2_SASPHY2_PHY_EVENT_MAX      (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SAS_PHY_2
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
    U32                                 Reserved1;                  /* 0x08 */
    U8                                  NumPhyEvents;               /* 0x0C */
    U8                                  Reserved2;                  /* 0x0D */
    U16                                 Reserved3;                  /* 0x0E */
    MPI2_SASPHY2_PHY_EVENT              PhyEvent[MPI2_SASPHY2_PHY_EVENT_MAX]; /*
} MPI2_CONFIG_PAGE_SAS_PHY_2, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_PHY_2,
  unchanged_portion_omitted_
  Mpi2SasPhy3PhyEventConfig_t, MPI2_POINTER pMpi2SasPhy3PhyEventConfig_t;

/* values for PhyEventCode field */
#define MPI2_SASPHY3_EVENT_CODE_NO_EVENT                    (0x00)
#define MPI2_SASPHY3_EVENT_CODE_INVALID_DWORD               (0x01)
#define MPI2_SASPHY3_EVENT_CODE_RUNNING_DISPARITY_ERROR     (0x02)
#define MPI2_SASPHY3_EVENT_CODE_LOSS_DWORD_SYNC             (0x03)
#define MPI2_SASPHY3_EVENT_CODE_PHY_RESET_PROBLEM           (0x04)
#define MPI2_SASPHY3_EVENT_CODE_ELASTICITY_BUF_OVERFLOW     (0x05)
#define MPI2_SASPHY3_EVENT_CODE_RX_ERROR                    (0x06)
#define MPI2_SASPHY3_EVENT_CODE_RX_ADDR_FRAME_ERROR         (0x20)
#define MPI2_SASPHY3_EVENT_CODE_TX_AC_OPEN_REJECT           (0x21)
#define MPI2_SASPHY3_EVENT_CODE_RX_AC_OPEN_REJECT           (0x22)
#define MPI2_SASPHY3_EVENT_CODE_TX_RC_OPEN_REJECT           (0x23)
#define MPI2_SASPHY3_EVENT_CODE_RX_RC_OPEN_REJECT           (0x24)
#define MPI2_SASPHY3_EVENT_CODE_RX_AIP_PARTIAL_WAITING_ON   (0x25)
#define MPI2_SASPHY3_EVENT_CODE_RX_AIP_CONNECT_WAITING_ON   (0x26)
#define MPI2_SASPHY3_EVENT_CODE_TX_BREAK                    (0x27)
#define MPI2_SASPHY3_EVENT_CODE_RX_BREAK                    (0x28)
```



```
#define MPI2_SASPHY3_EVENT_CODE_BREAK_TIMEOUT               (0x29)
#define MPI2_SASPHY3_EVENT_CODE_CONNECTION                  (0x2A)
#define MPI2_SASPHY3_EVENT_CODE_PEAKTX_PATHWAY_BLOCKED      (0x2B)
#define MPI2_SASPHY3_EVENT_CODE_PEAKTX_ARB_WAIT_TIME        (0x2C)
#define MPI2_SASPHY3_EVENT_CODE_PEAK_ARB_WAIT_TIME          (0x2D)
#define MPI2_SASPHY3_EVENT_CODE_PEAK_CONNECT_TIME           (0x2E)
#define MPI2_SASPHY3_EVENT_CODE_TX_SSP_FRAMES               (0x40)
#define MPI2_SASPHY3_EVENT_CODE_RX_SSP_FRAMES               (0x41)
#define MPI2_SASPHY3_EVENT_CODE_TX_SSP_ERROR_FRAMES         (0x42)
#define MPI2_SASPHY3_EVENT_CODE_RX_SSP_ERROR_FRAMES         (0x43)
#define MPI2_SASPHY3_EVENT_CODE_TX_CREDIT_BLOCKED           (0x44)
#define MPI2_SASPHY3_EVENT_CODE_RX_CREDIT_BLOCKED           (0x45)
#define MPI2_SASPHY3_EVENT_CODE_TX_SATA_FRAMES              (0x50)
#define MPI2_SASPHY3_EVENT_CODE_RX_SATA_FRAMES              (0x51)
#define MPI2_SASPHY3_EVENT_CODE_SATA_OVERFLOW               (0x52)
#define MPI2_SASPHY3_EVENT_CODE_TX_SMP_FRAMES               (0x60)
#define MPI2_SASPHY3_EVENT_CODE_RX_SMP_FRAMES               (0x61)
#define MPI2_SASPHY3_EVENT_CODE_RX_SMP_ERROR_FRAMES         (0x63)
#define MPI2_SASPHY3_EVENT_CODE_HOTPLUG_TIMEOUT             (0xD0)
#define MPI2_SASPHY3_EVENT_CODE_MISALIGNED_MUX_PRIMITIVE    (0xD1)
#define MPI2_SASPHY3_EVENT_CODE_RX_AIP                      (0xD2)

/* values for the CounterType field */
#define MPI2_SASPHY3_COUNTER_TYPE_WRAPPING                  (0x00)
#define MPI2_SASPHY3_COUNTER_TYPE_SATURATING                (0x01)
#define MPI2_SASPHY3_COUNTER_TYPE_PEAK_VALUE                (0x02)

/* values for the TimeUnits field */
#define MPI2_SASPHY3_TIME_UNITS_10_MICROSECONDS             (0x00)
#define MPI2_SASPHY3_TIME_UNITS_100_MICROSECONDS            (0x01)
#define MPI2_SASPHY3_TIME_UNITS_1_MILLISECOND               (0x02)
#define MPI2_SASPHY3_TIME_UNITS_10_MILLISECONDS             (0x03)

/* values for the ThresholdFlags field */
#define MPI2_SASPHY3_TFLAGS_PHY_RESET                       (0x0002)
#define MPI2_SASPHY3_TFLAGS_EVENT_NOTIFY                    (0x0001)

/*
 * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
 * one and check the value returned for NumPhyEvents at runtime.
 * one and check Header.ExtPageLength or NumPhyEvents at runtime.
 */
#ifndef MPI2_SASPHY3_PHY_EVENT_MAX
#define MPI2_SASPHY3_PHY_EVENT_MAX      (1)
#endif

typedef struct _MPI2_CONFIG_PAGE_SAS_PHY_3
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
    U32                                 Reserved1;                  /* 0x08 */
    U8                                  NumPhyEvents;               /* 0x0C */
    U8                                  Reserved2;                  /* 0x0D */
    U16                                 Reserved3;                  /* 0x0E */
    MPI2_SASPHY3_PHY_EVENT_CONFIG       PhyEventConfig[MPI2_SASPHY3_PHY_EVENT_MA
} MPI2_CONFIG_PAGE_SAS_PHY_3, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_PHY_3,
  Mpi2SasPhyPage3_t, MPI2_POINTER pMpi2SasPhyPage3_t;

#define MPI2_SASPHY3_PAGEVERSION            (0x00)


/* SAS PHY Page 4 */

typedef struct _MPI2_CONFIG_PAGE_SAS_PHY_4
{
    MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
    U16                                 Reserved1;                  /* 0x08 */
```

```
2806     U8                                  Reserved2;                  /* 0x0A */
2807     U8                                  Flags;                      /* 0x0B */
2808     U8                                  InitialFrame[28];           /* 0x0C */
2809 } MPI2_CONFIG_PAGE_SAS_PHY_4, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_PHY_4,
2810   Mpi2SasPhyPage4_t, MPI2_POINTER pMpi2SasPhyPage4_t;

2812 #define MPI2_SASPHY4_PAGEVERSION            (0x00)

2814 /* values for the Flags field */
2815 #define MPI2_SASPHY4_FLAGS_FRAME_VALID        (0x02)
2816 #define MPI2_SASPHY4_FLAGS_SATA_FRAME         (0x01)




2821 /****************************************************************************
2822 *   SAS Port Config Pages
2823 ****************************************************************************/

2825 /* SAS Port Page 0 */

2827 typedef struct _MPI2_CONFIG_PAGE_SAS_PORT_0
2828 {
2829     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
2830     U8                                  PortNumber;                 /* 0x08 */
2831     U8                                  PhysicalPort;               /* 0x09 */
2832     U8                                  PortWidth;                  /* 0x0A */
2833     U8                                  PhysicalPortWidth;          /* 0x0B */
2834     U8                                  ZoneGroup;                  /* 0x0C */
2835     U8                                  Reserved1;                  /* 0x0D */
2836     U16                                 Reserved2;                  /* 0x0E */
2837     U64                                 SASAddress;                 /* 0x10 */
2838     U32                                 DeviceInfo;                 /* 0x18 */
2839     U32                                 Reserved3;                  /* 0x1C */
2840     U32                                 Reserved4;                  /* 0x20 */
2841 } MPI2_CONFIG_PAGE_SAS_PORT_0, MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_PORT_0,
2842   Mpi2SasPortPage0_t, MPI2_POINTER pMpi2SasPortPage0_t;

2844 #define MPI2_SASPORT0_PAGEVERSION           (0x00)

2846 /* see mpi2_sas.h for values for SAS Port Page 0 DeviceInfo values */


2849 /****************************************************************************
2850 *   SAS Enclosure Config Pages
2851 ****************************************************************************/

2853 /* SAS Enclosure Page 0 */

2855 typedef struct _MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0
2856 {
2857     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                     /* 0x00 */
2858     U32                                 Reserved1;                  /* 0x08 */
2859     U64                                 EnclosureLogicalID;         /* 0x0C */
2860     U16                                 Flags;                      /* 0x14 */
2861     U16                                 EnclosureHandle;            /* 0x16 */
2862     U16                                 NumSlots;                   /* 0x18 */
2863     U16                                 StartSlot;                  /* 0x1A */
2864     U8                                  Reserved2;                  /* 0x1C */
2865     U8                                  EnclosureLevel;             /* 0x1D */
2387     U16                                 Reserved2;                  /* 0x1C */
2866     U16                                 SEPDevHandle;               /* 0x1E */
2867     U32                                 Reserved3;                  /* 0x20 */
2868     U32                                 Reserved4;                  /* 0x24 */
2869 } MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0,
2870   MPI2_POINTER PTR_MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0,
```



```
2871   Mpi2SasEnclosurePage0_t, MPI2_POINTER pMpi2SasEnclosurePage0_t;

2873 #define MPI2_SASENCLOSURE0_PAGEVERSION      (0x04)
2395 #define MPI2_SASENCLOSURE0_PAGEVERSION      (0x03)

2875 /* values for SAS Enclosure Page 0 Flags field */
2876 #define MPI2_SAS_ENCLS0_FLAGS_ENCL_LEVEL_VALID      (0x0010)
2877 #define MPI2_SAS_ENCLS0_FLAGS_MNG_MASK              (0x000F)
2878 #define MPI2_SAS_ENCLS0_FLAGS_MNG_UNKNOWN           (0x0000)
2879 #define MPI2_SAS_ENCLS0_FLAGS_MNG_IOC_SES           (0x0001)
2880 #define MPI2_SAS_ENCLS0_FLAGS_MNG_IOC_SGPIO         (0x0002)
2881 #define MPI2_SAS_ENCLS0_FLAGS_MNG_EXP_SGPIO         (0x0003)
2882 #define MPI2_SAS_ENCLS0_FLAGS_MNG_SES_ENCLOSURE     (0x0004)
2883 #define MPI2_SAS_ENCLS0_FLAGS_MNG_IOC_GPIO          (0x0005)


2886 /****************************************************************************
2887 *   Log Config Page
2888 ****************************************************************************/

2890 /* Log Page 0 */

2892 /*
2893  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
2894  * one and check the value returned for NumLogEntries at runtime.
2415  * one and check Header.ExtPageLength or NumPhys at runtime.
2895  */
2896 #ifndef MPI2_LOG_0_NUM_LOG_ENTRIES
2897 #define MPI2_LOG_0_NUM_LOG_ENTRIES          (1)
2898 #endif

2900 #define MPI2_LOG_0_LOG_DATA_LENGTH          (0x1C)

2902 typedef struct _MPI2_LOG_0_ENTRY
2903 {
2904     U64         TimeStamp;                          /* 0x00 */
2905     U32         Reserved1;                          /* 0x08 */
2906     U16         LogSequence;                        /* 0x0C */
2907     U16         LogEntryQualifier;                  /* 0x0E */
2908     U8          VP_ID;                              /* 0x10 */
2909     U8          VF_ID;                              /* 0x11 */
2910     U16         Reserved2;                          /* 0x12 */
2911     U8          LogData[MPI2_LOG_0_LOG_DATA_LENGTH];/* 0x14 */
2912 } MPI2_LOG_0_ENTRY, MPI2_POINTER PTR_MPI2_LOG_0_ENTRY,
_____unchanged_portion_omitted_
2931   Mpi2LogPage0_t, MPI2_POINTER pMpi2LogPage0_t;

2933 #define MPI2_LOG_0_PAGEVERSION              (0x02)


2936 /****************************************************************************
2937 *   RAID Config Page
2938 ****************************************************************************/

2940 /* RAID Page 0 */

2942 /*
2943  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
2944  * one and check the value returned for NumElements at runtime.
2465  * one and check Header.ExtPageLength or NumPhys at runtime.
2945  */
2946 #ifndef MPI2_RAIDCONFIG0_MAX_ELEMENTS
2947 #define MPI2_RAIDCONFIG0_MAX_ELEMENTS       (1)
2948 #endif

2950 typedef struct _MPI2_RAIDCONFIG0_CONFIG_ELEMENT
```

```
2951 {
2952     U16                     ElementFlags;               /* 0x00 */
2953     U16                     VolDevHandle;               /* 0x02 */
2954     U8                      HotSparePool;               /* 0x04 */
2955     U8                      PhysDiskNum;                /* 0x05 */
2956     U16                     PhysDiskDevHandle;          /* 0x06 */
2957 } MPI2_RAIDCONFIG0_CONFIG_ELEMENT,
_____unchanged_portion_omitted_
3117   Mpi2EthernetPage1_t, MPI2_POINTER pMpi2EthernetPage1_t;

3119 #define MPI2_ETHERNETPAGE1_PAGEVERSION   (0x00)

3121 /* values for Ethernet Page 1 Flags field */
3122 #define MPI2_ETHPG1_FLAG_SET_DEFAULT_IF             (0x00000100)
3123 #define MPI2_ETHPG1_FLAG_ENABLE_FW_DOWNLOAD         (0x00000080)
3124 #define MPI2_ETHPG1_FLAG_ENABLE_TELNET              (0x00000040)
3125 #define MPI2_ETHPG1_FLAG_ENABLE_SSH2                (0x00000020)
3126 #define MPI2_ETHPG1_FLAG_ENABLE_DHCP_CLIENT         (0x00000010)
3127 #define MPI2_ETHPG1_FLAG_ENABLE_IPV6                (0x00000008)
3128 #define MPI2_ETHPG1_FLAG_ENABLE_IPV4                (0x00000004)
3129 #define MPI2_ETHPG1_FLAG_USE_IPV6_ADDRESSES         (0x00000002)
3130 #define MPI2_ETHPG1_FLAG_ENABLE_ETH_IF              (0x00000001)

3132 /* values for Ethernet Page 1 MediaState field */
3133 #define MPI2_ETHPG1_MS_DUPLEX_MASK                  (0x80)
3134 #define MPI2_ETHPG1_MS_HALF_DUPLEX                  (0x00)
3135 #define MPI2_ETHPG1_MS_FULL_DUPLEX                  (0x80)

3137 #define MPI2_ETHPG1_MS_DATA_RATE_MASK               (0x07)
3138 #define MPI2_ETHPG1_MS_DATA_RATE_AUTO               (0x00)
3139 #define MPI2_ETHPG1_MS_DATA_RATE_10MBIT             (0x01)
3140 #define MPI2_ETHPG1_MS_DATA_RATE_100MBIT            (0x02)
3141 #define MPI2_ETHPG1_MS_DATA_RATE_1GBIT              (0x03)


3144 /*****************************************************************************
3145 *   Extended Manufacturing Config Pages
3146 *****************************************************************************/

3148 /*
3149  * Generic structure to use for product-specific extended manufacturing pages
3150  * (currently Extended Manufacturing Page 40 through Extended Manufacturing
3151  * Page 60).
3152  */

3154 typedef struct _MPI2_CONFIG_PAGE_EXT_MAN_PS
3155 {
3156     MPI2_CONFIG_EXTENDED_PAGE_HEADER    Header;                 /* 0x00 */
3157     U32                                 ProductSpecificInfo;    /* 0x08 */
3158 } MPI2_CONFIG_PAGE_EXT_MAN_PS,
3159   MPI2_POINTER PTR_MPI2_CONFIG_PAGE_EXT_MAN_PS,
3160   Mpi2ExtManufacturingPagePS_t, MPI2_POINTER pMpi2ExtManufacturingPagePS_t;

3162 /* PageVersion should be provided by product-specific code */

3164 #endif
```

```
**********************************************************
    6995 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_hbd.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

/*
 *  Copyright (c) 2009-2011 LSI Corporation.
 *
 *
 *           Name:  mpi2_hbd.h
 *          Title:  MPI Host Based Discovery messages and structures
 *  Creation Date:  October 21, 2009
 *
 *  mpi2_hbd.h Version:  02.00.02
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  10-28-09  02.00.00  Initial version.
 *  08-11-10  02.00.01  Removed PortGroups, DmaGroup, and ControlGroup from
 *                      HBD Action request, replaced by AdditionalInfo field.
 *  11-18-11  02.00.02  Incorporating additions for MPI v2.5.
 *  ------------------------------------------------------------------------
 */

#ifndef MPI2_HBD_H
#define MPI2_HBD_H

/*****************************************************************************
*  Host Based Discovery Action messages
```



```
******************************************************************************/

/* Host Based Discovery Action Request Message */
typedef struct _MPI2_HBD_ACTION_REQUEST
{
    U8                      Operation;          /* 0x00 */
    U8                      Reserved1;          /* 0x01 */
    U8                      ChainOffset;        /* 0x02 */
    U8                      Function;           /* 0x03 */
    U16                     DevHandle;          /* 0x04 */
    U8                      Reserved2;          /* 0x06 */
    U8                      MsgFlags;           /* 0x07 */
    U8                      VP_ID;              /* 0x08 */
    U8                      VF_ID;              /* 0x09 */
    U16                     Reserved3;          /* 0x0A */
    U32                     Reserved4;          /* 0x0C */
    U64                     SASAddress;         /* 0x10 */
    U32                     Reserved5;          /* 0x18 */
    U32                     HbdDeviceInfo;      /* 0x1C */
    U16                     ParentDevHandle;    /* 0x20 */
    U16                     MaxQDepth;          /* 0x22 */
    U8                      FirstPhyIdentifier; /* 0x24 */
    U8                      Port;               /* 0x25 */
    U8                      MaxConnections;     /* 0x26 */
    U8                      MaxRate;            /* 0x27 */
    U32                     AdditionalInfo;     /* 0x28 */
    U16                     InitialAWT;         /* 0x2C */
    U16                     Reserved7;          /* 0x2E */
    U32                     Reserved8;          /* 0x30 */
} MPI2_HBD_ACTION_REQUEST, MPI2_POINTER PTR_MPI2_HBD_ACTION_REQUEST,
  Mpi2HbdActionRequest_t, MPI2_POINTER pMpi2HbdActionRequest_t;

/* values for the Operation field */
#define MPI2_HBD_OP_ADD_DEVICE                  (0x01)
#define MPI2_HBD_OP_REMOVE_DEVICE               (0x02)
#define MPI2_HBD_OP_UPDATE_DEVICE               (0x03)

/* values for the HbdDeviceInfo field */
#define MPI2_HBD_DEVICE_INFO_VIRTUAL_DEVICE     (0x00004000)
#define MPI2_HBD_DEVICE_INFO_ATAPI_DEVICE       (0x00002000)
#define MPI2_HBD_DEVICE_INFO_DIRECT_ATTACH      (0x00000800)
#define MPI2_HBD_DEVICE_INFO_SSP_TARGET         (0x00000400)
#define MPI2_HBD_DEVICE_INFO_STP_TARGET         (0x00000200)
#define MPI2_HBD_DEVICE_INFO_SMP_TARGET         (0x00000100)
#define MPI2_HBD_DEVICE_INFO_SATA_DEVICE        (0x00000080)
#define MPI2_HBD_DEVICE_INFO_SSP_INITIATOR      (0x00000040)
#define MPI2_HBD_DEVICE_INFO_STP_INITIATOR      (0x00000020)
#define MPI2_HBD_DEVICE_INFO_SMP_INITIATOR      (0x00000010)
#define MPI2_HBD_DEVICE_INFO_SATA_HOST          (0x00000008)

#define MPI2_HBD_DEVICE_INFO_MASK_DEVICE_TYPE   (0x00000007)
#define MPI2_HBD_DEVICE_INFO_NO_DEVICE          (0x00000000)
#define MPI2_HBD_DEVICE_INFO_END_DEVICE         (0x00000001)
#define MPI2_HBD_DEVICE_INFO_EDGE_EXPANDER      (0x00000002)
#define MPI2_HBD_DEVICE_INFO_FANOUT_EXPANDER    (0x00000003)

/* values for the MaxRate field */
#define MPI2_HBD_MAX_RATE_MASK                  (0x0F)
#define MPI2_HBD_MAX_RATE_1_5                   (0x08)
#define MPI2_HBD_MAX_RATE_3_0                   (0x09)
#define MPI2_HBD_MAX_RATE_6_0                   (0x0A)
#define MPI25_HBD_MAX_RATE_12_0                 (0x0B)


/* Host Based Discovery Action Reply Message */
typedef struct _MPI2_HBD_ACTION_REPLY
```

```
128 {
129     U8                      Operation;          /* 0x00 */
130     U8                      Reserved1;          /* 0x01 */
131     U8                      MsgLength;          /* 0x02 */
132     U8                      Function;           /* 0x03 */
133     U16                     DevHandle;          /* 0x04 */
134     U8                      Reserved2;          /* 0x06 */
135     U8                      MsgFlags;           /* 0x07 */
136     U8                      VP_ID;              /* 0x08 */
137     U8                      VF_ID;              /* 0x09 */
138     U16                     Reserved3;          /* 0x0A */
139     U16                     Reserved4;          /* 0x0C */
140     U16                     IOCStatus;          /* 0x0E */
141     U32                     IOCLogInfo;         /* 0x10 */
142 } MPI2_HBD_ACTION_REPLY, MPI2_POINTER PTR_MPI2_HBD_ACTION_REPLY,
143   Mpi2HbdActionReply_t, MPI2_POINTER pMpi2HbdActionReply_t;


146 #endif
```

```
**********************************************************
   37705 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_history.txt
NEX-1888 upstream
**********************************************************
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

 ==============================
 Fusion-MPT MPI 2.0 / 2.5 Header File Change History
 ==============================

 Copyright (c) 2000-2013 LSI Corporation.

 ---------------------------------------
 Header Set Release Version:    02.00.33
 Header Set Release Date:       12-05-13
 ---------------------------------------

 Filename               Current version     Prior version
 ----------             ---------------     -------------
 mpi2.h                 02.00.33            02.00.32
 mpi2_cnfg.h            02.00.27            02.00.26
 mpi2_init.h            02.00.15            02.00.15
 mpi2_ioc.h             02.00.24            02.00.23
 mpi2_raid.h            02.00.10            02.00.10
 mpi2_sas.h             02.00.08            02.00.08
 mpi2_targ.h            02.00.06            02.00.06
 mpi2_tool.h            02.00.11            02.00.11
 mpi2_type.h            02.00.00            02.00.00
 mpi2_ra.h              02.00.00            02.00.00
 mpi2_hbd.h             02.00.02            02.00.02
 mpi2_history.txt       02.00.33            02.00.32


 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------

mpi2.h
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
```



```
 *  06-04-07  02.00.01  Bumped MPI2_HEADER_VERSION_UNIT.
 *  06-26-07  02.00.02  Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-31-07  02.00.03  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Moved ReplyPostHostIndex register to offset 0x6C of the
 *                      MPI2_SYSTEM_INTERFACE_REGS and modified the define for
 *                      MPI2_REPLY_POST_HOST_INDEX_OFFSET.
 *                      Added union of request descriptors.
 *                      Added union of reply descriptors.
 *  10-31-07  02.00.04  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added define for MPI2_VERSION_02_00.
 *                      Fixed the size of the FunctionDependent5 field in the
 *                      MPI2_DEFAULT_REPLY structure.
 *  12-18-07  02.00.05  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Removed the MPI-defined Fault Codes and extended the
 *                      product specific codes up to 0xEFFF.
 *                      Added a sixth key value for the WriteSequence register
 *                      and changed the flush value to 0x0.
 *                      Added message function codes for Diagnostic Buffer Post
 *                      and Diagnsotic Release.
 *                      New IOCStatus define: MPI2_IOCSTATUS_DIAGNOSTIC_RELEASED
 *                      Moved MPI2_VERSION_UNION from mpi2_ioc.h.
 *  02-29-08  02.00.06  Bumped MPI2_HEADER_VERSION_UNIT.
 *  03-03-08  02.00.07  Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-21-08  02.00.08  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added #defines for marking a reply descriptor as unused.
 *  06-27-08  02.00.09  Bumped MPI2_HEADER_VERSION_UNIT.
 *  10-02-08  02.00.10  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Moved LUN field defines from mpi2_init.h.
 *  01-19-09  02.00.11  Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-06-09  02.00.12  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      In all request and reply descriptors, replaced VF_ID
 *                      field with MSIxIndex field.
 *                      Removed DevHandle field from
 *                      MPI2_SCSI_IO_SUCCESS_REPLY_DESCRIPTOR and made those
 *                      bytes reserved.
 *                      Added RAID Accelerator functionality.
 *  07-30-09  02.00.13  Bumped MPI2_HEADER_VERSION_UNIT.
 *  10-28-09  02.00.14  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added MSI-x index mask and shift for Reply Post Host
 *                      Index register.
 *                      Added function code for Host Based Discovery Action.
 *  02-10-10  02.00.15  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added define for MPI2_FUNCTION_PWR_MGMT_CONTROL.
 *                      Added defines for product-specific range of message
 *                      function codes, 0xF0 to 0xFF.
 *  05-12-10  02.00.16  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added alternative defines for the SGE Direction bit.
 *  08-11-10  02.00.17  Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-10-10  02.00.18  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added MPI2_IEEE_SGE_FLAGS_SYSTEMPLBCPI_ADDR define.
 *  02-23-11  02.00.19  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added MPI2_FUNCTION_SEND_HOST_MESSAGE.
 *  03-09-11  02.00.20  Bumped MPI2_HEADER_VERSION_UNIT.
 *  05-25-11  02.00.21  Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-24-11  02.00.22  Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-18-11  02.00.23  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Incorporating additions for MPI v2.5.
 *  02-06-12  02.00.24  Bumped MPI2_HEADER_VERSION_UNIT.
 *  03-29-12  02.00.25  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added Hard Reset delay timings.
 *  07-10-12  02.00.26  Bumped MPI2_HEADER_VERSION_UNIT.
 *  07-26-12  02.00.27  Bumped MPI2_HEADER_VERSION_UNIT.
 *  11-27-12  02.00.28  Bumped MPI2_HEADER_VERSION_UNIT.
 *  12-20-12  02.00.29  Bumped MPI2_HEADER_VERSION_UNIT.
 *                      Added MPI25_SUP_REPLY_POST_HOST_INDEX_OFFSET.
 *  04-09-13  02.00.30  Bumped MPI2_HEADER_VERSION_UNIT.
```

```
 *  04-17-13  02.00.31  Bumped MPI2_HEADER_VERSION_UNIT.
 *  08-19-13  02.00.32  Bumped MPI2_HEADER_VERSION_UNIT.
 *  12-05-13  02.00.33  Bumped MPI2_HEADER_VERSION_UNIT.
 *  --------------------------------------------------------------------

mpi2_cnfg.h
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-04-07  02.00.01  Added defines for SAS IO Unit Page 2 PhyFlags.
 *                      Added Manufacturing Page 11.
 *                      Added MPI2_SAS_EXPANDER0_FLAGS_CONNECTOR_END_DEVICE
 *                      define.
 *  06-26-07  02.00.02  Adding generic structure for product-specific
 *                      Manufacturing pages: MPI2_CONFIG_PAGE_MANUFACTURING_PS.
 *                      Rework of BIOS Page 2 configuration page.
 *                      Fixed MPI2_BIOSPAGE2_BOOT_DEVICE to be a union of the
 *                      forms.
 *                      Added configuration pages IOC Page 8 and Driver
 *                      Persistent Mapping Page 0.
 *  08-31-07  02.00.03  Modified configuration pages dealing with Integrated
 *                      RAID (Manufacturing Page 4, RAID Volume Pages 0 and 1,
 *                      RAID Physical Disk Pages 0 and 1, RAID Configuration
 *                      Page 0).
 *                      Added new value for AccessStatus field of SAS Device
 *                      Page 0 (_SATA_NEEDS_INITIALIZATION).
 *  10-31-07  02.00.04  Added missing SEPDevHandle field to
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *  12-18-07  02.00.05  Modified IO Unit Page 0 to use 32-bit version fields for
 *                      NVDATA.
 *                      Modified IOC Page 7 to use masks and added field for
 *                      SASBroadcastPrimitiveMasks.
 *                      Added MPI2_CONFIG_PAGE_BIOS_4.
 *                      Added MPI2_CONFIG_PAGE_LOG_0.
 *  02-29-08  02.00.06  Modified various names to make them 32-character unique.
 *                      Added SAS Device IDs.
 *                      Updated Integrated RAID configuration pages including
 *                      Manufacturing Page 4, IOC Page 6, and RAID Configuration
 *                      Page 0.
 *  05-21-08  02.00.07  Added define MPI2_MANPAGE4_MIX_SSD_SAS_SATA.
 *                      Added define MPI2_MANPAGE4_PHYSDISK_128MB_COERCION.
 *                      Fixed define MPI2_IOCPAGE8_FLAGS_ENCLOSURE_SLOT_MAPPING.
 *                      Added missing MaxNumRoutedSasAddresses field to
 *                      MPI2_CONFIG_PAGE_EXPANDER_0.
 *                      Added SAS Port Page 0.
 *                      Modified structure layout for
 *                      MPI2_CONFIG_PAGE_DRIVER_MAPPING_0.
 *  06-27-08  02.00.08  Changed MPI2_CONFIG_PAGE_RD_PDISK_1 to use
 *                      MPI2_RAID_PHYS_DISK1_PATH_MAX to size the array.
 *  10-02-08  02.00.09  Changed MPI2_RAID_PGAD_CONFIGNUM_MASK from 0x0000FFFF
 *                      to 0x000000FF.
 *                      Added two new values for the Physical Disk Coercion Size
 *                      bits in the Flags field of Manufacturing Page 4.
 *                      Added product-specific Manufacturing pages 16 to 31.
 *                      Modified Flags bits for controlling write cache on SATA
 *                      drives in IO Unit Page 1.
 *                      Added new bit to AdditionalControlFlags of SAS IO Unit
 *                      Page 1 to control Invalid Topology Correction.
 *                      Added SupportedPhysDisks field to RAID Volume Page 1 and
 *                      added related defines.
 *                      Added additional defines for RAID Volume Page 0
 *                      VolumeStatusFlags field.
 *                      Modified meaning of RAID Volume Page 0 VolumeSettings
 *                      define for auto-configure of hot-swap drives.
 *                      Added PhysDiskAttributes field (and related defines) to
 *                      RAID Physical Disk Page 0.
 *                      Added MPI2_SAS_PHYINFO_PHY_VACANT define.
 *                      Added three new DiscoveryStatus bits for SAS IO Unit
```



```
 *                      Page 0 and SAS Expander Page 0.
 *                      Removed multiplexing information from SAS IO Unit pages.
 *                      Added BootDeviceWaitTime field to SAS IO Unit Page 4.
 *                      Removed Zone Address Resolved bit from PhyInfo and from
 *                      Expander Page 0 Flags field.
 *                      Added two new AccessStatus values to SAS Device Page 0
 *                      for indicating routing problems. Added 3 reserved words
 *                      to this page.
 *  01-19-09  02.00.10  Fixed defines for GPIOVal field of IO Unit Page 3.
 *                      Inserted missing reserved field into structure for IOC
 *                      Page 6.
 *                      Added more pending task bits to RAID Volume Page 0
 *                      VolumeStatusFlags defines.
 *                      Added MPI2_PHYSDISK0_STATUS_FLAG_NOT_CERTIFIED define.
 *                      Added a new DiscoveryStatus bit for SAS IO Unit Page 0
 *                      and SAS Expander Page 0 to flag a downstream initiator
 *                      when in simplified routing mode.
 *                      Removed SATA Init Failure defines for DiscoveryStatus
 *                      fields of SAS IO Unit Page 0 and SAS Expander Page 0.
 *                      Added MPI2_SAS_DEVICE0_ASTATUS_DEVICE_BLOCKED define.
 *                      Added PortGroups, DmaGroup, and ControlGroup fields to
 *                      SAS Device Page 0.
 *  05-06-09  02.00.11  Added structures and defines for IO Unit Page 5 and IO
 *                      Unit Page 6.
 *                      Added expander reduced functionality data to SAS
 *                      Expander Page 0.
 *                      Added SAS PHY Page 2 and SAS PHY Page 3.
 *  07-30-09  02.00.12  Added IO Unit Page 7.
 *                      Added new device ids.
 *                      Added SAS IO Unit Page 5.
 *                      Added partial and slumber power management capable flags
 *                      to SAS Device Page 0 Flags field.
 *                      Added PhyInfo defines for power condition.
 *                      Added Ethernet configuration pages.
 *  10-28-09  02.00.13  Added MPI2_IOUNITPAGE1_ENABLE_HOST_BASED_DISCOVERY.
 *                      Added SAS PHY Page 4 structure and defines.
 *  02-10-10  02.00.14  Modified the comments for the configuration page
 *                      structures that contain an array of data. The host
 *                      should use the "count" field in the page data (e.g. the
 *                      NumPhys field) to determine the number of valid elements
 *                      in the array.
 *                      Added/modified some MPI2_MFGPAGE_DEVID_SAS defines.
 *                      Added PowerManagementCapabilities to IO Unit Page 7.
 *                      Added PortWidthModGroup field to
 *                      MPI2_SAS_IO_UNIT5_PHY_PM_SETTINGS.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_6 and related defines.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_7 and related defines.
 *                      Added MPI2_CONFIG_PAGE_SASIOUNIT_8 and related defines.
 *  05-12-10  02.00.15  Added MPI2_RAIDVOL0_STATUS_FLAG_VOL_NOT_CONSISTENT
 *                      define.
 *                      Added MPI2_PHYSDISK0_INCOMPATIBLE_MEDIA_TYPE define.
 *                      Added MPI2_SAS_NEG_LINK_RATE_UNSUPPORTED_PHY define.
 *  08-11-10  02.00.16  Removed IO Unit Page 1 device path (multi-pathing)
 *                      defines.
 *  11-10-10  02.00.17  Added ReceptacleID field (replacing Reserved1) to
 *                      MPI2_MANPAGE7_CONNECTOR_INFO and reworked defines for
 *                      the Pinout field.
 *                      Added BoardTemperature and BoardTemperatureUnits fields
 *                      to MPI2_CONFIG_PAGE_IO_UNIT_7.
 *                      Added MPI2_CONFIG_EXTPAGETYPE_EXT_MANUFACTURING define
 *                      and MPI2_CONFIG_PAGE_EXT_MAN_PS structure.
 *  02-23-11  02.00.18  Added ProxyVF_ID field to MPI2_CONFIG_REQUEST.
 *                      Added IO Unit Page 8, IO Unit Page 9,
 *                      and IO Unit Page 10.
 *                      Added SASNotifyPrimitiveMasks field to
 *                      MPI2_CONFIG_PAGE_IOC_7.
```

```
 *  03-09-11  02.00.19  Fixed IO Unit Page 10 (to match the spec).
 *  05-25-11  02.00.20  Cleaned up a few comments.
 *  08-24-11  02.00.21  Marked the IO Unit Page 7 PowerManagementCapabilities
 *                      for PCIe link as obsolete.
 *                      Added SpinupFlags field containing a Disable Spin-up bit
 *                      to the MPI2_SAS_IOUNIT4_SPINUP_GROUP fields of SAS IO
 *                      Unit Page 4.
 *  11-18-11  02.00.22  Added define MPI2_IOCPAGE6_CAP_FLAGS_4K_SECTORS_SUPPORT.
 *                      Added UEFIVersion field to BIOS Page 1 and defined new
 *                      BiosOptions bits.
 *                      Incorporating additions for MPI v2.5.
 *  11-27-12  02.00.23  Added MPI2_MANPAGE7_FLAG_EVENTREPLAY_SLOT_ORDER.
 *                      Added MPI2_BIOSPAGE1_OPTIONS_MASK_OEM_ID.
 *  12-20-12  02.00.24  Marked MPI2_SASIOUNIT1_CONTROL_CLEAR_AFFILIATION as
 *                      obsolete for MPI v2.5 and later.
 *                      Added some defines for 12G SAS speeds.
 *  04-09-13  02.00.25  Added MPI2_IOUNITPAGE1_ATA_SECURITY_FREEZE_LOCK.
 *                      Fixed MPI2_IOUNITPAGE5_DMA_CAP_MASK_MAX_REQUESTS to
 *                      match the specification.
 *  08-19-13  02.00.26  Added reserved words to MPI2_CONFIG_PAGE_IO_UNIT_7 for
 *                      future use.
 *  12-05-13  02.00.27  Added MPI2_MANPAGE7_FLAG_BASE_ENCLOSURE_LEVEL for
 *                      MPI2_CONFIG_PAGE_MAN_7.
 *                      Added EnclosureLevel and ConnectorName fields to
 *                      MPI2_CONFIG_PAGE_SAS_DEV_0.
 *                      Added MPI2_SAS_DEVICE0_FLAGS_ENCL_LEVEL_VALID for
 *                      MPI2_CONFIG_PAGE_SAS_DEV_0.
 *                      Added EnclosureLevel field to
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *                      Added MPI2_SAS_ENCLS0_FLAGS_ENCL_LEVEL_VALID for
 *                      MPI2_CONFIG_PAGE_SAS_ENCLOSURE_0.
 *  ----------------------------------------------------------------------------

mpi2_init.h
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  10-31-07  02.00.01  Fixed name for pMpi2SCSITaskManagementRequest_t.
 *  12-18-07  02.00.02  Modified Task Management Target Reset Method defines.
 *  02-29-08  02.00.03  Added Query Task Set and Query Unit Attention.
 *  03-03-08  02.00.04  Fixed name of struct _MPI2_SCSI_TASK_MANAGE_REPLY.
 *  05-21-08  02.00.05  Fixed typo in name of Mpi2SepRequest_t.
 *  10-02-08  02.00.06  Removed Untagged and No Disconnect values from SCSI IO
 *                      Control field Task Attribute flags.
 *                      Moved LUN field defines to mpi2.h becasue they are
 *                      common to many structures.
 *  05-06-09  02.00.07  Changed task management type of Query Unit Attention to
 *                      Query Asynchronous Event.
 *                      Defined two new bits in the SlotStatus field of the SCSI
 *                      Enclosure Processor Request and Reply.
 *  10-28-09  02.00.08  Added defines for decoding the ResponseInfo bytes for
 *                      both SCSI IO Error Reply and SCSI Task Management Reply.
 *                      Added ResponseInfo field to MPI2_SCSI_TASK_MANAGE_REPLY.
 *                      Added MPI2_SCSITASKMGMT_RSP_TM_OVERLAPPED_TAG define.
 *  02-10-10  02.00.09  Removed unused structure that had "#if 0" around it.
 *  05-12-10  02.00.10  Added optional vendor-unique region to SCSI IO Request.
 *  11-10-10  02.00.11  Added MPI2_SCSIIO_NUM_SGLOFFSETS define.
 *  11-18-11  02.00.12  Incorporating additions for MPI v2.5.
 *  02-06-12  02.00.13  Added alternate defines for Task Priority / Command
 *                      Priority to match SAM-4.
 *                      Added EEDPErrorOffset to MPI2_SCSI_IO_REPLY.
 *  07-10-12  02.00.14  Added MPI2_SCSIIO_CONTROL_SHIFT_DATADIRECTION.
 *  ----------------------------------------------------------------------------

mpi2_ioc.h
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-04-07  02.00.01  In IOCFacts Reply structure, renamed MaxDevices to
 *                      MaxTargets.
```



```
 *                      Added TotalImageSize field to FWDownload Request.
 *                      Added reserved words to FWUpload Request.
 *  06-26-07  02.00.02  Added IR Configuration Change List Event.
 *  08-31-07  02.00.03  Removed SystemReplyQueueDepth field from the IOCInit
 *                      request and replaced it with
 *                      ReplyDescriptorPostQueueDepth and ReplyFreeQueueDepth.
 *                      Replaced the MinReplyQueueDepth field of the IOCFacts
 *                      reply with MaxReplyDescriptorPostQueueDepth.
 *                      Added MPI2_RDPQ_DEPTH_MIN define to specify the minimum
 *                      depth for the Reply Descriptor Post Queue.
 *                      Added SASAddress field to Initiator Device Table
 *                      Overflow Event data.
 *  10-31-07  02.00.04  Added ReasonCode MPI2_EVENT_SAS_INIT_RC_NOT_RESPONDING
 *                      for SAS Initiator Device Status Change Event data.
 *                      Modified Reason Code defines for SAS Topology Change
 *                      List Event data, including adding a bit for PHY Vacant
 *                      status, and adding a mask for the Reason Code.
 *                      Added define for
 *                      MPI2_EVENT_SAS_TOPO_ES_DELAY_NOT_RESPONDING.
 *                      Added define for MPI2_EXT_IMAGE_TYPE_MEGARAID.
 *  12-18-07  02.00.05  Added Boot Status defines for the IOCExceptions field of
 *                      the IOCFacts Reply.
 *                      Removed MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER define.
 *                      Moved MPI2_VERSION_UNION to mpi2.h.
 *                      Changed MPI2_EVENT_NOTIFICATION_REQUEST to use masks
 *                      instead of enables, and added SASBroadcastPrimitiveMasks
 *                      field.
 *                      Added Log Entry Added Event and related structure.
 *  02-29-08  02.00.06  Added define MPI2_IOCFACTS_CAPABILITY_INTEGRATED_RAID.
 *                      Removed define MPI2_IOCFACTS_PROTOCOL_SMP_TARGET.
 *                      Added MaxVolumes and MaxPersistentEntries fields to
 *                      IOCFacts reply.
 *                      Added ProtocalFlags and IOCCapabilities fields to
 *                      MPI2_FW_IMAGE_HEADER.
 *                      Removed MPI2_PORTENABLE_FLAGS_ENABLE_SINGLE_PORT.
 *  03-03-08  02.00.07  Fixed MPI2_FW_IMAGE_HEADER by changing Reserved26 to
 *                      a U16 (from a U32).
 *                      Removed extra 's' from EventMasks name.
 *  06-27-08  02.00.08  Fixed an offset in a comment.
 *  10-02-08  02.00.09  Removed SystemReplyFrameSize from MPI2_IOC_INIT_REQUEST.
 *                      Removed CurReplyFrameSize from MPI2_IOC_FACTS_REPLY and
 *                      renamed MinReplyFrameSize to ReplyFrameSize.
 *                      Added MPI2_IOCFACTS_EXCEPT_IR_FOREIGN_CONFIG_MAX.
 *                      Added two new RAIDOperation values for Integrated RAID
 *                      Operations Status Event data.
 *                      Added four new IR Configuration Change List Event data
 *                      ReasonCode values.
 *                      Added two new ReasonCode defines for SAS Device Status
 *                      Change Event data.
 *                      Added three new DiscoveryStatus bits for the SAS
 *                      Discovery event data.
 *                      Added Multiplexing Status Change bit to the PhyStatus
 *                      field of the SAS Topology Change List event data.
 *                      Removed define for MPI2_INIT_IMAGE_BOOTFLAGS_XMEMCOPY.
 *                      BootFlags are now product-specific.
 *                      Added defines for the indivdual signature bytes
 *                      for MPI2_INIT_IMAGE_FOOTER.
 *  01-19-09  02.00.10  Added MPI2_IOCFACTS_CAPABILITY_EVENT_REPLAY define.
 *                      Added MPI2_EVENT_SAS_DISC_DS_DOWNSTREAM_INITIATOR
 *                      define.
 *                      Added MPI2_EVENT_SAS_DEV_STAT_RC_SATA_INIT_FAILURE
 *                      define.
 *                      Removed MPI2_EVENT_SAS_DISC_DS_SATA_INIT_FAILURE define.
 *  05-06-09  02.00.11  Added MPI2_IOCFACTS_CAPABILITY_RAID_ACCELERATOR define.
 *                      Added MPI2_IOCFACTS_CAPABILITY_MSI_X_INDEX define.
 *                      Added two new reason codes for SAS Device Status Change
```

```
392  *                     Event.
393  *                     Added new event: SAS PHY Counter.
394  *  07-30-09  02.00.12  Added GPIO Interrupt event define and structure.
395  *                     Added MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER define.
396  *                     Added new product id family for 2208.
397  *  10-28-09  02.00.13  Added HostMSIxVectors field to MPI2_IOC_INIT_REQUEST.
398  *                     Added MaxMSIxVectors field to MPI2_IOC_FACTS_REPLY.
399  *                     Added MinDevHandle field to MPI2_IOC_FACTS_REPLY.
400  *                     Added MPI2_IOCFACTS_CAPABILITY_HOST_BASED_DISCOVERY.
401  *                     Added MPI2_EVENT_HOST_BASED_DISCOVERY_PHY define.
402  *                     Added MPI2_EVENT_SAS_TOPO_ES_NO_EXPANDER define.
403  *                     Added Host Based Discovery Phy Event data.
404  *                     Added defines for ProductID Product field
405  *                     (MPI2_FW_HEADER_PID_).
406  *                     Modified values for SAS ProductID Family
407  *                     (MPI2_FW_HEADER_PID_FAMILY_).
408  *  02-10-10  02.00.14  Added SAS Quiesce Event structure and defines.
409  *                     Added PowerManagementControl Request structures and
410  *                     defines.
411  *  05-12-10  02.00.15  Marked Task Set Full Event as obsolete.
412  *                     Added MPI2_EVENT_SAS_TOPO_LR_UNSUPPORTED_PHY define.
413  *  11-10-10  02.00.16  Added MPI2_FW_DOWNLOAD_ITYPE_MIN_PRODUCT_SPECIFIC.
414  *  02-23-11  02.00.17  Added SAS NOTIFY Primitive event, and added
415  *                     SASNotifyPrimitiveMasks field to
416  *                     MPI2_EVENT_NOTIFICATION_REQUEST.
417  *                     Added Temperature Threshold Event.
418  *                     Added Host Message Event.
419  *                     Added Send Host Message request and reply.
420  *  05-25-11  02.00.18  For Extended Image Header, added
421  *                     MPI2_EXT_IMAGE_TYPE_MIN_PRODUCT_SPECIFIC and
422  *                     MPI2_EXT_IMAGE_TYPE_MAX_PRODUCT_SPECIFIC defines.
423  *                     Deprecated MPI2_EXT_IMAGE_TYPE_MAX define.
424  *  08-24-11  02.00.19  Added PhysicalPort field to
425  *                     MPI2_EVENT_DATA_SAS_DEVICE_STATUS_CHANGE structure.
426  *                     Marked MPI2_PM_CONTROL_FEATURE_PCIE_LINK as obsolete.
427  *  11-18-11  02.00.20  Incorporating additions for MPI v2.5.
428  *  03-29-12  02.00.21  Added a product specific range to event values.
429  *  07-26-12  02.00.22  Added MPI2_IOCFACTS_EXCEPT_PARTIAL_MEMORY_FAILURE.
430  *                     Added ElapsedSeconds field to
431  *                     MPI2_EVENT_DATA_IR_OPERATION_STATUS.
432  *  08-19-13  02.00.23  For IOCInit, added MPI2_IOCINIT_MSGFLAG_RDPQ_ARRAY_MODE
433  *                     and MPI2_IOC_INIT_RDPQ_ARRAY_ENTRY.
434  *                     Added MPI2_IOCFACTS_CAPABILITY_RDPQ_ARRAY_CAPABLE.
435  *                     Added MPI2_FW_DOWNLOAD_ITYPE_PUBLIC_KEY.
436  *                     Added Encrypted Hash Extended Image.
437  *  12-05-13  02.00.24  Added MPI25_HASH_IMAGE_TYPE_BIOS.
438  *  --------------------------------------------------------------------

440 mpi2_raid.h
441  *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
442  *  08-31-07  02.00.01  Modifications to RAID Action request and reply,
443  *                     including the Actions and ActionData.
444  *  02-29-08  02.00.02  Added MPI2_RAID_ACTION_ADATA_DISABL_FULL_REBUILD.
445  *  05-21-08  02.00.03  Added MPI2_RAID_VOL_CREATION_NUM_PHYSDISKS so that
446  *                     the PhysDisk array in MPI2_RAID_VOLUME_CREATION_STRUCT
447  *                     can be sized by the build environment.
448  *  07-30-09  02.00.04  Added proper define for the Use Default Settings bit of
449  *                     VolumeCreationFlags and marked the old one as obsolete.
450  *  05-12-10  02.00.05  Added MPI2_RAID_VOL_FLAGS_OP_MDC define.
451  *  08-24-10  02.00.06  Added MPI2_RAID_ACTION_COMPATIBILITY_CHECK along with
452  *                     related structures and defines.
453  *                     Added product-specific range to RAID Action values.
454  *  11-18-11  02.00.07  Incorporating additions for MPI v2.5.
455  *  02-06-12  02.00.08  Added MPI2_RAID_ACTION_PHYSDISK_HIDDEN.
456  *  07-26-12  02.00.09  Added ElapsedSeconds field to MPI2_RAID_VOL_INDICATOR.
457  *                     Added MPI2_RAID_VOL_FLAGS_ELAPSED_SECONDS_VALID define.
```



```
458  *  04-17-13  02.00.10  Added MPI25_RAID_ACTION_ADATA_ALLOW_PI.
459  *  --------------------------------------------------------------------

461 mpi2_sas.h
462  *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
463  *  06-26-07  02.00.01  Added Clear All Persistent Operation to SAS IO Unit
464  *                     Control Request.
465  *  10-02-08  02.00.02  Added Set IOC Parameter Operation to SAS IO Unit Control
466  *                     Request.
467  *  10-28-09  02.00.03  Changed the type of SGL in MPI2_SATA_PASSTHROUGH_REQUEST
468  *                     to MPI2_SGE_IO_UNION since it supports chained SGLs.
469  *  05-12-10  02.00.04  Modified some comments.
470  *  08-11-10  02.00.05  Added NCQ operations to SAS IO Unit Control.
471  *  11-18-11  02.00.06  Incorporating additions for MPI v2.5.
472  *  07-10-12  02.00.07  Added MPI2_SATA_PT_SGE_UNION for use in the SATA
473  *                     Passthrough Request message.
474  *  08-19-13  02.00.08  Made MPI2_SAS_OP_TRANSMIT_PORT_SELECT_SIGNAL obsolete
475  *                     for anything newer than MPI v2.0.
476  *  --------------------------------------------------------------------

478 mpi2_targ.h
479  *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
480  *  08-31-07  02.00.01  Added Command Buffer Data Location Address Space bits to
481  *                     BufferPostFlags field of CommandBufferPostBase Request.
482  *  02-29-08  02.00.02  Modified various names to make them 32-character unique.
483  *  10-02-08  02.00.03  Removed NextCmdBufferOffset from
484  *                     MPI2_TARGET_CMD_BUF_POST_BASE_REQUEST.
485  *                     Target Status Send Request only takes a single SGE for
486  *                     response data.
487  *  02-10-10  02.00.04  Added comment to MPI2_TARGET_SSP_RSP_IU structure.
488  *  11-18-11  02.00.05  Incorporating additions for MPI v2.5.
489  *  11-27-12  02.00.06  Added InitiatorDevHandle field to MPI2_TARGET_MODE_ABORT
490  *                     request message structure.
491  *                     Added AbortType MPI2_TARGET_MODE_ABORT_DEVHANDLE and
492  *                     MPI2_TARGET_MODE_ABORT_ALL_COMMANDS.
493  *  --------------------------------------------------------------------

495 mpi2_tool.h
496  *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
497  *  12-18-07  02.00.01  Added Diagnostic Buffer Post and Diagnostic Release
498  *                     structures and defines.
499  *  02-29-08  02.00.02  Modified various names to make them 32-character unique.
500  *  05-06-09  02.00.03  Added ISTWI Read Write Tool and Diagnostic CLI Tool.
501  *  07-30-09  02.00.04  Added ExtendedType field to DiagnosticBufferPost request
502  *                     and reply messages.
503  *                     Added MPI2_DIAG_BUF_TYPE_EXTENDED.
504  *                     Incremented MPI2_DIAG_BUF_TYPE_COUNT.
505  *  05-12-10  02.00.05  Added Diagnostic Data Upload tool.
506  *  08-11-10  02.00.06  Added defines that were missing for Diagnostic Buffer
507  *                     Post Request.
508  *  05-25-11  02.00.07  Added Flags field and related defines to
509  *                     MPI2_TOOLBOX_ISTWI_READ_WRITE_REQUEST.
510  *  11-18-11  02.00.08  Incorporating additions for MPI v2.5.
511  *  07-10-12  02.00.09  Add MPI v2.5 Toolbox Diagnostic CLI Tool Request
512  *                     message.
513  *  07-26-12  02.00.10  Modified MPI2_TOOLBOX_DIAGNOSTIC_CLI_REQUEST so that
514  *                     it uses MPI Chain SGE as well as MPI Simple SGE.
515  *  08-19-13  02.00.11  Added MPI2_TOOLBOX_TEXT_DISPLAY_TOOL and related info.
516  *  --------------------------------------------------------------------

518 mpi2_type.h
519  *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
520  *  --------------------------------------------------------------------

522 mpi2_ra.h
523  *  05-06-09  02.00.00  Initial version.
```

```
 *  --------------------------------------------------------------------------

mpi2_hbd.h
 *  10-28-09  02.00.00  Initial version.
 *  08-11-10  02.00.01  Removed PortGroups, DmaGroup, and ControlGroup from
 *                      HBD Action request, replaced by AdditionalInfo field.
 *  11-18-11  02.00.02  Incorporating additions for MPI v2.5.
 *  --------------------------------------------------------------------------


mpi2_history.txt         Parts list history

Filename     02.00.33  02.00.32  02.00.31  02.00.30
----------   --------  --------  --------  --------
mpi2.h       02.00.33  02.00.32  02.00.31  02.00.30
mpi2_cnfg.h  02.00.27  02.00.26  02.00.25  02.00.25
mpi2_init.h  02.00.15  02.00.15  02.00.15  02.00.15
mpi2_ioc.h   02.00.24  02.00.23  02.00.22  02.00.22
mpi2_raid.h  02.00.10  02.00.10  02.00.10  02.00.09
mpi2_sas.h   02.00.08  02.00.08  02.00.07  02.00.07
mpi2_targ.h  02.00.06  02.00.06  02.00.06  02.00.06
mpi2_tool.h  02.00.11  02.00.11  02.00.10  02.00.10
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_ra.h    02.00.00  02.00.00  02.00.00  02.00.00
mpi2_hbd.h   02.00.02  02.00.02  02.00.02  02.00.02

Filename     02.00.29  02.00.28  02.00.27  02.00.26  02.00.25  02.00.24
----------   --------  --------  --------  --------  --------  --------
mpi2.h       02.00.29  02.00.28  02.00.27  02.00.26  02.00.25  02.00.24
mpi2_cnfg.h  02.00.24  02.00.23  02.00.22  02.00.22  02.00.22  02.00.22
mpi2_init.h  02.00.14  02.00.14  02.00.14  02.00.14  02.00.13  02.00.13
mpi2_ioc.h   02.00.22  02.00.22  02.00.22  02.00.21  02.00.21  02.00.20
mpi2_raid.h  02.00.09  02.00.09  02.00.09  02.00.08  02.00.08  02.00.08
mpi2_sas.h   02.00.07  02.00.07  02.00.07  02.00.07  02.00.06  02.00.06
mpi2_targ.h  02.00.06  02.00.06  02.00.05  02.00.05  02.00.05  02.00.05
mpi2_tool.h  02.00.10  02.00.10  02.00.10  02.00.09  02.00.08  02.00.08
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_ra.h    02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_hbd.h   02.00.02  02.00.02  02.00.02  02.00.02  02.00.02  02.00.02

Filename     02.00.23  02.00.22  02.00.21  02.00.20  02.00.19  02.00.18
----------   --------  --------  --------  --------  --------  --------
mpi2.h       02.00.23  02.00.22  02.00.21  02.00.20  02.00.19  02.00.18
mpi2_cnfg.h  02.00.22  02.00.21  02.00.20  02.00.19  02.00.18  02.00.17
mpi2_init.h  02.00.12  02.00.11  02.00.11  02.00.11  02.00.11  02.00.11
mpi2_ioc.h   02.00.20  02.00.19  02.00.18  02.00.17  02.00.17  02.00.16
mpi2_raid.h  02.00.07  02.00.06  02.00.05  02.00.05  02.00.05  02.00.05
mpi2_sas.h   02.00.06  02.00.05  02.00.05  02.00.05  02.00.05  02.00.05
mpi2_targ.h  02.00.05  02.00.04  02.00.04  02.00.04  02.00.04  02.00.04
mpi2_tool.h  02.00.08  02.00.07  02.00.07  02.00.06  02.00.06  02.00.06
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_ra.h    02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_hbd.h   02.00.02  02.00.01  02.00.01  02.00.01  02.00.01  02.00.01

Filename     02.00.17  02.00.16  02.00.15  02.00.14  02.00.13  02.00.12
----------   --------  --------  --------  --------  --------  --------
mpi2.h       02.00.17  02.00.16  02.00.15  02.00.14  02.00.13  02.00.12
mpi2_cnfg.h  02.00.16  02.00.15  02.00.14  02.00.13  02.00.12  02.00.11
mpi2_init.h  02.00.10  02.00.10  02.00.09  02.00.08  02.00.07  02.00.07
mpi2_ioc.h   02.00.15  02.00.15  02.00.14  02.00.13  02.00.12  02.00.11
mpi2_raid.h  02.00.05  02.00.05  02.00.04  02.00.04  02.00.04  02.00.03
mpi2_sas.h   02.00.05  02.00.04  02.00.03  02.00.03  02.00.02  02.00.02
mpi2_targ.h  02.00.04  02.00.04  02.00.04  02.00.03  02.00.03  02.00.03
mpi2_tool.h  02.00.06  02.00.05  02.00.04  02.00.04  02.00.04  02.00.03
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_ra.h    02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
```



```
mpi2_hbd.h   02.00.01  02.00.00  02.00.00  02.00.00

Filename     02.00.11  02.00.10  02.00.09  02.00.08  02.00.07  02.00.06
----------   --------  --------  --------  --------  --------  --------
mpi2.h       02.00.11  02.00.10  02.00.09  02.00.08  02.00.07  02.00.06
mpi2_cnfg.h  02.00.10  02.00.09  02.00.08  02.00.07  02.00.06  02.00.06
mpi2_init.h  02.00.06  02.00.06  02.00.05  02.00.05  02.00.04  02.00.03
mpi2_ioc.h   02.00.10  02.00.09  02.00.08  02.00.07  02.00.07  02.00.06
mpi2_raid.h  02.00.03  02.00.03  02.00.03  02.00.03  02.00.02  02.00.02
mpi2_sas.h   02.00.02  02.00.02  02.00.01  02.00.01  02.00.01  02.00.01
mpi2_targ.h  02.00.03  02.00.03  02.00.02  02.00.02  02.00.02  02.00.02
mpi2_tool.h  02.00.02  02.00.02  02.00.02  02.00.02  02.00.02  02.00.02
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00

Filename     02.00.05  02.00.04  02.00.03  02.00.02  02.00.01  02.00.00
----------   --------  --------  --------  --------  --------  --------
mpi2.h       02.00.05  02.00.04  02.00.03  02.00.02  02.00.01  02.00.00
mpi2_cnfg.h  02.00.05  02.00.04  02.00.03  02.00.02  02.00.01  02.00.00
mpi2_init.h  02.00.02  02.00.01  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_ioc.h   02.00.05  02.00.04  02.00.03  02.00.02  02.00.01  02.00.00
mpi2_raid.h  02.00.01  02.00.01  02.00.01  02.00.00  02.00.00  02.00.00
mpi2_sas.h   02.00.01  02.00.01  02.00.01  02.00.01  02.00.00  02.00.00
mpi2_targ.h  02.00.01  02.00.01  02.00.01  02.00.00  02.00.00  02.00.00
mpi2_tool.h  02.00.01  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
mpi2_type.h  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00  02.00.00
```

```
**********************************************************
   30283 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_init.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_init.h
 *         Title:  MPI SCSI initiator mode messages and structures
 *  Creation Date:  June 23, 2006
 *
 *  mpi2_init.h Version:  02.00.15
 *    mpi2_init.h Version:  02.00.07
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  10-31-07  02.00.01  Fixed name for pMpi2SCSITaskManagementRequest_t.
 *  12-18-07  02.00.02  Modified Task Management Target Reset Method defines.
 *  02-29-08  02.00.03  Added Query Task Set and Query Unit Attention.
 *  03-03-08  02.00.04  Fixed name of struct _MPI2_SCSI_TASK_MANAGE_REPLY.
 *  05-21-08  02.00.05  Fixed typo in name of Mpi2SepRequest_t.
 *  10-02-08  02.00.06  Removed Untagged and No Disconnect values from SCSI IO
 *                      Control field Task Attribute flags.
 *                      Moved LUN field defines to mpi2.h becasue they are
 *                      common to many structures.
 *  05-06-09  02.00.07  Changed task management type of Query Unit Attention to
 *                      Query Asynchronous Event.
 *                      Defined two new bits in the SlotStatus field of the SCSI
 *                      Enclosure Processor Request and Reply.
 *  10-28-09  02.00.08  Added defines for decoding the ResponseInfo bytes for
 *                      both SCSI IO Error Reply and SCSI Task Management Reply.
 *                      Added ResponseInfo field to MPI2_SCSI_TASK_MANAGE_REPLY.
 *                      Added MPI2_SCSITASKMGMT_RSP_TM_OVERLAPPED_TAG define.
 *  02-10-10  02.00.09  Removed unused structure that had "#if 0" around it.
 *  05-12-10  02.00.10  Added optional vendor-unique region to SCSI IO Request.
 *  11-10-10  02.00.11  Added MPI2_SCSIIO_NUM_SGLOFFSETS define.
 *  11-18-11  02.00.12  Incorporating additions for MPI v2.5.
 *  02-06-12  02.00.13  Added alternate defines for Task Priority / Command
 *                      Priority to match SAM-4.
 *                      Added EEDPErrorOffset to MPI2_SCSI_IO_REPLY.
 *  07-10-12  02.00.14  Added MPI2_SCSIIO_CONTROL_SHIFT_DATADIRECTION.
 *  04-09-13  02.00.15  Added SCSIStatusQualifier field to MPI2_SCSI_IO_REPLY,
 *                      replacing the Reserved4 field.
 *  ---------------------------------------------------------------------------
 */

#ifndef MPI2_INIT_H
#define MPI2_INIT_H

/*****************************************************************************
```

```
 85 *
 86 *               SCSI Initiator Messages
 87 *
 88 ****************************************************************************/

 90 /****************************************************************************
 91 *  SCSI IO messages and associated structures
 92 ****************************************************************************/

 94 typedef struct _MPI2_SCSI_IO_CDB_EEDP32
 91 typedef struct
 95 {
 96     U8                      CDB[20];                /* 0x00 */
 97     U32                     PrimaryReferenceTag;    /* 0x14 */
 98     U16                     PrimaryApplicationTag;  /* 0x18 */
 99     U16                     PrimaryApplicationTagMask;  /* 0x1A */
100     U32                     TransferLength;         /* 0x1C */
101 } MPI2_SCSI_IO_CDB_EEDP32, MPI2_POINTER PTR_MPI2_SCSI_IO_CDB_EEDP32,
102   Mpi2ScsiIoCdbEedp32_t, MPI2_POINTER pMpi2ScsiIoCdbEedp32_t;

104 /* MPI v2.0 CDB field */
105 typedef union _MPI2_SCSI_IO_CDB_UNION
101 /* TBD: I don't think this is needed for MPI2/Gen2 */
102 #if 0
103 typedef struct
106 {
105     U8                      CDB[16];                /* 0x00 */
106     U32                     DataLength;             /* 0x10 */
107     U32                     PrimaryReferenceTag;    /* 0x14 */
108     U16                     PrimaryApplicationTag;  /* 0x18 */
109     U16                     PrimaryApplicationTagMask;  /* 0x1A */
110     U32                     TransferLength;         /* 0x1C */
111 } MPI2_SCSI_IO32_CDB_EEDP16, MPI2_POINTER PTR_MPI2_SCSI_IO32_CDB_EEDP16,
112   Mpi2ScsiIo32CdbEedp16_t, MPI2_POINTER pMpi2ScsiIo32CdbEedp16_t;
113 #endif

115 typedef union
116 {
107     U8                      CDB32[32];
108     MPI2_SCSI_IO_CDB_EEDP32 EEDP32;
109     MPI2_SGE_SIMPLE_UNION   SGE;
110 } MPI2_SCSI_IO_CDB_UNION, MPI2_POINTER PTR_MPI2_SCSI_IO_CDB_UNION,
111   Mpi2ScsiIoCdb_t, MPI2_POINTER pMpi2ScsiIoCdb_t;

113 /* MPI v2.0 SCSI IO Request Message */
123 /* SCSI IO Request Message */
114 typedef struct _MPI2_SCSI_IO_REQUEST
115 {
116     U16                     DevHandle;                  /* 0x00 */
117     U8                      ChainOffset;                /* 0x02 */
118     U8                      Function;                   /* 0x03 */
119     U16                     Reserved1;                  /* 0x04 */
120     U8                      Reserved2;                  /* 0x06 */
121     U8                      MsgFlags;                   /* 0x07 */
122     U8                      VP_ID;                      /* 0x08 */
123     U8                      VF_ID;                      /* 0x09 */
124     U16                     Reserved3;                  /* 0x0A */
125     U32                     SenseBufferLowAddress;      /* 0x0C */
126     U16                     SGLFlags;                   /* 0x10 */
127     U8                      SenseBufferLength;          /* 0x12 */
128     U8                      Reserved4;                  /* 0x13 */
129     U8                      SGLOffset0;                 /* 0x14 */
130     U8                      SGLOffset1;                 /* 0x15 */
131     U8                      SGLOffset2;                 /* 0x16 */
132     U8                      SGLOffset3;                 /* 0x17 */
133     U32                     SkipCount;                  /* 0x18 */
```



```
134     U32                     DataLength;                 /* 0x1C */
135     U32                     BidirectionalDataLength;    /* 0x20 */
136     U16                     IoFlags;                    /* 0x24 */
137     U16                     EEDPFlags;                  /* 0x26 */
138     U32                     EEDPBlockSize;              /* 0x28 */
139     U32                     SecondaryReferenceTag;      /* 0x2C */
140     U16                     SecondaryApplicationTag;    /* 0x30 */
141     U16                     ApplicationTagTranslationMask;  /* 0x32 */
142     U8                      LUN[8];                     /* 0x34 */
143     U32                     Control;                    /* 0x3C */
144     MPI2_SCSI_IO_CDB_UNION  CDB;                        /* 0x40 */

146 #ifdef MPI2_SCSI_IO_VENDOR_UNIQUE_REGION /* typically this is left undefined */
147     MPI2_SCSI_IO_VENDOR_UNIQUE VendorRegion;
148 #endif

150     MPI2_SGE_IO_UNION       SGL;                        /* 0x60 */

152 } MPI2_SCSI_IO_REQUEST, MPI2_POINTER PTR_MPI2_SCSI_IO_REQUEST,
153   Mpi2SCSIIORequest_t, MPI2_POINTER pMpi2SCSIIORequest_t;

155 /* SCSI IO MsgFlags bits */

157 /* MsgFlags for SenseBufferAddressSpace */
158 #define MPI2_SCSIIO_MSGFLAGS_MASK_SENSE_ADDR        (0x0C)
159 #define MPI2_SCSIIO_MSGFLAGS_SYSTEM_SENSE_ADDR      (0x00)
160 #define MPI2_SCSIIO_MSGFLAGS_IOCDDR_SENSE_ADDR      (0x04)
161 #define MPI2_SCSIIO_MSGFLAGS_IOCPLB_SENSE_ADDR      (0x08)
162 #define MPI2_SCSIIO_MSGFLAGS_IOCPLBNTA_SENSE_ADDR   (0x0C)

164 /* SCSI IO SGLFlags bits */

166 /* base values for Data Location Address Space */
167 #define MPI2_SCSIIO_SGLFLAGS_ADDR_MASK              (0x0C)
168 #define MPI2_SCSIIO_SGLFLAGS_SYSTEM_ADDR            (0x00)
169 #define MPI2_SCSIIO_SGLFLAGS_IOCDDR_ADDR            (0x04)
170 #define MPI2_SCSIIO_SGLFLAGS_IOCPLB_ADDR            (0x08)
171 #define MPI2_SCSIIO_SGLFLAGS_IOCPLBNTA_ADDR         (0x0C)

173 /* base values for Type */
174 #define MPI2_SCSIIO_SGLFLAGS_TYPE_MASK              (0x03)
175 #define MPI2_SCSIIO_SGLFLAGS_TYPE_MPI               (0x00)
176 #define MPI2_SCSIIO_SGLFLAGS_TYPE_IEEE32            (0x01)
177 #define MPI2_SCSIIO_SGLFLAGS_TYPE_IEEE64            (0x02)

179 /* shift values for each sub-field */
180 #define MPI2_SCSIIO_SGLFLAGS_SGL3_SHIFT             (12)
181 #define MPI2_SCSIIO_SGLFLAGS_SGL2_SHIFT             (8)
182 #define MPI2_SCSIIO_SGLFLAGS_SGL1_SHIFT             (4)
183 #define MPI2_SCSIIO_SGLFLAGS_SGL0_SHIFT             (0)

185 /* number of SGLOffset fields */
186 #define MPI2_SCSIIO_NUM_SGLOFFSETS                  (4)

188 /* SCSI IO IoFlags bits */

190 /* Large CDB Address Space */
191 #define MPI2_SCSIIO_CDB_ADDR_MASK                   (0x6000)
192 #define MPI2_SCSIIO_CDB_ADDR_SYSTEM                 (0x0000)
193 #define MPI2_SCSIIO_CDB_ADDR_IOCDDR                 (0x2000)
194 #define MPI2_SCSIIO_CDB_ADDR_IOCPLB                 (0x4000)
195 #define MPI2_SCSIIO_CDB_ADDR_IOCPLBNTA              (0x6000)

197 #define MPI2_SCSIIO_IOFLAGS_LARGE_CDB               (0x1000)
198 #define MPI2_SCSIIO_IOFLAGS_BIDIRECTIONAL           (0x0800)
199 #define MPI2_SCSIIO_IOFLAGS_MULTICAST               (0x0400)
```

```
200 #define MPI2_SCSIIO_IOFLAGS_CMD_DETERMINES_DATA_DIR (0x0200)
201 #define MPI2_SCSIIO_IOFLAGS_CDBLENGTH_MASK          (0x01FF)

203 /* SCSI IO EEDPFlags bits */

205 #define MPI2_SCSIIO_EEDPFLAGS_INC_PRI_REFTAG        (0x8000)
206 #define MPI2_SCSIIO_EEDPFLAGS_INC_SEC_REFTAG        (0x4000)
207 #define MPI2_SCSIIO_EEDPFLAGS_INC_PRI_APPTAG        (0x2000)
208 #define MPI2_SCSIIO_EEDPFLAGS_INC_SEC_APPTAG        (0x1000)

210 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_REFTAG          (0x0400)
211 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_APPTAG          (0x0200)
212 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_GUARD           (0x0100)

214 #define MPI2_SCSIIO_EEDPFLAGS_PASSTHRU_REFTAG       (0x0008)

216 #define MPI2_SCSIIO_EEDPFLAGS_MASK_OP               (0x0007)
217 #define MPI2_SCSIIO_EEDPFLAGS_NOOP_OP               (0x0000)
218 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_OP              (0x0001)
219 #define MPI2_SCSIIO_EEDPFLAGS_STRIP_OP              (0x0002)
220 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_REMOVE_OP       (0x0003)
221 #define MPI2_SCSIIO_EEDPFLAGS_INSERT_OP             (0x0004)
222 #define MPI2_SCSIIO_EEDPFLAGS_REPLACE_OP            (0x0006)
223 #define MPI2_SCSIIO_EEDPFLAGS_CHECK_REGEN_OP        (0x0007)

225 /* SCSI IO LUN fields: use MPI2_LUN_ from mpi2.h */

227 /* SCSI IO Control bits */
228 #define MPI2_SCSIIO_CONTROL_ADDCDBLEN_MASK      (0xFC000000)
229 #define MPI2_SCSIIO_CONTROL_ADDCDBLEN_SHIFT     (26)

231 #define MPI2_SCSIIO_CONTROL_DATADIRECTION_MASK  (0x03000000)
232 #define MPI2_SCSIIO_CONTROL_SHIFT_DATADIRECTION (24)
233 #define MPI2_SCSIIO_CONTROL_NODATATRANSFER      (0x00000000)
234 #define MPI2_SCSIIO_CONTROL_WRITE               (0x01000000)
235 #define MPI2_SCSIIO_CONTROL_READ                (0x02000000)
236 #define MPI2_SCSIIO_CONTROL_BIDIRECTIONAL       (0x03000000)

238 #define MPI2_SCSIIO_CONTROL_TASKPRI_MASK        (0x00007800)
239 #define MPI2_SCSIIO_CONTROL_TASKPRI_SHIFT       (11)
240 /* alternate name for the previous field; called Command Priority in SAM-4 */
241 #define MPI2_SCSIIO_CONTROL_CMDPRI_MASK         (0x00007800)
242 #define MPI2_SCSIIO_CONTROL_CMDPRI_SHIFT        (11)

244 #define MPI2_SCSIIO_CONTROL_TASKATTRIBUTE_MASK  (0x00000700)
245 #define MPI2_SCSIIO_CONTROL_SIMPLEQ             (0x00000000)
246 #define MPI2_SCSIIO_CONTROL_HEADOFQ             (0x00000100)
247 #define MPI2_SCSIIO_CONTROL_ORDEREDQ            (0x00000200)
248 #define MPI2_SCSIIO_CONTROL_ACAQ                (0x00000400)

250 #define MPI2_SCSIIO_CONTROL_TLR_MASK            (0x000000C0)
251 #define MPI2_SCSIIO_CONTROL_NO_TLR              (0x00000000)
252 #define MPI2_SCSIIO_CONTROL_TLR_ON              (0x00000040)
253 #define MPI2_SCSIIO_CONTROL_TLR_OFF             (0x00000080)


256 /* MPI v2.5 CDB field */
257 typedef union _MPI25_SCSI_IO_CDB_UNION
258 {
259     U8                      CDB32[32];
260     MPI2_SCSI_IO_CDB_EEDP32 EEDP32;
261     MPI2_IEEE_SGE_SIMPLE64  SGE;
262 } MPI25_SCSI_IO_CDB_UNION, MPI2_POINTER PTR_MPI25_SCSI_IO_CDB_UNION,
263   Mpi25ScsiIoCdb_t, MPI2_POINTER pMpi25ScsiIoCdb_t;

265 /* MPI v2.5 SCSI IO Request Message */
```



```
266 typedef struct _MPI25_SCSI_IO_REQUEST
267 {
268     U16                     DevHandle;                  /* 0x00 */
269     U8                      ChainOffset;                /* 0x02 */
270     U8                      Function;                   /* 0x03 */
271     U16                     Reserved1;                  /* 0x04 */
272     U8                      Reserved2;                  /* 0x06 */
273     U8                      MsgFlags;                   /* 0x07 */
274     U8                      VP_ID;                      /* 0x08 */
275     U8                      VF_ID;                      /* 0x09 */
276     U16                     Reserved3;                  /* 0x0A */
277     U32                     SenseBufferLowAddress;      /* 0x0C */
278     U8                      DMAFlags;                   /* 0x10 */
279     U8                      Reserved5;                  /* 0x11 */
280     U8                      SenseBufferLength;          /* 0x12 */
281     U8                      Reserved4;                  /* 0x13 */
282     U8                      SGLOffset0;                 /* 0x14 */
283     U8                      SGLOffset1;                 /* 0x15 */
284     U8                      SGLOffset2;                 /* 0x16 */
285     U8                      SGLOffset3;                 /* 0x17 */
286     U32                     SkipCount;                  /* 0x18 */
287     U32                     DataLength;                 /* 0x1C */
288     U32                     BidirectionalDataLength;    /* 0x20 */
289     U16                     IoFlags;                    /* 0x24 */
290     U16                     EEDPFlags;                  /* 0x26 */
291     U16                     EEDPBlockSize;              /* 0x28 */
292     U16                     Reserved6;                  /* 0x2A */
293     U32                     SecondaryReferenceTag;      /* 0x2C */
294     U16                     SecondaryApplicationTag;    /* 0x30 */
295     U16                     ApplicationTagTranslationMask; /* 0x32 */
296     U8                      LUN[8];                     /* 0x34 */
297     U32                     Control;                    /* 0x3C */
298     MPI25_SCSI_IO_CDB_UNION CDB;                        /* 0x40 */

300 #ifdef MPI25_SCSI_IO_VENDOR_UNIQUE_REGION /* typically this is left undefined */
301     MPI25_SCSI_IO_VENDOR_UNIQUE VendorRegion;
302 #endif

304     MPI25_SGE_IO_UNION      SGL;                        /* 0x60 */

306 } MPI25_SCSI_IO_REQUEST, MPI2_POINTER PTR_MPI25_SCSI_IO_REQUEST,
307   Mpi25SCSIIORequest_t, MPI2_POINTER pMpi25SCSIIORequest_t;

309 /* use MPI2_SCSIIO_MSGFLAGS_ defines for the MsgFlags field */

311 /* Defines for the DMAFlags field
312  *  Each setting affects 4 SGLS, from SGL0 to SGL3.
313  *      D = Data
314  *      C = Cache DIF
315  *      I = Interleaved
316  *      H = Host DIF
317  */
318 #define MPI25_SCSIIO_DMAFLAGS_OP_MASK               (0x0F)
319 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_D_D            (0x00)
320 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_D_C            (0x01)
321 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_D_I            (0x02)
322 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_C_C            (0x03)
323 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_C_I            (0x04)
324 #define MPI25_SCSIIO_DMAFLAGS_OP_D_D_I_I            (0x05)
325 #define MPI25_SCSIIO_DMAFLAGS_OP_D_C_C_C            (0x06)
326 #define MPI25_SCSIIO_DMAFLAGS_OP_D_C_C_I            (0x07)
327 #define MPI25_SCSIIO_DMAFLAGS_OP_D_C_I_I            (0x08)
328 #define MPI25_SCSIIO_DMAFLAGS_OP_D_I_I_I            (0x09)
329 #define MPI25_SCSIIO_DMAFLAGS_OP_D_H_D_D            (0x0A)
330 #define MPI25_SCSIIO_DMAFLAGS_OP_D_H_D_C            (0x0B)
331 #define MPI25_SCSIIO_DMAFLAGS_OP_D_H_D_I            (0x0C)
```

```
#define MPI25_SCSIIO_DMAFLAGS_OP_D_H_C_C            (0x0D)
#define MPI25_SCSIIO_DMAFLAGS_OP_D_H_C_I            (0x0E)
#define MPI25_SCSIIO_DMAFLAGS_OP_D_H_I_I            (0x0F)

/* number of SGLOffset fields */
#define MPI25_SCSIIO_NUM_SGLOFFSETS                 (4)

/* defines for the IoFlags field */
#define MPI25_SCSIIO_IOFLAGS_IO_PATH_MASK           (0xC000)
#define MPI25_SCSIIO_IOFLAGS_NORMAL_PATH            (0x0000)
#define MPI25_SCSIIO_IOFLAGS_FAST_PATH              (0x4000)

#define MPI25_SCSIIO_IOFLAGS_LARGE_CDB                  (0x1000)
#define MPI25_SCSIIO_IOFLAGS_BIDIRECTIONAL              (0x0800)
#define MPI25_SCSIIO_IOFLAGS_CDBLENGTH_MASK             (0x01FF)

/* MPI v2.5 defines for the EEDPFlags bits */
/* use MPI2_SCSIIO_EEDPFLAGS_ defines for the other EEDPFlags bits */
#define MPI25_SCSIIO_EEDPFLAGS_ESCAPE_MODE_MASK             (0x00C0)
#define MPI25_SCSIIO_EEDPFLAGS_COMPATIBLE_MODE              (0x0000)
#define MPI25_SCSIIO_EEDPFLAGS_DO_NOT_DISABLE_MODE          (0x0040)
#define MPI25_SCSIIO_EEDPFLAGS_APPTAG_DISABLE_MODE          (0x0080)
#define MPI25_SCSIIO_EEDPFLAGS_APPTAG_REFTAG_DISABLE_MODE   (0x00C0)

#define MPI25_SCSIIO_EEDPFLAGS_HOST_GUARD_METHOD_MASK       (0x0030)
#define MPI25_SCSIIO_EEDPFLAGS_T10_CRC_HOST_GUARD           (0x0000)
#define MPI25_SCSIIO_EEDPFLAGS_IP_CHKSUM_HOST_GUARD         (0x0010)

/* use MPI2_LUN_ defines from mpi2.h for the LUN field */

/* use MPI2_SCSIIO_CONTROL_ defines for the Control field */


/* NOTE: The SCSI IO Reply is nearly the same for MPI 2.0 and MPI 2.5, so
 *       MPI2_SCSI_IO_REPLY is used for both.
 */

/* SCSI IO Error Reply Message */
typedef struct _MPI2_SCSI_IO_REPLY
{
    U16                     DevHandle;                  /* 0x00 */
    U8                      MsgLength;                  /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved1;                  /* 0x04 */
    U8                      Reserved2;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved3;                  /* 0x0A */
    U8                      SCSIStatus;                 /* 0x0C */
    U8                      SCSIState;                  /* 0x0D */
    U16                     IOCStatus;                  /* 0x0E */
    U32                     IOCLogInfo;                 /* 0x10 */
    U32                     TransferCount;              /* 0x14 */
    U32                     SenseCount;                 /* 0x18 */
    U32                     ResponseInfo;               /* 0x1C */
    U16                     TaskTag;                    /* 0x20 */
    U16                     SCSIStatusQualifier;        /* 0x22 */
    U16                     Reserved4;                  /* 0x22 */
    U32                     BidirectionalTransferCount; /* 0x24 */
    U32                     EEDPErrorOffset;            /* 0x28 */ /* MPI 2.
    U32                     Reserved5;                  /* 0x28 */
    U32                     Reserved6;                  /* 0x2C */
} MPI2_SCSI_IO_REPLY, MPI2_POINTER PTR_MPI2_SCSI_IO_REPLY,
  Mpi2SCSIIOReply_t, MPI2_POINTER pMpi2SCSIIOReply_t;
```



```
/* SCSI IO Reply SCSIStatus values (SAM-4 status codes) */

#define MPI2_SCSI_STATUS_GOOD                   (0x00)
#define MPI2_SCSI_STATUS_CHECK_CONDITION        (0x02)
#define MPI2_SCSI_STATUS_CONDITION_MET          (0x04)
#define MPI2_SCSI_STATUS_BUSY                   (0x08)
#define MPI2_SCSI_STATUS_INTERMEDIATE           (0x10)
#define MPI2_SCSI_STATUS_INTERMEDIATE_CONDMET   (0x14)
#define MPI2_SCSI_STATUS_RESERVATION_CONFLICT   (0x18)
#define MPI2_SCSI_STATUS_COMMAND_TERMINATED     (0x22) /* obsolete */
#define MPI2_SCSI_STATUS_TASK_SET_FULL          (0x28)
#define MPI2_SCSI_STATUS_ACA_ACTIVE             (0x30)
#define MPI2_SCSI_STATUS_TASK_ABORTED           (0x40)

/* SCSI IO Reply SCSIState flags */

#define MPI2_SCSI_STATE_RESPONSE_INFO_VALID     (0x10)
#define MPI2_SCSI_STATE_TERMINATED              (0x08)
#define MPI2_SCSI_STATE_NO_SCSI_STATUS          (0x04)
#define MPI2_SCSI_STATE_AUTOSENSE_FAILED        (0x02)
#define MPI2_SCSI_STATE_AUTOSENSE_VALID         (0x01)

/* masks and shifts for the ResponseInfo field */

#define MPI2_SCSI_RI_MASK_REASONCODE            (0x000000FF)
#define MPI2_SCSI_RI_SHIFT_REASONCODE           (0)

#define MPI2_SCSI_TASKTAG_UNKNOWN               (0xFFFF)


/*****************************************************************************
*  SCSI Task Management messages
*****************************************************************************/

/* SCSI Task Management Request Message */
typedef struct _MPI2_SCSI_TASK_MANAGE_REQUEST
{
    U16                     DevHandle;                  /* 0x00 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U8                      Reserved1;                  /* 0x04 */
    U8                      TaskType;                   /* 0x05 */
    U8                      Reserved2;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved3;                  /* 0x0A */
    U8                      LUN[8];                     /* 0x0C */
    U32                     Reserved4[7];               /* 0x14 */
    U16                     TaskMID;                    /* 0x30 */
    U16                     Reserved5;                  /* 0x32 */
} MPI2_SCSI_TASK_MANAGE_REQUEST,
  MPI2_POINTER PTR_MPI2_SCSI_TASK_MANAGE_REQUEST,
  Mpi2SCSITaskManagementRequest_t,
  MPI2_POINTER pMpi2SCSITaskManagementRequest_t;

/* TaskType values */

#define MPI2_SCSITASKMGMT_TASKTYPE_ABORT_TASK           (0x01)
#define MPI2_SCSITASKMGMT_TASKTYPE_ABRT_TASK_SET        (0x02)
#define MPI2_SCSITASKMGMT_TASKTYPE_TARGET_RESET         (0x03)
#define MPI2_SCSITASKMGMT_TASKTYPE_LOGICAL_UNIT_RESET   (0x05)
#define MPI2_SCSITASKMGMT_TASKTYPE_CLEAR_TASK_SET       (0x06)
#define MPI2_SCSITASKMGMT_TASKTYPE_QUERY_TASK           (0x07)
#define MPI2_SCSITASKMGMT_TASKTYPE_CLR_ACA              (0x08)
#define MPI2_SCSITASKMGMT_TASKTYPE_QRY_TASK_SET         (0x09)
```

```
462 #define MPI2_SCSITASKMGMT_TASKTYPE_QRY_ASYNC_EVENT      (0x0A)

464 /* obsolete TaskType name */
465 #define MPI2_SCSITASKMGMT_TASKTYPE_QRY_UNIT_ATTENTION   (MPI2_SCSITASKMGMT_TASKT

467 /* MsgFlags bits */

469 #define MPI2_SCSITASKMGMT_MSGFLAGS_MASK_TARGET_RESET    (0x18)
470 #define MPI2_SCSITASKMGMT_MSGFLAGS_LINK_RESET           (0x00)
471 #define MPI2_SCSITASKMGMT_MSGFLAGS_NEXUS_RESET_SRST     (0x08)
472 #define MPI2_SCSITASKMGMT_MSGFLAGS_SAS_HARD_LINK_RESET  (0x10)

474 #define MPI2_SCSITASKMGMT_MSGFLAGS_DO_NOT_SEND_TASK_IU  (0x01)



478 /* SCSI Task Management Reply Message */
479 typedef struct _MPI2_SCSI_TASK_MANAGE_REPLY
480 {
481     U16                     DevHandle;                  /* 0x00 */
482     U8                      MsgLength;                  /* 0x02 */
483     U8                      Function;                   /* 0x03 */
484     U8                      ResponseCode;               /* 0x04 */
485     U8                      TaskType;                   /* 0x05 */
486     U8                      Reserved1;                  /* 0x06 */
487     U8                      MsgFlags;                   /* 0x07 */
488     U8                      VP_ID;                      /* 0x08 */
489     U8                      VF_ID;                      /* 0x09 */
490     U16                     Reserved2;                  /* 0x0A */
491     U16                     Reserved3;                  /* 0x0C */
492     U16                     IOCStatus;                  /* 0x0E */
493     U32                     IOCLogInfo;                 /* 0x10 */
494     U32                     TerminationCount;           /* 0x14 */
495     U32                     ResponseInfo;               /* 0x18 */
496 } MPI2_SCSI_TASK_MANAGE_REPLY,
497   MPI2_POINTER PTR_MPI2_SCSI_TASK_MANAGE_REPLY,
498   Mpi2SCSITaskManagementReply_t, MPI2_POINTER pMpi2SCSIManagementReply_t;

500 /* ResponseCode values */

502 #define MPI2_SCSITASKMGMT_RSP_TM_COMPLETE               (0x00)
503 #define MPI2_SCSITASKMGMT_RSP_INVALID_FRAME             (0x02)
504 #define MPI2_SCSITASKMGMT_RSP_TM_NOT_SUPPORTED          (0x04)
505 #define MPI2_SCSITASKMGMT_RSP_TM_FAILED                 (0x05)
506 #define MPI2_SCSITASKMGMT_RSP_TM_SUCCEEDED              (0x08)
507 #define MPI2_SCSITASKMGMT_RSP_TM_INVALID_LUN            (0x09)
508 #define MPI2_SCSITASKMGMT_RSP_TM_OVERLAPPED_TAG         (0x0A)
509 #define MPI2_SCSITASKMGMT_RSP_IO_QUEUED_ON_IOC          (0x80)

511 /* masks and shifts for the ResponseInfo field */

513 #define MPI2_SCSITASKMGMT_RI_MASK_REASONCODE            (0x000000FF)
514 #define MPI2_SCSITASKMGMT_RI_SHIFT_REASONCODE           (0)
515 #define MPI2_SCSITASKMGMT_RI_MASK_ARI2                  (0x0000FF00)
516 #define MPI2_SCSITASKMGMT_RI_SHIFT_ARI2                 (8)
517 #define MPI2_SCSITASKMGMT_RI_MASK_ARI1                  (0x00FF0000)
518 #define MPI2_SCSITASKMGMT_RI_SHIFT_ARI1                 (16)
519 #define MPI2_SCSITASKMGMT_RI_MASK_ARI0                  (0xFF000000)
520 #define MPI2_SCSITASKMGMT_RI_SHIFT_ARI0                 (24)


523 /*****************************************************************************
524 *  SCSI Enclosure Processor messages
525 *****************************************************************************/

527 /* SCSI Enclosure Processor Request Message */
```



```
528 typedef struct _MPI2_SEP_REQUEST
529 {
530     U16                     DevHandle;          /* 0x00 */
531     U8                      ChainOffset;        /* 0x02 */
532     U8                      Function;           /* 0x03 */
533     U8                      Action;             /* 0x04 */
534     U8                      Flags;              /* 0x05 */
535     U8                      Reserved1;          /* 0x06 */
536     U8                      MsgFlags;           /* 0x07 */
537     U8                      VP_ID;              /* 0x08 */
538     U8                      VF_ID;              /* 0x09 */
539     U16                     Reserved2;          /* 0x0A */
540     U32                     SlotStatus;         /* 0x0C */
541     U32                     Reserved3;          /* 0x10 */
542     U32                     Reserved4;          /* 0x14 */
543     U32                     Reserved5;          /* 0x18 */
544     U16                     Slot;               /* 0x1C */
545     U16                     EnclosureHandle;    /* 0x1E */
546 } MPI2_SEP_REQUEST, MPI2_POINTER PTR_MPI2_SEP_REQUEST,
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   94287 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_ioc.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 */
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_ioc.h
 *         Title:  MPI IOC, Port, Event, FW Download, and FW Upload messages
 *  Creation Date:  October 11, 2006
 *
 *  mpi2_ioc.h Version:  02.00.24
 *  mpi2_ioc.h Version:  02.00.12
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-04-07  02.00.01  In IOCFacts Reply structure, renamed MaxDevices to
 *                      MaxTargets.
 *                      Added TotalImageSize field to FWDownload Request.
 *                      Added reserved words to FWUpload Request.
 *  06-26-07  02.00.02  Added IR Configuration Change List Event.
 *  08-31-07  02.00.03  Removed SystemReplyQueueDepth field from the IOCInit
 *                      request and replaced it with
 *                      ReplyDescriptorPostQueueDepth and ReplyFreeQueueDepth.
 *                      Replaced the MinReplyQueueDepth field of the IOCFacts
 *                      reply with MaxReplyDescriptorPostQueueDepth.
 *                      Added MPI2_RDPQ_DEPTH_MIN define to specify the minimum
 *                      depth for the Reply Descriptor Post Queue.
 *                      Added SASAddress field to Initiator Device Table
 *                      Overflow Event data.
 *  10-31-07  02.00.04  Added ReasonCode MPI2_EVENT_SAS_INIT_RC_NOT_RESPONDING
 *                      for SAS Initiator Device Status Change Event data.
 *                      Modified Reason Code defines for SAS Topology Change
 *                      List Event data, including adding a bit for PHY Vacant
 *                      status, and adding a mask for the Reason Code.
 *                      Added define for
 *                      MPI2_EVENT_SAS_TOPO_ES_DELAY_NOT_RESPONDING.
 *                      Added define for MPI2_EXT_IMAGE_TYPE_MEGARAID.
 *  12-18-07  02.00.05  Added Boot Status defines for the IOCExceptions field of
 *                      the IOCFacts Reply.
 *                      Removed MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER define.
 *                      Moved MPI2_VERSION_UNION to mpi2.h.
 *                      Changed MPI2_EVENT_NOTIFICATION_REQUEST to use masks
 *                      instead of enables, and added SASBroadcastPrimitiveMasks
 *                      field.
 *                      Added Log Entry Added Event and related structure.
 *  02-29-08  02.00.06  Added define MPI2_IOCFACTS_CAPABILITY_INTEGRATED_RAID.
 *                      Removed define MPI2_IOCFACTS_PROTOCOL_SMP_TARGET.
 *                      Added MaxVolumes and MaxPersistentEntries fields to
 *                      IOCFacts reply.
```

```
 *                      Added ProtocalFlags and IOCCapabilities fields to
 *                      MPI2_FW_IMAGE_HEADER.
 *                      Removed MPI2_PORTENABLE_FLAGS_ENABLE_SINGLE_PORT.
 *  03-03-08  02.00.07  Fixed MPI2_FW_IMAGE_HEADER by changing Reserved26 to
 *                      a U16 (from a U32).
 *                      Removed extra 's' from EventMasks name.
 *  06-27-08  02.00.08  Fixed an offset in a comment.
 *  10-02-08  02.00.09  Removed SystemReplyFrameSize from MPI2_IOC_INIT_REQUEST.
 *                      Removed CurReplyFrameSize from MPI2_IOC_FACTS_REPLY and
 *                      renamed MinReplyFrameSize to ReplyFrameSize.
 *                      Added MPI2_IOCFACTS_EXCEPT_IR_FOREIGN_CONFIG_MAX.
 *                      Added two new RAIDOperation values for Integrated RAID
 *                      Operations Status Event data.
 *                      Added four new IR Configuration Change List Event data
 *                      ReasonCode values.
 *                      Added two new ReasonCode defines for SAS Device Status
 *                      Change Event data.
 *                      Added three new DiscoveryStatus bits for the SAS
 *                      Discovery event data.
 *                      Added Multiplexing Status Change bit to the PhyStatus
 *                      field of the SAS Topology Change List event data.
 *                      Removed define for MPI2_INIT_IMAGE_BOOTFLAGS_XMEMCOPY.
 *                      BootFlags are now product-specific.
 *                      Added defines for the indivdual signature bytes
 *                      for MPI2_INIT_IMAGE_FOOTER.
 *  01-19-09  02.00.10  Added MPI2_IOCFACTS_CAPABILITY_EVENT_REPLAY define.
 *                      Added MPI2_EVENT_SAS_DISC_DS_DOWNSTREAM_INITIATOR
 *                      define.
 *                      Added MPI2_EVENT_SAS_DEV_STAT_RC_SATA_INIT_FAILURE
 *                      define.
 *                      Removed MPI2_EVENT_SAS_DISC_DS_SATA_INIT_FAILURE define.
 *  05-06-09  02.00.11  Added MPI2_IOCFACTS_CAPABILITY_RAID_ACCELERATOR define.
 *                      Added MPI2_IOCFACTS_CAPABILITY_MSI_X_INDEX define.
 *                      Added two new reason codes for SAS Device Status Change
 *                      Event.
 *                      Added new event: SAS PHY Counter.
 *  07-30-09  02.00.12  Added GPIO Interrupt event define and structure.
 *                      Added MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER define.
 *                      Added new product id family for 2208.
 *  10-28-09  02.00.13  Added HostMSIxVectors field to MPI2_IOC_INIT_REQUEST.
 *                      Added MaxMSIxVectors field to MPI2_IOC_FACTS_REPLY.
 *                      Added MinDevHandle field to MPI2_IOC_FACTS_REPLY.
 *                      Added MPI2_IOCFACTS_CAPABILITY_HOST_BASED_DISCOVERY.
 *                      Added MPI2_EVENT_HOST_BASED_DISCOVERY_PHY define.
 *                      Added MPI2_EVENT_SAS_TOPO_ES_NO_EXPANDER define.
 *                      Added Host Based Discovery Phy Event data.
 *                      Added defines for ProductID Product field
 *                      (MPI2_FW_HEADER_PID_).
 *                      Modified values for SAS ProductID Family
 *                      (MPI2_FW_HEADER_PID_FAMILY_).
 *  02-10-10  02.00.14  Added SAS Quiesce Event structure and defines.
 *                      Added PowerManagementControl Request structures and
 *                      defines.
 *  05-12-10  02.00.15  Marked Task Set Full Event as obsolete.
 *                      Added MPI2_EVENT_SAS_TOPO_LR_UNSUPPORTED_PHY define.
 *  11-10-10  02.00.16  Added MPI2_FW_DOWNLOAD_ITYPE_MIN_PRODUCT_SPECIFIC.
 *  02-23-11  02.00.17  Added SAS NOTIFY Primitive event, and added
 *                      SASNotifyPrimitiveMasks field to
 *                      MPI2_EVENT_NOTIFICATION_REQUEST.
 *                      Added Temperature Threshold Event.
 *                      Added Host Message Event.
 *                      Added Send Host Message request and reply.
 *  05-25-11  02.00.18  For Extended Image Header, added
 *                      MPI2_EXT_IMAGE_TYPE_MIN_PRODUCT_SPECIFIC and
 *                      MPI2_EXT_IMAGE_TYPE_MAX_PRODUCT_SPECIFIC defines.
 *                      Deprecated MPI2_EXT_IMAGE_TYPE_MAX define.
```



```
 *  08-24-11  02.00.19  Added PhysicalPort field to
 *                      MPI2_EVENT_DATA_SAS_DEVICE_STATUS_CHANGE structure.
 *                      Marked MPI2_PM_CONTROL_FEATURE_PCIE_LINK as obsolete.
 *  11-18-11  02.00.20  Incorporating additions for MPI v2.5.
 *  03-29-12  02.00.21  Added a product specific range to event values.
 *  07-26-12  02.00.22  Added MPI2_IOCFACTS_EXCEPT_PARTIAL_MEMORY_FAILURE.
 *                      Added ElapsedSeconds field to
 *                      MPI2_EVENT_DATA_IR_OPERATION_STATUS.
 *  08-19-13  02.00.23  For IOCInit, added MPI2_IOCINIT_MSGFLAG_RDPQ_ARRAY_MODE
 *                      and MPI2_IOC_INIT_RDPQ_ARRAY_ENTRY.
 *                      Added MPI2_IOCFACTS_CAPABILITY_RDPQ_ARRAY_CAPABLE.
 *                      Added MPI2_FW_DOWNLOAD_ITYPE_PUBLIC_KEY.
 *                      Added Encrypted Hash Extended Image.
 *  12-05-13  02.00.24  Added MPI25_HASH_IMAGE_TYPE_BIOS.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_IOC_H
#define MPI2_IOC_H

/*****************************************************************************
*
*               IOC Messages
*
*****************************************************************************/

/*****************************************************************************
*  IOCInit message
*****************************************************************************/

/* IOCInit Request message */
typedef struct _MPI2_IOC_INIT_REQUEST
{
    U8                      WhoInit;                        /* 0x00 */
    U8                      Reserved1;                      /* 0x01 */
    U8                      ChainOffset;                    /* 0x02 */
    U8                      Function;                       /* 0x03 */
    U16                     Reserved2;                      /* 0x04 */
    U8                      Reserved3;                      /* 0x06 */
    U8                      MsgFlags;                       /* 0x07 */
    U8                      VP_ID;                          /* 0x08 */
    U8                      VF_ID;                          /* 0x09 */
    U16                     Reserved4;                      /* 0x0A */
    U16                     MsgVersion;                     /* 0x0C */
    U16                     HeaderVersion;                  /* 0x0E */
    U32                     Reserved5;                      /* 0x10 */
    U16                     Reserved6;                      /* 0x14 */
    U8                      Reserved7;                      /* 0x16 */
    U8                      HostMSIxVectors;                /* 0x17 */
    U16                     Reserved8;                      /* 0x18 */
    U32                     Reserved6;                      /* 0x14 */
    U16                     Reserved7;                      /* 0x18 */
    U16                     SystemRequestFrameSize;         /* 0x1A */
    U16                     ReplyDescriptorPostQueueDepth;  /* 0x1C */
    U16                     ReplyFreeQueueDepth;            /* 0x1E */
    U32                     SenseBufferAddressHigh;         /* 0x20 */
    U32                     SystemReplyAddressHigh;         /* 0x24 */
    U64                     SystemRequestFrameBaseAddress;  /* 0x28 */
    U64                     ReplyDescriptorPostQueueAddress;/* 0x30 */
    U64                     ReplyFreeQueueAddress;          /* 0x38 */
    U64                     TimeStamp;                      /* 0x40 */
} MPI2_IOC_INIT_REQUEST, MPI2_POINTER PTR_MPI2_IOC_INIT_REQUEST,
  Mpi2IOCInitRequest_t, MPI2_POINTER pMpi2IOCInitRequest_t;

/* WhoInit values */
#define MPI2_WHOINIT_NOT_INITIALIZED            (0x00)
```

```
#define MPI2_WHOINIT_SYSTEM_BIOS                (0x01)
#define MPI2_WHOINIT_ROM_BIOS                   (0x02)
#define MPI2_WHOINIT_PCI_PEER                   (0x03)
#define MPI2_WHOINIT_HOST_DRIVER                (0x04)
#define MPI2_WHOINIT_MANUFACTURER               (0x05)

/* MsgFlags */
#define MPI2_IOCINIT_MSGFLAG_RDPQ_ARRAY_MODE    (0x01)

/* MsgVersion */
#define MPI2_IOCINIT_MSGVERSION_MAJOR_MASK      (0xFF00)
#define MPI2_IOCINIT_MSGVERSION_MAJOR_SHIFT     (8)
#define MPI2_IOCINIT_MSGVERSION_MINOR_MASK      (0x00FF)
#define MPI2_IOCINIT_MSGVERSION_MINOR_SHIFT     (0)

/* HeaderVersion */
#define MPI2_IOCINIT_HDRVERSION_UNIT_MASK       (0xFF00)
#define MPI2_IOCINIT_HDRVERSION_UNIT_SHIFT      (8)
#define MPI2_IOCINIT_HDRVERSION_DEV_MASK        (0x00FF)
#define MPI2_IOCINIT_HDRVERSION_DEV_SHIFT       (0)

/* minimum depth for a Reply Descriptor Post Queue */
/* minimum depth for the Reply Descriptor Post Queue */
#define MPI2_RDPQ_DEPTH_MIN                     (16)

/* Reply Descriptor Post Queue Array Entry */
typedef struct _MPI2_IOC_INIT_RDPQ_ARRAY_ENTRY
{
    U64                 RDPQBaseAddress;                /* 0x00 */
    U32                 Reserved1;                      /* 0x08 */
    U32                 Reserved2;                      /* 0x0C */
} MPI2_IOC_INIT_RDPQ_ARRAY_ENTRY,
  MPI2_POINTER PTR_MPI2_IOC_INIT_RDPQ_ARRAY_ENTRY,
  Mpi2IOCInitRDPQArrayEntry, MPI2_POINTER pMpi2IOCInitRDPQArrayEntry;

/* IOCInit Reply message */
typedef struct _MPI2_IOC_INIT_REPLY
{
    U8                  WhoInit;                        /* 0x00 */
    U8                  Reserved1;                      /* 0x01 */
    U8                  MsgLength;                      /* 0x02 */
    U8                  Function;                       /* 0x03 */
    U16                 Reserved2;                      /* 0x04 */
    U8                  Reserved3;                      /* 0x06 */
    U8                  MsgFlags;                       /* 0x07 */
    U8                  VP_ID;                          /* 0x08 */
    U8                  VF_ID;                          /* 0x09 */
    U16                 Reserved4;                      /* 0x0A */
    U16                 Reserved5;                      /* 0x0C */
    U16                 IOCStatus;                      /* 0x0E */
    U32                 IOCLogInfo;                     /* 0x10 */
} MPI2_IOC_INIT_REPLY, MPI2_POINTER PTR_MPI2_IOC_INIT_REPLY,
  unchanged_portion_omitted_
  Mpi2IOCFactsRequest_t, MPI2_POINTER pMpi2IOCFactsRequest_t;


/* IOCFacts Reply message */
typedef struct _MPI2_IOC_FACTS_REPLY
{
    U16                 MsgVersion;                     /* 0x00 */
    U8                  MsgLength;                      /* 0x02 */
    U8                  Function;                       /* 0x03 */
    U16                 HeaderVersion;                  /* 0x04 */
    U8                  IOCNumber;                      /* 0x06 */
    U8                  MsgFlags;                       /* 0x07 */
    U8                  VP_ID;                          /* 0x08 */
```



```
    U8                  VF_ID;                          /* 0x09 */
    U16                 Reserved1;                      /* 0x0A */
    U16                 IOCExceptions;                  /* 0x0C */
    U16                 IOCStatus;                      /* 0x0E */
    U32                 IOCLogInfo;                     /* 0x10 */
    U8                  MaxChainDepth;                  /* 0x14 */
    U8                  WhoInit;                        /* 0x15 */
    U8                  NumberOfPorts;                  /* 0x16 */
    U8                  MaxMSIxVectors;                 /* 0x17 */
    U8                  Reserved2;                      /* 0x17 */
    U16                 RequestCredit;                  /* 0x18 */
    U16                 ProductID;                      /* 0x1A */
    U32                 IOCCapabilities;                /* 0x1C */
    MPI2_VERSION_UNION  FWVersion;                      /* 0x20 */
    U16                 IOCRequestFrameSize;            /* 0x24 */
    U16                 IOCMaxChainSegmentSize;         /* 0x26 */ /* MPI 2.
    U16                 Reserved3;                      /* 0x26 */
    U16                 MaxInitiators;                  /* 0x28 */
    U16                 MaxTargets;                     /* 0x2A */
    U16                 MaxSasExpanders;                /* 0x2C */
    U16                 MaxEnclosures;                  /* 0x2E */
    U16                 ProtocolFlags;                  /* 0x30 */
    U16                 HighPriorityCredit;             /* 0x32 */
    U16                 MaxReplyDescriptorPostQueueDepth; /* 0x34 */
    U8                  ReplyFrameSize;                 /* 0x36 */
    U8                  MaxVolumes;                     /* 0x37 */
    U16                 MaxDevHandle;                   /* 0x38 */
    U16                 MaxPersistentEntries;           /* 0x3A */
    U16                 MinDevHandle;                   /* 0x3C */
    U16                 Reserved4;                      /* 0x3E */
    U32                 Reserved4;                      /* 0x3C */
} MPI2_IOC_FACTS_REPLY, MPI2_POINTER PTR_MPI2_IOC_FACTS_REPLY,
  Mpi2IOCFactsReply_t, MPI2_POINTER pMpi2IOCFactsReply_t;

/* MsgVersion */
#define MPI2_IOCFACTS_MSGVERSION_MAJOR_MASK             (0xFF00)
#define MPI2_IOCFACTS_MSGVERSION_MAJOR_SHIFT            (8)
#define MPI2_IOCFACTS_MSGVERSION_MINOR_MASK             (0x00FF)
#define MPI2_IOCFACTS_MSGVERSION_MINOR_SHIFT            (0)

/* HeaderVersion */
#define MPI2_IOCFACTS_HDRVERSION_UNIT_MASK              (0xFF00)
#define MPI2_IOCFACTS_HDRVERSION_UNIT_SHIFT             (8)
#define MPI2_IOCFACTS_HDRVERSION_DEV_MASK               (0x00FF)
#define MPI2_IOCFACTS_HDRVERSION_DEV_SHIFT              (0)

/* IOCExceptions */
#define MPI2_IOCFACTS_EXCEPT_PARTIAL_MEMORY_FAILURE     (0x0200)
#define MPI2_IOCFACTS_EXCEPT_IR_FOREIGN_CONFIG_MAX      (0x0100)

#define MPI2_IOCFACTS_EXCEPT_BOOTSTAT_MASK              (0x00E0)
#define MPI2_IOCFACTS_EXCEPT_BOOTSTAT_GOOD              (0x0000)
#define MPI2_IOCFACTS_EXCEPT_BOOTSTAT_BACKUP            (0x0020)
#define MPI2_IOCFACTS_EXCEPT_BOOTSTAT_RESTORED          (0x0040)
#define MPI2_IOCFACTS_EXCEPT_BOOTSTAT_CORRUPT_BACKUP    (0x0060)

#define MPI2_IOCFACTS_EXCEPT_METADATA_UNSUPPORTED       (0x0010)
#define MPI2_IOCFACTS_EXCEPT_MANUFACT_CHECKSUM_FAIL     (0x0008)
#define MPI2_IOCFACTS_EXCEPT_FW_CHECKSUM_FAIL           (0x0004)
#define MPI2_IOCFACTS_EXCEPT_RAID_CONFIG_INVALID        (0x0002)
#define MPI2_IOCFACTS_EXCEPT_CONFIG_CHECKSUM_FAIL       (0x0001)

/* defines for WhoInit field are after the IOCInit Request */

/* ProductID field uses MPI2_FW_HEADER_PID_ */
```

```
 362 /* IOCCapabilities */
 363 #define MPI2_IOCFACTS_CAPABILITY_RDPQ_ARRAY_CAPABLE     (0x00040000)
 364 #define MPI25_IOCFACTS_CAPABILITY_FAST_PATH_CAPABLE     (0x00020000)
 365 #define MPI2_IOCFACTS_CAPABILITY_HOST_BASED_DISCOVERY   (0x00010000)
 366 #define MPI2_IOCFACTS_CAPABILITY_MSI_X_INDEX            (0x00008000)
 367 #define MPI2_IOCFACTS_CAPABILITY_RAID_ACCELERATOR       (0x00004000)
 368 #define MPI2_IOCFACTS_CAPABILITY_EVENT_REPLAY           (0x00002000)
 369 #define MPI2_IOCFACTS_CAPABILITY_INTEGRATED_RAID        (0x00001000)
 370 #define MPI2_IOCFACTS_CAPABILITY_TLR                    (0x00000800)
 371 #define MPI2_IOCFACTS_CAPABILITY_MULTICAST              (0x00000100)
 372 #define MPI2_IOCFACTS_CAPABILITY_BIDIRECTIONAL_TARGET   (0x00000080)
 373 #define MPI2_IOCFACTS_CAPABILITY_EEDP                   (0x00000040)
 374 #define MPI2_IOCFACTS_CAPABILITY_EXTENDED_BUFFER        (0x00000020)
 375 #define MPI2_IOCFACTS_CAPABILITY_SNAPSHOT_BUFFER        (0x00000010)
 376 #define MPI2_IOCFACTS_CAPABILITY_DIAG_TRACE_BUFFER      (0x00000008)
 377 #define MPI2_IOCFACTS_CAPABILITY_TASK_SET_FULL_HANDLING (0x00000004)

 379 /* ProtocolFlags */
 380 #define MPI2_IOCFACTS_PROTOCOL_SCSI_TARGET              (0x0001)
 381 #define MPI2_IOCFACTS_PROTOCOL_SCSI_INITIATOR           (0x0002)


 384 /****************************************************************************
 385 *  PortFacts message
 386 ****************************************************************************/

 388 /* PortFacts Request message */
 389 typedef struct _MPI2_PORT_FACTS_REQUEST
 390 {
 391     U16                     Reserved1;                  /* 0x00 */
 392     U8                      ChainOffset;                /* 0x02 */
 393     U8                      Function;                   /* 0x03 */
 394     U16                     Reserved2;                  /* 0x04 */
 395     U8                      PortNumber;                 /* 0x06 */
 396     U8                      MsgFlags;                   /* 0x07 */
 397     U8                      VP_ID;                      /* 0x08 */
 398     U8                      VF_ID;                      /* 0x09 */
 399     U16                     Reserved3;                  /* 0x0A */
 400 } MPI2_PORT_FACTS_REQUEST, MPI2_POINTER PTR_MPI2_PORT_FACTS_REQUEST,
_____unchanged_portion_omitted_
 472   Mpi2PortEnableReply_t, MPI2_POINTER pMpi2PortEnableReply_t;


 475 /****************************************************************************
 476 *  EventNotification message
 477 ****************************************************************************/

 479 /* EventNotification Request message */
 480 #define MPI2_EVENT_NOTIFY_EVENTMASK_WORDS           (4)

 482 typedef struct _MPI2_EVENT_NOTIFICATION_REQUEST
 483 {
 484     U16                     Reserved1;                  /* 0x00 */
 485     U8                      ChainOffset;                /* 0x02 */
 486     U8                      Function;                   /* 0x03 */
 487     U16                     Reserved2;                  /* 0x04 */
 488     U8                      Reserved3;                  /* 0x06 */
 489     U8                      MsgFlags;                   /* 0x07 */
 490     U8                      VP_ID;                      /* 0x08 */
 491     U8                      VF_ID;                      /* 0x09 */
 492     U16                     Reserved4;                  /* 0x0A */
 493     U32                     Reserved5;                  /* 0x0C */
 494     U32                     Reserved6;                  /* 0x10 */
 495     U32                     EventMasks[MPI2_EVENT_NOTIFY_EVENTMASK_WORDS];/* 0x1
 496     U16                     SASBroadcastPrimitiveMasks; /* 0x24 */
 497     U16                     SASNotifyPrimitiveMasks;    /* 0x26 */
```



```
 448     U16                     Reserved7;                  /* 0x26 */
 498     U32                     Reserved8;                  /* 0x28 */
 499 } MPI2_EVENT_NOTIFICATION_REQUEST,
_____unchanged_portion_omitted_
 524   Mpi2EventNotificationReply_t, MPI2_POINTER pMpi2EventNotificationReply_t;

 526 /* AckRequired */
 527 #define MPI2_EVENT_NOTIFICATION_ACK_NOT_REQUIRED    (0x00)
 528 #define MPI2_EVENT_NOTIFICATION_ACK_REQUIRED        (0x01)

 530 /* Event */
 531 #define MPI2_EVENT_LOG_DATA                         (0x0001)
 532 #define MPI2_EVENT_STATE_CHANGE                     (0x0002)
 533 #define MPI2_EVENT_HARD_RESET_RECEIVED              (0x0005)
 534 #define MPI2_EVENT_EVENT_CHANGE                     (0x000A)
 535 #define MPI2_EVENT_TASK_SET_FULL                    (0x000E) /* obsolete */
 486 #define MPI2_EVENT_TASK_SET_FULL                    (0x000E)
 536 #define MPI2_EVENT_SAS_DEVICE_STATUS_CHANGE         (0x000F)
 537 #define MPI2_EVENT_IR_OPERATION_STATUS              (0x0014)
 538 #define MPI2_EVENT_SAS_DISCOVERY                    (0x0016)
 539 #define MPI2_EVENT_SAS_BROADCAST_PRIMITIVE          (0x0017)
 540 #define MPI2_EVENT_SAS_INIT_DEVICE_STATUS_CHANGE    (0x0018)
 541 #define MPI2_EVENT_SAS_INIT_TABLE_OVERFLOW          (0x0019)
 542 #define MPI2_EVENT_SAS_TOPOLOGY_CHANGE_LIST         (0x001C)
 543 #define MPI2_EVENT_SAS_ENCL_DEVICE_STATUS_CHANGE    (0x001D)
 544 #define MPI2_EVENT_IR_VOLUME                        (0x001E)
 545 #define MPI2_EVENT_IR_PHYSICAL_DISK                 (0x001F)
 546 #define MPI2_EVENT_IR_CONFIGURATION_CHANGE_LIST     (0x0020)
 547 #define MPI2_EVENT_LOG_ENTRY_ADDED                  (0x0021)
 548 #define MPI2_EVENT_SAS_PHY_COUNTER                  (0x0022)
 549 #define MPI2_EVENT_GPIO_INTERRUPT                   (0x0023)
 550 #define MPI2_EVENT_HOST_BASED_DISCOVERY_PHY         (0x0024)
 551 #define MPI2_EVENT_SAS_QUIESCE                      (0x0025)
 552 #define MPI2_EVENT_SAS_NOTIFY_PRIMITIVE             (0x0026)
 553 #define MPI2_EVENT_TEMP_THRESHOLD                   (0x0027)
 554 #define MPI2_EVENT_HOST_MESSAGE                     (0x0028)
 555 #define MPI2_EVENT_POWER_PERFORMANCE_CHANGE         (0x0029)
 556 #define MPI2_EVENT_MIN_PRODUCT_SPECIFIC             (0x006E)
 557 #define MPI2_EVENT_MAX_PRODUCT_SPECIFIC             (0x007F)


 560 /* Log Entry Added Event data */

 562 /* the following structure matches MPI2_LOG_0_ENTRY in mpi2_cnfg.h */
 563 #define MPI2_EVENT_DATA_LOG_DATA_LENGTH             (0x1C)

 565 typedef struct _MPI2_EVENT_DATA_LOG_ENTRY_ADDED
 566 {
 567     U64         TimeStamp;                          /* 0x00 */
 568     U32         Reserved1;                          /* 0x08 */
 569     U16         LogSequence;                        /* 0x0C */
 570     U16         LogEntryQualifier;                  /* 0x0E */
 571     U8          VP_ID;                              /* 0x10 */
 572     U8          VF_ID;                              /* 0x11 */
 573     U16         Reserved2;                          /* 0x12 */
 574     U8          LogData[MPI2_EVENT_DATA_LOG_DATA_LENGTH];/* 0x14 */
 575 } MPI2_EVENT_DATA_LOG_ENTRY_ADDED,
 576   MPI2_POINTER PTR_MPI2_EVENT_DATA_LOG_ENTRY_ADDED,
 577   Mpi2EventDataLogEntryAdded_t, MPI2_POINTER pMpi2EventDataLogEntryAdded_t;


 580 /* GPIO Interrupt Event data */

 582 typedef struct _MPI2_EVENT_DATA_GPIO_INTERRUPT
 583 {
 584     U8          GPIONum;                            /* 0x00 */
```

```
585     U8          Reserved1;                          /* 0x01 */
586     U16         Reserved2;                          /* 0x02 */
587 } MPI2_EVENT_DATA_GPIO_INTERRUPT,
588   MPI2_POINTER PTR_MPI2_EVENT_DATA_GPIO_INTERRUPT,
589   Mpi2EventDataGpioInterrupt_t, MPI2_POINTER pMpi2EventDataGpioInterrupt_t;

592 /* Temperature Threshold Event data */

594 typedef struct _MPI2_EVENT_DATA_TEMPERATURE
595 {
596     U16         Status;                             /* 0x00 */
597     U8          SensorNum;                          /* 0x02 */
598     U8          Reserved1;                          /* 0x03 */
599     U16         CurrentTemperature;                 /* 0x04 */
600     U16         Reserved2;                          /* 0x06 */
601     U32         Reserved3;                          /* 0x08 */
602     U32         Reserved4;                          /* 0x0C */
603 } MPI2_EVENT_DATA_TEMPERATURE,
604   MPI2_POINTER PTR_MPI2_EVENT_DATA_TEMPERATURE,
605   Mpi2EventDataTemperature_t, MPI2_POINTER pMpi2EventDataTemperature_t;

607 /* Temperature Threshold Event data Status bits */
608 #define MPI2_EVENT_TEMPERATURE3_EXCEEDED            (0x0008)
609 #define MPI2_EVENT_TEMPERATURE2_EXCEEDED            (0x0004)
610 #define MPI2_EVENT_TEMPERATURE1_EXCEEDED            (0x0002)
611 #define MPI2_EVENT_TEMPERATURE0_EXCEEDED            (0x0001)


614 /* Host Message Event data */

616 typedef struct _MPI2_EVENT_DATA_HOST_MESSAGE
617 {
618     U8          SourceVF_ID;                        /* 0x00 */
619     U8          Reserved1;                          /* 0x01 */
620     U16         Reserved2;                          /* 0x02 */
621     U32         Reserved3;                          /* 0x04 */
622     U32         HostData[1];                        /* 0x08 */
623 } MPI2_EVENT_DATA_HOST_MESSAGE, MPI2_POINTER PTR_MPI2_EVENT_DATA_HOST_MESSAGE,
624   Mpi2EventDataHostMessage_t, MPI2_POINTER pMpi2EventDataHostMessage_t;


627 /* Power Performance Change Event */

629 typedef struct _MPI2_EVENT_DATA_POWER_PERF_CHANGE
630 {
631     U8          CurrentPowerMode;                   /* 0x00 */
632     U8          PreviousPowerMode;                  /* 0x01 */
633     U16         Reserved1;                          /* 0x02 */
634 } MPI2_EVENT_DATA_POWER_PERF_CHANGE,
635   MPI2_POINTER PTR_MPI2_EVENT_DATA_POWER_PERF_CHANGE,
636   Mpi2EventDataPowerPerfChange_t, MPI2_POINTER pMpi2EventDataPowerPerfChange_t;

638 /* defines for CurrentPowerMode and PreviousPowerMode fields */
639 #define MPI2_EVENT_PM_INIT_MASK             (0xC0)
640 #define MPI2_EVENT_PM_INIT_UNAVAILABLE      (0x00)
641 #define MPI2_EVENT_PM_INIT_HOST             (0x40)
642 #define MPI2_EVENT_PM_INIT_IO_UNIT          (0x80)
643 #define MPI2_EVENT_PM_INIT_PCIE_DPA         (0xC0)

645 #define MPI2_EVENT_PM_MODE_MASK             (0x07)
646 #define MPI2_EVENT_PM_MODE_UNAVAILABLE      (0x00)
647 #define MPI2_EVENT_PM_MODE_UNKNOWN          (0x01)
648 #define MPI2_EVENT_PM_MODE_FULL_POWER       (0x04)
649 #define MPI2_EVENT_PM_MODE_REDUCED_POWER    (0x05)
650 #define MPI2_EVENT_PM_MODE_STANDBY          (0x06)
```



```
653 /* Hard Reset Received Event data */

655 typedef struct _MPI2_EVENT_DATA_HARD_RESET_RECEIVED
656 {
657     U8                      Reserved1;              /* 0x00 */
658     U8                      Port;                   /* 0x01 */
659     U16                     Reserved2;              /* 0x02 */
660 } MPI2_EVENT_DATA_HARD_RESET_RECEIVED,
661   MPI2_POINTER PTR_MPI2_EVENT_DATA_HARD_RESET_RECEIVED,
662   Mpi2EventDataHardResetReceived_t,
663   MPI2_POINTER pMpi2EventDataHardResetReceived_t;


666 /* Task Set Full Event data */
667 /*   this event is obsolete */

669 typedef struct _MPI2_EVENT_DATA_TASK_SET_FULL
670 {
671     U16                     DevHandle;              /* 0x00 */
672     U16                     CurrentDepth;           /* 0x02 */
673 } MPI2_EVENT_DATA_TASK_SET_FULL, MPI2_POINTER PTR_MPI2_EVENT_DATA_TASK_SET_FULL,
674   Mpi2EventDataTaskSetFull_t, MPI2_POINTER pMpi2EventDataTaskSetFull_t;


677 /* SAS Device Status Change Event data */

679 typedef struct _MPI2_EVENT_DATA_SAS_DEVICE_STATUS_CHANGE
680 {
681     U16                     TaskTag;                /* 0x00 */
682     U8                      ReasonCode;             /* 0x02 */
683     U8                      PhysicalPort;           /* 0x03 */
561     U8                      Reserved1;              /* 0x03 */
684     U8                      ASC;                    /* 0x04 */
685     U8                      ASCQ;                   /* 0x05 */
686     U16                     DevHandle;              /* 0x06 */
687     U32                     Reserved2;              /* 0x08 */
688     U64                     SASAddress;             /* 0x0C */
689     U8                      LUN[8];                 /* 0x14 */
690 } MPI2_EVENT_DATA_SAS_DEVICE_STATUS_CHANGE,
691   MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_DEVICE_STATUS_CHANGE,
692   Mpi2EventDataSasDeviceStatusChange_t,
693   MPI2_POINTER pMpi2EventDataSasDeviceStatusChange_t;

695 /* SAS Device Status Change Event data ReasonCode values */
696 #define MPI2_EVENT_SAS_DEV_STAT_RC_SMART_DATA                           (0x05)
697 #define MPI2_EVENT_SAS_DEV_STAT_RC_UNSUPPORTED                          (0x07)
698 #define MPI2_EVENT_SAS_DEV_STAT_RC_INTERNAL_DEVICE_RESET                (0x08)
699 #define MPI2_EVENT_SAS_DEV_STAT_RC_TASK_ABORT_INTERNAL                  (0x09)
700 #define MPI2_EVENT_SAS_DEV_STAT_RC_ABORT_TASK_SET_INTERNAL              (0x0A)
701 #define MPI2_EVENT_SAS_DEV_STAT_RC_CLEAR_TASK_SET_INTERNAL              (0x0B)
702 #define MPI2_EVENT_SAS_DEV_STAT_RC_QUERY_TASK_INTERNAL                  (0x0C)
703 #define MPI2_EVENT_SAS_DEV_STAT_RC_ASYNC_NOTIFICATION                   (0x0D)
704 #define MPI2_EVENT_SAS_DEV_STAT_RC_CMP_INTERNAL_DEV_RESET               (0x0E)
705 #define MPI2_EVENT_SAS_DEV_STAT_RC_CMP_TASK_ABORT_INTERNAL              (0x0F)
706 #define MPI2_EVENT_SAS_DEV_STAT_RC_SATA_INIT_FAILURE                    (0x10)
707 #define MPI2_EVENT_SAS_DEV_STAT_RC_EXPANDER_REDUCED_FUNCTIONALITY       (0x11)
708 #define MPI2_EVENT_SAS_DEV_STAT_RC_CMP_EXPANDER_REDUCED_FUNCTIONALITY   (0x12)


711 /* Integrated RAID Operation Status Event data */

713 typedef struct _MPI2_EVENT_DATA_IR_OPERATION_STATUS
714 {
715     U16                     VolDevHandle;           /* 0x00 */
```

```
    U16                     Reserved1;                  /* 0x02 */
    U8                      RAIDOperation;              /* 0x04 */
    U8                      PercentComplete;            /* 0x05 */
    U16                     Reserved2;                  /* 0x06 */
    U32                     ElapsedSeconds;             /* 0x08 */
    U32                     Resereved3;                 /* 0x08 */
} MPI2_EVENT_DATA_IR_OPERATION_STATUS,
      unchanged_portion_omitted
  MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_BROADCAST_PRIMITIVE,
  Mpi2EventDataSasBroadcastPrimitive_t,
  MPI2_POINTER pMpi2EventDataSasBroadcastPrimitive_t;

/* defines for the Primitive field */
#define MPI2_EVENT_PRIMITIVE_CHANGE                         (0x01)
#define MPI2_EVENT_PRIMITIVE_SES                            (0x02)
#define MPI2_EVENT_PRIMITIVE_EXPANDER                       (0x03)
#define MPI2_EVENT_PRIMITIVE_ASYNCHRONOUS_EVENT             (0x04)
#define MPI2_EVENT_PRIMITIVE_RESERVED3                      (0x05)
#define MPI2_EVENT_PRIMITIVE_RESERVED4                      (0x06)
#define MPI2_EVENT_PRIMITIVE_CHANGE0_RESERVED               (0x07)
#define MPI2_EVENT_PRIMITIVE_CHANGE1_RESERVED               (0x08)


/* SAS Notify Primitive Event data */

typedef struct _MPI2_EVENT_DATA_SAS_NOTIFY_PRIMITIVE
{
    U8                      PhyNum;                 /* 0x00 */
    U8                      Port;                   /* 0x01 */
    U8                      Reserved1;              /* 0x02 */
    U8                      Primitive;              /* 0x03 */
} MPI2_EVENT_DATA_SAS_NOTIFY_PRIMITIVE,
  MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_NOTIFY_PRIMITIVE,
  Mpi2EventDataSasNotifyPrimitive_t,
  MPI2_POINTER pMpi2EventDataSasNotifyPrimitive_t;

/* defines for the Primitive field */
#define MPI2_EVENT_NOTIFY_ENABLE_SPINUP                     (0x01)
#define MPI2_EVENT_NOTIFY_POWER_LOSS_EXPECTED               (0x02)
#define MPI2_EVENT_NOTIFY_RESERVED1                         (0x03)
#define MPI2_EVENT_NOTIFY_RESERVED2                         (0x04)


/* SAS Initiator Device Status Change Event data */

typedef struct _MPI2_EVENT_DATA_SAS_INIT_DEV_STATUS_CHANGE
{
    U8                      ReasonCode;             /* 0x00 */
    U8                      PhysicalPort;           /* 0x01 */
    U16                     DevHandle;              /* 0x02 */
    U64                     SASAddress;             /* 0x04 */
} MPI2_EVENT_DATA_SAS_INIT_DEV_STATUS_CHANGE,
      unchanged_portion_omitted
  MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_TOPOLOGY_CHANGE_LIST,
  Mpi2EventDataSasTopologyChangeList_t,
  MPI2_POINTER pMpi2EventDataSasTopologyChangeList_t;

/* values for the ExpStatus field */
#define MPI2_EVENT_SAS_TOPO_ES_NO_EXPANDER                  (0x00)
#define MPI2_EVENT_SAS_TOPO_ES_ADDED                        (0x01)
#define MPI2_EVENT_SAS_TOPO_ES_NOT_RESPONDING               (0x02)
#define MPI2_EVENT_SAS_TOPO_ES_RESPONDING                   (0x03)
#define MPI2_EVENT_SAS_TOPO_ES_DELAY_NOT_RESPONDING         (0x04)

/* defines for the LinkRate field */
#define MPI2_EVENT_SAS_TOPO_LR_CURRENT_MASK                 (0xF0)
```



```
#define MPI2_EVENT_SAS_TOPO_LR_CURRENT_SHIFT                (4)
#define MPI2_EVENT_SAS_TOPO_LR_PREV_MASK                    (0x0F)
#define MPI2_EVENT_SAS_TOPO_LR_PREV_SHIFT                   (0)

#define MPI2_EVENT_SAS_TOPO_LR_UNKNOWN_LINK_RATE            (0x00)
#define MPI2_EVENT_SAS_TOPO_LR_PHY_DISABLED                 (0x01)
#define MPI2_EVENT_SAS_TOPO_LR_NEGOTIATION_FAILED           (0x02)
#define MPI2_EVENT_SAS_TOPO_LR_SATA_OOB_COMPLETE            (0x03)
#define MPI2_EVENT_SAS_TOPO_LR_PORT_SELECTOR                (0x04)
#define MPI2_EVENT_SAS_TOPO_LR_SMP_RESET_IN_PROGRESS        (0x05)
#define MPI2_EVENT_SAS_TOPO_LR_UNSUPPORTED_PHY              (0x06)
#define MPI2_EVENT_SAS_TOPO_LR_RATE_1_5                     (0x08)
#define MPI2_EVENT_SAS_TOPO_LR_RATE_3_0                     (0x09)
#define MPI2_EVENT_SAS_TOPO_LR_RATE_6_0                     (0x0A)
#define MPI25_EVENT_SAS_TOPO_LR_RATE_12_0                   (0x0B)

/* values for the PhyStatus field */
#define MPI2_EVENT_SAS_TOPO_PHYSTATUS_VACANT                (0x80)
#define MPI2_EVENT_SAS_TOPO_PS_MULTIPLEX_CHANGE             (0x10)
/* values for the PhyStatus ReasonCode sub-field */
#define MPI2_EVENT_SAS_TOPO_RC_MASK                         (0x0F)
#define MPI2_EVENT_SAS_TOPO_RC_TARG_ADDED                   (0x01)
#define MPI2_EVENT_SAS_TOPO_RC_TARG_NOT_RESPONDING          (0x02)
#define MPI2_EVENT_SAS_TOPO_RC_PHY_CHANGED                  (0x03)
#define MPI2_EVENT_SAS_TOPO_RC_NO_CHANGE                    (0x04)
#define MPI2_EVENT_SAS_TOPO_RC_DELAY_NOT_RESPONDING         (0x05)


/* SAS Enclosure Device Status Change Event data */

typedef struct _MPI2_EVENT_DATA_SAS_ENCL_DEV_STATUS_CHANGE
{
    U16                     EnclosureHandle;        /* 0x00 */
    U8                      ReasonCode;             /* 0x02 */
    U8                      PhysicalPort;           /* 0x03 */
    U64                     EnclosureLogicalID;     /* 0x04 */
    U16                     NumSlots;               /* 0x0C */
    U16                     StartSlot;              /* 0x0E */
    U32                     PhyBits;                /* 0x10 */
} MPI2_EVENT_DATA_SAS_ENCL_DEV_STATUS_CHANGE,
      unchanged_portion_omitted
  MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_PHY_COUNTER,
  Mpi2EventDataSasPhyCounter_t, MPI2_POINTER pMpi2EventDataSasPhyCounter_t;

/* use MPI2_SASPHY3_EVENT_CODE_ values from mpi2_cnfg.h for the PhyEventCode fie

/* use MPI2_SASPHY3_COUNTER_TYPE_ values from mpi2_cnfg.h for the CounterType fi

/* use MPI2_SASPHY3_TIME_UNITS_ values from mpi2_cnfg.h for the TimeUnits field

/* use MPI2_SASPHY3_TFLAGS_ values from mpi2_cnfg.h for the ThresholdFlags field


/* SAS Quiesce Event data */

typedef struct _MPI2_EVENT_DATA_SAS_QUIESCE
{
    U8                      ReasonCode;             /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U16                     Reserved2;              /* 0x02 */
    U32                     Reserved3;              /* 0x04 */
} MPI2_EVENT_DATA_SAS_QUIESCE,
  MPI2_POINTER PTR_MPI2_EVENT_DATA_SAS_QUIESCE,
  Mpi2EventDataSasQuiesce_t, MPI2_POINTER pMpi2EventDataSasQuiesce_t;

/* SAS Quiesce Event data ReasonCode values */
```

```
#define MPI2_EVENT_SAS_QUIESCE_RC_STARTED                   (0x01)
#define MPI2_EVENT_SAS_QUIESCE_RC_COMPLETED                 (0x02)


/* Host Based Discovery Phy Event data */

typedef struct _MPI2_EVENT_HBD_PHY_SAS
{
    U8          Flags;                      /* 0x00 */
    U8          NegotiatedLinkRate;         /* 0x01 */
    U8          PhyNum;                     /* 0x02 */
    U8          PhysicalPort;               /* 0x03 */
    U32         Reserved1;                  /* 0x04 */
    U8          InitialFrame[28];           /* 0x08 */
} MPI2_EVENT_HBD_PHY_SAS, MPI2_POINTER PTR_MPI2_EVENT_HBD_PHY_SAS,
  Mpi2EventHbdPhySas_t, MPI2_POINTER pMpi2EventHbdPhySas_t;

/* values for the Flags field */
#define MPI2_EVENT_HBD_SAS_FLAGS_FRAME_VALID        (0x02)
#define MPI2_EVENT_HBD_SAS_FLAGS_SATA_FRAME         (0x01)

/* use MPI2_SAS_NEG_LINK_RATE_ defines from mpi2_cnfg.h for the NegotiatedLinkRa

typedef union _MPI2_EVENT_HBD_DESCRIPTOR
{
    MPI2_EVENT_HBD_PHY_SAS      Sas;
} MPI2_EVENT_HBD_DESCRIPTOR, MPI2_POINTER PTR_MPI2_EVENT_HBD_DESCRIPTOR,
  Mpi2EventHbdDescriptor_t, MPI2_POINTER pMpi2EventHbdDescriptor_t;

typedef struct _MPI2_EVENT_DATA_HBD_PHY
{
    U8                          DescriptorType;     /* 0x00 */
    U8                          Reserved1;          /* 0x01 */
    U16                         Reserved2;          /* 0x02 */
    U32                         Reserved3;          /* 0x04 */
    MPI2_EVENT_HBD_DESCRIPTOR   Descriptor;         /* 0x08 */
} MPI2_EVENT_DATA_HBD_PHY, MPI2_POINTER PTR_MPI2_EVENT_DATA_HBD_PHY,
  Mpi2EventDataHbdPhy_t, MPI2_POINTER pMpi2EventDataMpi2EventDataHbdPhy_t;

/* values for the DescriptorType field */
#define MPI2_EVENT_HBD_DT_SAS               (0x01)



/*****************************************************************************
*  EventAck message
*****************************************************************************/

/* EventAck Request message */
typedef struct _MPI2_EVENT_ACK_REQUEST
{
    U16                     Reserved1;                  /* 0x00 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved4;                  /* 0x0A */
    U16                     Event;                      /* 0x0C */
    U16                     Reserved5;                  /* 0x0E */
    U32                     EventContext;               /* 0x10 */
} MPI2_EVENT_ACK_REQUEST, MPI2_POINTER PTR_MPI2_EVENT_ACK_REQUEST,
______unchanged_portion_omitted_
  Mpi2EventAckReply_t, MPI2_POINTER pMpi2EventAckReply_t;
```



```
/*****************************************************************************
*  SendHostMessage message
*****************************************************************************/

/* SendHostMessage Request message */
typedef struct _MPI2_SEND_HOST_MESSAGE_REQUEST
{
    U16                     HostDataLength;             /* 0x00 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved1;                  /* 0x04 */
    U8                      Reserved2;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved3;                  /* 0x0A */
    U8                      Reserved4;                  /* 0x0C */
    U8                      DestVF_ID;                  /* 0x0D */
    U16                     Reserved5;                  /* 0x0E */
    U32                     Reserved6;                  /* 0x10 */
    U32                     Reserved7;                  /* 0x14 */
    U32                     Reserved8;                  /* 0x18 */
    U32                     Reserved9;                  /* 0x1C */
    U32                     Reserved10;                 /* 0x20 */
    U32                     HostData[1];                /* 0x24 */
} MPI2_SEND_HOST_MESSAGE_REQUEST,
  MPI2_POINTER PTR_MPI2_SEND_HOST_MESSAGE_REQUEST,
  Mpi2SendHostMessageRequest_t, MPI2_POINTER pMpi2SendHostMessageRequest_t;


/* SendHostMessage Reply message */
typedef struct _MPI2_SEND_HOST_MESSAGE_REPLY
{
    U16                     HostDataLength;             /* 0x00 */
    U8                      MsgLength;                  /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved1;                  /* 0x04 */
    U8                      Reserved2;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved3;                  /* 0x0A */
    U16                     Reserved4;                  /* 0x0C */
    U16                     IOCStatus;                  /* 0x0E */
    U32                     IOCLogInfo;                 /* 0x10 */
} MPI2_SEND_HOST_MESSAGE_REPLY, MPI2_POINTER PTR_MPI2_SEND_HOST_MESSAGE_REPLY,
  Mpi2SendHostMessageReply_t, MPI2_POINTER pMpi2SendHostMessageReply_t;


/*****************************************************************************
*  FWDownload message
*****************************************************************************/

/* MPI v2.0 FWDownload Request message */
/* FWDownload Request message */
typedef struct _MPI2_FW_DOWNLOAD_REQUEST
{
    U8                      ImageType;                  /* 0x00 */
    U8                      Reserved1;                  /* 0x01 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
```

```
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U32                     TotalImageSize;         /* 0x0C */
    U32                     Reserved5;              /* 0x10 */
    MPI2_MPI_SGE_UNION      SGL;                    /* 0x14 */
} MPI2_FW_DOWNLOAD_REQUEST, MPI2_POINTER PTR_MPI2_FW_DOWNLOAD_REQUEST,
  Mpi2FWDownloadRequest, MPI2_POINTER pMpi2FWDownloadRequest;

#define MPI2_FW_DOWNLOAD_MSGFLGS_LAST_SEGMENT   (0x01)

#define MPI2_FW_DOWNLOAD_ITYPE_FW                   (0x01)
#define MPI2_FW_DOWNLOAD_ITYPE_BIOS                 (0x02)
#define MPI2_FW_DOWNLOAD_ITYPE_MANUFACTURING        (0x06)
#define MPI2_FW_DOWNLOAD_ITYPE_CONFIG_1             (0x07)
#define MPI2_FW_DOWNLOAD_ITYPE_CONFIG_2             (0x08)
#define MPI2_FW_DOWNLOAD_ITYPE_MEGARAID             (0x09)
#define MPI2_FW_DOWNLOAD_ITYPE_COMPLETE             (0x0A)
#define MPI2_FW_DOWNLOAD_ITYPE_COMMON_BOOT_BLOCK    (0x0B)
#define MPI2_FW_DOWNLOAD_ITYPE_PUBLIC_KEY           (0x0C) /* MPI v2.5 and newer
#define MPI2_FW_DOWNLOAD_ITYPE_MIN_PRODUCT_SPECIFIC (0xF0)

/* MPI v2.0 FWDownload TransactionContext Element */
/* FWDownload TransactionContext Element */
typedef struct _MPI2_FW_DOWNLOAD_TCSGE
{
    U8                      Reserved1;              /* 0x00 */
    U8                      ContextSize;            /* 0x01 */
    U8                      DetailsLength;          /* 0x02 */
    U8                      Flags;                  /* 0x03 */
    U32                     Reserved2;              /* 0x04 */
    U32                     ImageOffset;            /* 0x08 */
    U32                     ImageSize;              /* 0x0C */
} MPI2_FW_DOWNLOAD_TCSGE, MPI2_POINTER PTR_MPI2_FW_DOWNLOAD_TCSGE,
  Mpi2FWDownloadTCSGE_t, MPI2_POINTER pMpi2FWDownloadTCSGE_t;


/* MPI v2.5 FWDownload Request message */
typedef struct _MPI25_FW_DOWNLOAD_REQUEST
{
    U8                      ImageType;              /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved2;              /* 0x04 */
    U8                      Reserved3;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U32                     TotalImageSize;         /* 0x0C */
    U32                     Reserved5;              /* 0x10 */
    U32                     Reserved6;              /* 0x14 */
    U32                     ImageOffset;            /* 0x18 */
    U32                     ImageSize;              /* 0x1C */
    MPI25_SGE_IO_UNION      SGL;                    /* 0x20 */
} MPI25_FW_DOWNLOAD_REQUEST, MPI2_POINTER PTR_MPI25_FW_DOWNLOAD_REQUEST,
  Mpi25FWDownloadRequest, MPI2_POINTER pMpi25FWDownloadRequest;


/* FWDownload Reply message */
typedef struct _MPI2_FW_DOWNLOAD_REPLY
{
    U8                      ImageType;              /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      MsgLength;              /* 0x02 */
```



```
    U8                      Function;               /* 0x03 */
    U16                     Reserved2;              /* 0x04 */
    U8                      Reserved3;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U16                     Reserved5;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
} MPI2_FW_DOWNLOAD_REPLY, MPI2_POINTER PTR_MPI2_FW_DOWNLOAD_REPLY,
  Mpi2FWDownloadReply_t, MPI2_POINTER pMpi2FWDownloadReply_t;


/*****************************************************************************
*  FWUpload message
*****************************************************************************/

/* MPI v2.0 FWUpload Request message */
/* FWUpload Request message */
typedef struct _MPI2_FW_UPLOAD_REQUEST
{
    U8                      ImageType;              /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved2;              /* 0x04 */
    U8                      Reserved3;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U32                     Reserved5;              /* 0x0C */
    U32                     Reserved6;              /* 0x10 */
    MPI2_MPI_SGE_UNION      SGL;                    /* 0x14 */
} MPI2_FW_UPLOAD_REQUEST, MPI2_POINTER PTR_MPI2_FW_UPLOAD_REQUEST,
  Mpi2FWUploadRequest_t, MPI2_POINTER pMpi2FWUploadRequest_t;

#define MPI2_FW_UPLOAD_ITYPE_FW_CURRENT         (0x00)
#define MPI2_FW_UPLOAD_ITYPE_FW_FLASH           (0x01)
#define MPI2_FW_UPLOAD_ITYPE_BIOS_FLASH         (0x02)
#define MPI2_FW_UPLOAD_ITYPE_FW_BACKUP          (0x05)
#define MPI2_FW_UPLOAD_ITYPE_MANUFACTURING      (0x06)
#define MPI2_FW_UPLOAD_ITYPE_CONFIG_1           (0x07)
#define MPI2_FW_UPLOAD_ITYPE_CONFIG_2           (0x08)
#define MPI2_FW_UPLOAD_ITYPE_MEGARAID           (0x09)
#define MPI2_FW_UPLOAD_ITYPE_COMPLETE           (0x0A)
#define MPI2_FW_UPLOAD_ITYPE_COMMON_BOOT_BLOCK  (0x0B)

/* MPI v2.0 FWUpload TransactionContext Element */
typedef struct _MPI2_FW_UPLOAD_TCSGE
{
    U8                      Reserved1;              /* 0x00 */
    U8                      ContextSize;            /* 0x01 */
    U8                      DetailsLength;          /* 0x02 */
    U8                      Flags;                  /* 0x03 */
    U32                     Reserved2;              /* 0x04 */
    U32                     ImageOffset;            /* 0x08 */
    U32                     ImageSize;              /* 0x0C */
} MPI2_FW_UPLOAD_TCSGE, MPI2_POINTER PTR_MPI2_FW_UPLOAD_TCSGE,
  Mpi2FWUploadTCSGE_t, MPI2_POINTER pMpi2FWUploadTCSGE_t;


/* MPI v2.5 FWUpload Request message */
typedef struct _MPI25_FW_UPLOAD_REQUEST
{
```

```
    U8                      ImageType;                  /* 0x00 */
    U8                      Reserved1;                  /* 0x01 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved4;                  /* 0x0A */
    U32                     Reserved5;                  /* 0x0C */
    U32                     Reserved6;                  /* 0x10 */
    U32                     Reserved7;                  /* 0x14 */
    U32                     ImageOffset;                /* 0x18 */
    U32                     ImageSize;                  /* 0x1C */
    MPI25_SGE_IO_UNION      SGL;                        /* 0x20 */
} MPI25_FW_UPLOAD_REQUEST, MPI2_POINTER PTR_MPI25_FW_UPLOAD_REQUEST,
  Mpi25FWUploadRequest_t, MPI2_POINTER pMpi25FWUploadRequest_t;


/* FWUpload Reply message */
typedef struct _MPI2_FW_UPLOAD_REPLY
{
    U8                      ImageType;                  /* 0x00 */
    U8                      Reserved1;                  /* 0x01 */
    U8                      MsgLength;                  /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved4;                  /* 0x0A */
    U16                     Reserved5;                  /* 0x0C */
    U16                     IOCStatus;                  /* 0x0E */
    U32                     IOCLogInfo;                 /* 0x10 */
    U32                     ActualImageSize;            /* 0x14 */
} MPI2_FW_UPLOAD_REPLY, MPI2_POINTER PTR_MPI2_FW_UPLOAD_REPLY,
______unchanged_portion_omitted_
  Mpi2FWImageHeader_t, MPI2_POINTER pMpi2FWImageHeader_t;

/* Signature field */
#define MPI2_FW_HEADER_SIGNATURE_OFFSET         (0x00)
#define MPI2_FW_HEADER_SIGNATURE_MASK           (0xFF000000)
#define MPI2_FW_HEADER_SIGNATURE                (0xEA000000)

/* Signature0 field */
#define MPI2_FW_HEADER_SIGNATURE0_OFFSET        (0x04)
#define MPI2_FW_HEADER_SIGNATURE0               (0x5AFAA55A)

/* Signature1 field */
#define MPI2_FW_HEADER_SIGNATURE1_OFFSET        (0x08)
#define MPI2_FW_HEADER_SIGNATURE1               (0xA55AFAA5)

/* Signature2 field */
#define MPI2_FW_HEADER_SIGNATURE2_OFFSET        (0x0C)
#define MPI2_FW_HEADER_SIGNATURE2               (0x5AA55AFA)


/* defines for using the ProductID field */
#define MPI2_FW_HEADER_PID_TYPE_MASK            (0xF000)
#define MPI2_FW_HEADER_PID_TYPE_SAS             (0x2000)

#define MPI2_FW_HEADER_PID_PROD_MASK                    (0x0F00)
#define MPI2_FW_HEADER_PID_PROD_A                       (0x0000)
#define MPI2_FW_HEADER_PID_PROD_TARGET_INITIATOR_SCSI   (0x0200)
```



```
#define MPI2_FW_HEADER_PID_PROD_IR_SCSI                 (0x0700)


#define MPI2_FW_HEADER_PID_FAMILY_MASK          (0x00FF)
/* SAS ProductID Family bits */
#define MPI2_FW_HEADER_PID_FAMILY_2108_SAS      (0x0013)
#define MPI2_FW_HEADER_PID_FAMILY_2208_SAS      (0x0014)
#define MPI25_FW_HEADER_PID_FAMILY_3108_SAS     (0x0021)
/* SAS */
#define MPI2_FW_HEADER_PID_FAMILY_2108_SAS      (0x0010)
#define MPI2_FW_HEADER_PID_FAMILY_2208_SAS      (0x0011)

/* use MPI2_IOCFACTS_PROTOCOL_ defines for ProtocolFlags field */

/* use MPI2_IOCFACTS_CAPABILITY_ defines for IOCCapabilities field */


#define MPI2_FW_HEADER_IMAGESIZE_OFFSET         (0x2C)
#define MPI2_FW_HEADER_NEXTIMAGE_OFFSET         (0x30)
#define MPI2_FW_HEADER_VERNMHWAT_OFFSET         (0x64)

#define MPI2_FW_HEADER_WHAT_SIGNATURE           (0x29232840)

#define MPI2_FW_HEADER_SIZE                     (0x100)


/* Extended Image Header */
typedef struct _MPI2_EXT_IMAGE_HEADER

{
    U8                      ImageType;                  /* 0x00 */
    U8                      Reserved1;                  /* 0x01 */
    U16                     Reserved2;                  /* 0x02 */
    U32                     Checksum;                   /* 0x04 */
    U32                     ImageSize;                  /* 0x08 */
    U32                     NextImageHeaderOffset;      /* 0x0C */
    U32                     PackageVersion;             /* 0x10 */
    U32                     Reserved3;                  /* 0x14 */
    U32                     Reserved4;                  /* 0x18 */
    U32                     Reserved5;                  /* 0x1C */
    U8                      IdentifyString[32];         /* 0x20 */
} MPI2_EXT_IMAGE_HEADER, MPI2_POINTER PTR_MPI2_EXT_IMAGE_HEADER,
  Mpi2ExtImageHeader_t, MPI2_POINTER pMpi2ExtImageHeader_t;

/* useful offsets */
#define MPI2_EXT_IMAGE_IMAGETYPE_OFFSET         (0x00)
#define MPI2_EXT_IMAGE_IMAGESIZE_OFFSET         (0x08)
#define MPI2_EXT_IMAGE_NEXTIMAGE_OFFSET         (0x0C)

#define MPI2_EXT_IMAGE_HEADER_SIZE              (0x40)

/* defines for the ImageType field */
#define MPI2_EXT_IMAGE_TYPE_UNSPECIFIED             (0x00)
#define MPI2_EXT_IMAGE_TYPE_FW                      (0x01)
#define MPI2_EXT_IMAGE_TYPE_NVDATA                  (0x03)
#define MPI2_EXT_IMAGE_TYPE_BOOTLOADER              (0x04)
#define MPI2_EXT_IMAGE_TYPE_INITIALIZATION          (0x05)
#define MPI2_EXT_IMAGE_TYPE_FLASH_LAYOUT            (0x06)
#define MPI2_EXT_IMAGE_TYPE_SUPPORTED_DEVICES       (0x07)
#define MPI2_EXT_IMAGE_TYPE_MEGARAID                (0x08)
#define MPI2_EXT_IMAGE_TYPE_ENCRYPTED_HASH          (0x09) /* MPI v2.5 and newer
#define MPI2_EXT_IMAGE_TYPE_MIN_PRODUCT_SPECIFIC    (0x80)
#define MPI2_EXT_IMAGE_TYPE_MAX_PRODUCT_SPECIFIC    (0xFF)

#define MPI2_EXT_IMAGE_TYPE_MAX (MPI2_EXT_IMAGE_TYPE_MAX_PRODUCT_SPECIFIC)  /* d
#define MPI2_EXT_IMAGE_TYPE_MAX                 (MPI2_EXT_IMAGE_TYPE_MEGARAID)
```

```
1545 /* FLASH Layout Extended Image Data */

1547 /*
1548  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
1549  * one and check RegionsPerLayout at runtime.
1550  */
1551 #ifndef MPI2_FLASH_NUMBER_OF_REGIONS
1552 #define MPI2_FLASH_NUMBER_OF_REGIONS        (1)
1553 #endif

1555 /*
1556  * Host code (drivers, BIOS, utilities, etc.) should leave this define set to
1557  * one and check NumberOfLayouts at runtime.
1558  */
1559 #ifndef MPI2_FLASH_NUMBER_OF_LAYOUTS
1560 #define MPI2_FLASH_NUMBER_OF_LAYOUTS        (1)
1561 #endif

1563 typedef struct _MPI2_FLASH_REGION
1564 {
1565     U8                      RegionType;             /* 0x00 */
1566     U8                      Reserved1;              /* 0x01 */
1567     U16                     Reserved2;              /* 0x02 */
1568     U32                     RegionOffset;           /* 0x04 */
1569     U32                     RegionSize;             /* 0x08 */
1570     U32                     Reserved3;              /* 0x0C */
1571 } MPI2_FLASH_REGION, MPI2_POINTER PTR_MPI2_FLASH_REGION,
______unchanged_portion_omitted_
1666   Mpi2InitImageFooter_t, MPI2_POINTER pMpi2InitImageFooter_t;

1668 /* defines for the BootFlags field */
1669 #define MPI2_INIT_IMAGE_BOOTFLAGS_OFFSET        (0x00)

1671 /* defines for the ImageSize field */
1672 #define MPI2_INIT_IMAGE_IMAGESIZE_OFFSET        (0x04)

1674 /* defines for the Signature0 field */
1675 #define MPI2_INIT_IMAGE_SIGNATURE0_OFFSET       (0x08)
1676 #define MPI2_INIT_IMAGE_SIGNATURE0              (0x5AA55AEA)

1678 /* defines for the Signature1 field */
1679 #define MPI2_INIT_IMAGE_SIGNATURE1_OFFSET       (0x0C)
1680 #define MPI2_INIT_IMAGE_SIGNATURE1              (0xA55AEAA5)

1682 /* defines for the Signature2 field */
1683 #define MPI2_INIT_IMAGE_SIGNATURE2_OFFSET       (0x10)
1684 #define MPI2_INIT_IMAGE_SIGNATURE2              (0x5AEAA55A)

1686 /* Signature fields as individual bytes */
1687 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_0        (0xEA)
1688 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_1        (0x5A)
1689 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_2        (0xA5)
1690 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_3        (0x5A)

1692 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_4        (0xA5)
1693 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_5        (0xEA)
1694 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_6        (0x5A)
1695 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_7        (0xA5)

1697 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_8        (0x5A)
1698 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_9        (0xA5)
1699 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_A        (0xEA)
1700 #define MPI2_INIT_IMAGE_SIGNATURE_BYTE_B        (0x5A)
```



```
1702 /* defines for the ResetVector field */
1703 #define MPI2_INIT_IMAGE_RESETVECTOR_OFFSET      (0x14)


1706 /* Encrypted Hash Extended Image Data */

1708 typedef struct _MPI25_ENCRYPTED_HASH_ENTRY
1709 {
1710     U8                  HashImageType;          /* 0x00 */
1711     U8                  HashAlgorithm;          /* 0x01 */
1712     U8                  EncryptionAlgorithm;    /* 0x02 */
1713     U8                  Reserved1;              /* 0x03 */
1714     U32                 Reserved2;              /* 0x04 */
1715     U32                 EncryptedHash[1];       /* 0x08 */ /* variable length */
1716 } MPI25_ENCRYPTED_HASH_ENTRY, MPI2_POINTER PTR_MPI25_ENCRYPTED_HASH_ENTRY,
1717   Mpi25EncryptedHashEntry_t, MPI2_POINTER pMpi25EncryptedHashEntry_t;

1719 /* values for HashImageType */
1720 #define MPI25_HASH_IMAGE_TYPE_UNUSED            (0x00)
1721 #define MPI25_HASH_IMAGE_TYPE_FIRMWARE          (0x01)
1722 #define MPI25_HASH_IMAGE_TYPE_BIOS              (0x02)

1724 /* values for HashAlgorithm */
1725 #define MPI25_HASH_ALGORITHM_UNUSED             (0x00)
1726 #define MPI25_HASH_ALGORITHM_SHA256             (0x01)

1728 /* values for EncryptionAlgorithm */
1729 #define MPI25_ENCRYPTION_ALG_UNUSED             (0x00)
1730 #define MPI25_ENCRYPTION_ALG_RSA256             (0x01)

1732 typedef struct _MPI25_ENCRYPTED_HASH_DATA
1733 {
1734     U8                              ImageVersion;           /* 0x00 */
1735     U8                              NumHash;                /* 0x01 */
1736     U16                             Reserved1;              /* 0x02 */
1737     U32                             Reserved2;              /* 0x04 */
1738     MPI25_ENCRYPTED_HASH_ENTRY      EncryptedHashEntry[1];  /* 0x08 */ /* variab
1739 } MPI25_ENCRYPTED_HASH_DATA, MPI2_POINTER PTR_MPI25_ENCRYPTED_HASH_DATA,
1740   Mpi25EncryptedHashData_t, MPI2_POINTER pMpi25EncryptedHashData_t;

1742 /****************************************************************************
1743 *  PowerManagementControl message
1744 ****************************************************************************/

1746 /* PowerManagementControl Request message */
1747 typedef struct _MPI2_PWR_MGMT_CONTROL_REQUEST
1748 {
1749     U8                      Feature;                /* 0x00 */
1750     U8                      Reserved1;              /* 0x01 */
1751     U8                      ChainOffset;            /* 0x02 */
1752     U8                      Function;               /* 0x03 */
1753     U16                     Reserved2;              /* 0x04 */
1754     U8                      Reserved3;              /* 0x06 */
1755     U8                      MsgFlags;               /* 0x07 */
1756     U8                      VP_ID;                  /* 0x08 */
1757     U8                      VF_ID;                  /* 0x09 */
1758     U16                     Reserved4;              /* 0x0A */
1759     U8                      Parameter1;             /* 0x0C */
1760     U8                      Parameter2;             /* 0x0D */
1761     U8                      Parameter3;             /* 0x0E */
1762     U8                      Parameter4;             /* 0x0F */
1763     U32                     Reserved5;              /* 0x10 */
1764     U32                     Reserved6;              /* 0x14 */
1765 } MPI2_PWR_MGMT_CONTROL_REQUEST, MPI2_POINTER PTR_MPI2_PWR_MGMT_CONTROL_REQUEST,
1766   Mpi2PwrMgmtControlRequest_t, MPI2_POINTER pMpi2PwrMgmtControlRequest_t;
```

```
/* defines for the Feature field */
#define MPI2_PM_CONTROL_FEATURE_DA_PHY_POWER_COND       (0x01)
#define MPI2_PM_CONTROL_FEATURE_PORT_WIDTH_MODULATION   (0x02)
#define MPI2_PM_CONTROL_FEATURE_PCIE_LINK               (0x03) /* obsolete */
#define MPI2_PM_CONTROL_FEATURE_IOC_SPEED               (0x04)
#define MPI2_PM_CONTROL_FEATURE_GLOBAL_PWR_MGMT_MODE    (0x05) /* reserved in MP
#define MPI2_PM_CONTROL_FEATURE_MIN_PRODUCT_SPECIFIC    (0x80)
#define MPI2_PM_CONTROL_FEATURE_MAX_PRODUCT_SPECIFIC    (0xFF)

/* parameter usage for the MPI2_PM_CONTROL_FEATURE_DA_PHY_POWER_COND Feature */
/* Parameter1 contains a PHY number */
/* Parameter2 indicates power condition action using these defines */
#define MPI2_PM_CONTROL_PARAM2_PARTIAL                  (0x01)
#define MPI2_PM_CONTROL_PARAM2_SLUMBER                  (0x02)
#define MPI2_PM_CONTROL_PARAM2_EXIT_PWR_MGMT            (0x03)
/* Parameter3 and Parameter4 are reserved */

/* parameter usage for the MPI2_PM_CONTROL_FEATURE_PORT_WIDTH_MODULATION Feature
/* Parameter1 contains SAS port width modulation group number */
/* Parameter2 indicates IOC action using these defines */
#define MPI2_PM_CONTROL_PARAM2_REQUEST_OWNERSHIP        (0x01)
#define MPI2_PM_CONTROL_PARAM2_CHANGE_MODULATION        (0x02)
#define MPI2_PM_CONTROL_PARAM2_RELINQUISH_OWNERSHIP     (0x03)
/* Parameter3 indicates desired modulation level using these defines */
#define MPI2_PM_CONTROL_PARAM3_25_PERCENT               (0x00)
#define MPI2_PM_CONTROL_PARAM3_50_PERCENT               (0x01)
#define MPI2_PM_CONTROL_PARAM3_75_PERCENT               (0x02)
#define MPI2_PM_CONTROL_PARAM3_100_PERCENT              (0x03)
/* Parameter4 is reserved */

/* this next set (_PCIE_LINK) is obsolete */
/* parameter usage for the MPI2_PM_CONTROL_FEATURE_PCIE_LINK Feature */
/* Parameter1 indicates desired PCIe link speed using these defines */
#define MPI2_PM_CONTROL_PARAM1_PCIE_2_5_GBPS            (0x00) /* obsolete */
#define MPI2_PM_CONTROL_PARAM1_PCIE_5_0_GBPS            (0x01) /* obsolete */
#define MPI2_PM_CONTROL_PARAM1_PCIE_8_0_GBPS            (0x02) /* obsolete */
/* Parameter2 indicates desired PCIe link width using these defines */
#define MPI2_PM_CONTROL_PARAM2_WIDTH_X1                 (0x01) /* obsolete */
#define MPI2_PM_CONTROL_PARAM2_WIDTH_X2                 (0x02) /* obsolete */
#define MPI2_PM_CONTROL_PARAM2_WIDTH_X4                 (0x04) /* obsolete */
#define MPI2_PM_CONTROL_PARAM2_WIDTH_X8                 (0x08) /* obsolete */
/* Parameter3 and Parameter4 are reserved */

/* parameter usage for the MPI2_PM_CONTROL_FEATURE_IOC_SPEED Feature */
/* Parameter1 indicates desired IOC hardware clock speed using these defines */
#define MPI2_PM_CONTROL_PARAM1_FULL_IOC_SPEED           (0x01)
#define MPI2_PM_CONTROL_PARAM1_HALF_IOC_SPEED           (0x02)
#define MPI2_PM_CONTROL_PARAM1_QUARTER_IOC_SPEED        (0x04)
#define MPI2_PM_CONTROL_PARAM1_EIGHTH_IOC_SPEED         (0x08)
/* Parameter2, Parameter3, and Parameter4 are reserved */

/* parameter usage for the MPI2_PM_CONTROL_FEATURE_GLOBAL_PWR_MGMT_MODE Feature
/* Parameter1 indicates host action regarding global power management mode */
#define MPI2_PM_CONTROL_PARAM1_TAKE_CONTROL             (0x01)
#define MPI2_PM_CONTROL_PARAM1_CHANGE_GLOBAL_MODE       (0x02)
#define MPI2_PM_CONTROL_PARAM1_RELEASE_CONTROL          (0x03)
/* Parameter2 indicates the requested global power management mode */
#define MPI2_PM_CONTROL_PARAM2_FULL_PWR_PERF            (0x01)
#define MPI2_PM_CONTROL_PARAM2_REDUCED_PWR_PERF         (0x08)
#define MPI2_PM_CONTROL_PARAM2_STANDBY                  (0x40)
/* Parameter3 and Parameter4 are reserved */


/* PowerManagementControl Reply message */
typedef struct _MPI2_PWR_MGMT_CONTROL_REPLY
```



```
{
    U8                      Feature;                /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      MsgLength;              /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved2;              /* 0x04 */
    U8                      Reserved3;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U16                     Reserved5;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
} MPI2_PWR_MGMT_CONTROL_REPLY, MPI2_POINTER PTR_MPI2_PWR_MGMT_CONTROL_REPLY,
  Mpi2PwrMgmtControlReply_t, MPI2_POINTER pMpi2PwrMgmtControlReply_t;


#endif
```

```
**********************************************************
    5332 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_ra.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

/*
 *  Copyright (c) 2009 LSI Corporation.
 *
 *
 *           Name:  mpi2_ra.h
 *          Title:  MPI RAID Accelerator messages and structures
 *  Creation Date:  April 13, 2009
 *
 *  mpi2_ra.h Version:  02.00.00
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  05-06-09  02.00.00  Initial version.
 *  ------------------------------------------------------------------------
 */

#ifndef MPI2_RA_H
#define MPI2_RA_H

/* generic structure for RAID Accelerator Control Block */
typedef struct _MPI2_RAID_ACCELERATOR_CONTROL_BLOCK
{
    U32                 Reserved[8];                /* 0x00 */
    U32                 RaidAcceleratorCDB[1];      /* 0x20 */
} MPI2_RAID_ACCELERATOR_CONTROL_BLOCK,
  MPI2_POINTER PTR_MPI2_RAID_ACCELERATOR_CONTROL_BLOCK,
  Mpi2RAIDAcceleratorControlBlock_t,
  MPI2_POINTER pMpi2RAIDAcceleratorControlBlock_t;
```



```
/*****************************************************************************
*
*        RAID Accelerator Messages
*
*****************************************************************************/

/* RAID Accelerator Request Message */
typedef struct _MPI2_RAID_ACCELERATOR_REQUEST
{
    U16                 Reserved0;                          /* 0x00 */
    U8                  ChainOffset;                        /* 0x02 */
    U8                  Function;                           /* 0x03 */
    U16                 Reserved1;                          /* 0x04 */
    U8                  Reserved2;                          /* 0x06 */
    U8                  MsgFlags;                           /* 0x07 */
    U8                  VP_ID;                              /* 0x08 */
    U8                  VF_ID;                              /* 0x09 */
    U16                 Reserved3;                          /* 0x0A */
    U64                 RaidAcceleratorControlBlockAddress; /* 0x0C */
    U8                  DmaEngineNumber;                    /* 0x14 */
    U8                  Reserved4;                          /* 0x15 */
    U16                 Reserved5;                          /* 0x16 */
    U32                 Reserved6;                          /* 0x18 */
    U32                 Reserved7;                          /* 0x1C */
    U32                 Reserved8;                          /* 0x20 */
} MPI2_RAID_ACCELERATOR_REQUEST, MPI2_POINTER PTR_MPI2_RAID_ACCELERATOR_REQUEST,
  Mpi2RAIDAcceleratorRequest_t, MPI2_POINTER pMpi2RAIDAcceleratorRequest_t;


/* RAID Accelerator Error Reply Message */
typedef struct _MPI2_RAID_ACCELERATOR_REPLY
{
    U16                 Reserved0;                  /* 0x00 */
    U8                  MsgLength;                  /* 0x02 */
    U8                  Function;                   /* 0x03 */
    U16                 Reserved1;                  /* 0x04 */
    U8                  Reserved2;                  /* 0x06 */
    U8                  MsgFlags;                   /* 0x07 */
    U8                  VP_ID;                      /* 0x08 */
    U8                  VF_ID;                      /* 0x09 */
    U16                 Reserved3;                  /* 0x0A */
    U16                 Reserved4;                  /* 0x0C */
    U16                 IOCStatus;                  /* 0x0E */
    U32                 IOCLogInfo;                 /* 0x10 */
    U32                 ProductSpecificData[3];     /* 0x14 */
} MPI2_RAID_ACCELERATOR_REPLY, MPI2_POINTER PTR_MPI2_RAID_ACCELERATOR_REPLY,
  Mpi2RAIDAcceleratorReply_t, MPI2_POINTER pMpi2RAIDAcceleratorReply_t;


#endif
```

```
**********************************************************
   19273 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_raid.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 */
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_raid.h
 *         Title:  MPI Integrated RAID messages and structures
 *  Creation Date:  April 26, 2007
 *
 *    mpi2_raid.h Version:  02.00.10
 *    mpi2_raid.h Version:  02.00.04
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  08-31-07  02.00.01  Modifications to RAID Action request and reply,
 *                      including the Actions and ActionData.
 *  02-29-08  02.00.02  Added MPI2_RAID_ACTION_ADATA_DISABL_FULL_REBUILD.
 *  05-21-08  02.00.03  Added MPI2_RAID_VOL_CREATION_NUM_PHYSDISKS so that
 *                      the PhysDisk array in MPI2_RAID_VOLUME_CREATION_STRUCT
 *                      can be sized by the build environment.
 *  07-30-09  02.00.04  Added proper define for the Use Default Settings bit of
 *                      VolumeCreationFlags and marked the old one as obsolete.
 *  05-12-10  02.00.05  Added MPI2_RAID_VOL_FLAGS_OP_MDC define.
 *  08-24-10  02.00.06  Added MPI2_RAID_ACTION_COMPATIBILITY_CHECK along with
 *                      related structures and defines.
 *                      Added product-specific range to RAID Action values.
 *  11-18-11  02.00.07  Incorporating additions for MPI v2.5.
 *  02-06-12  02.00.08  Added MPI2_RAID_ACTION_PHYSDISK_HIDDEN.
 *  07-26-12  02.00.09  Added ElapsedSeconds field to MPI2_RAID_VOL_INDICATOR.
 *                      Added MPI2_RAID_VOL_FLAGS_ELAPSED_SECONDS_VALID define.
 *  04-17-13  02.00.10  Added MPI25_RAID_ACTION_ADATA_ALLOW_PI.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_RAID_H
#define MPI2_RAID_H

/*****************************************************************************
*
*               Integrated RAID Messages
*
*****************************************************************************/

/*****************************************************************************
*  RAID Action messages
*****************************************************************************/

/* ActionDataWord defines for use with MPI2_RAID_ACTION_CREATE_VOLUME action */
#define MPI25_RAID_ACTION_ADATA_ALLOW_PI            (0x80000000)

/* ActionDataWord defines for use with MPI2_RAID_ACTION_DELETE_VOLUME action */
#define MPI2_RAID_ACTION_ADATA_KEEP_LBA0            (0x00000000)
#define MPI2_RAID_ACTION_ADATA_ZERO_LBA0            (0x00000001)
```

```
 86 /* use MPI2_RAIDVOL0_SETTING_ defines from mpi2_cnfg.h for MPI2_RAID_ACTION_CHAN

 88 /* ActionDataWord defines for use with MPI2_RAID_ACTION_DISABLE_ALL_VOLUMES acti
 89 #define MPI2_RAID_ACTION_ADATA_DISABL_FULL_REBUILD  (0x00000001)

 91 /* ActionDataWord for MPI2_RAID_ACTION_SET_RAID_FUNCTION_RATE Action */
 92 typedef struct _MPI2_RAID_ACTION_RATE_DATA
 93 {
 94     U8              RateToChange;           /* 0x00 */
 95     U8              RateOrMode;             /* 0x01 */
 96     U16             DataScrubDuration;      /* 0x02 */
 97 } MPI2_RAID_ACTION_RATE_DATA, MPI2_POINTER PTR_MPI2_RAID_ACTION_RATE_DATA,
_____unchanged_portion_omitted_
196   Mpi2RaidActionRequest_t, MPI2_POINTER pMpi2RaidActionRequest_t;

198 /* RAID Action request Action values */

200 #define MPI2_RAID_ACTION_INDICATOR_STRUCT           (0x01)
201 #define MPI2_RAID_ACTION_CREATE_VOLUME              (0x02)
202 #define MPI2_RAID_ACTION_DELETE_VOLUME              (0x03)
203 #define MPI2_RAID_ACTION_DISABLE_ALL_VOLUMES        (0x04)
204 #define MPI2_RAID_ACTION_ENABLE_ALL_VOLUMES         (0x05)
205 #define MPI2_RAID_ACTION_PHYSDISK_OFFLINE           (0x0A)
206 #define MPI2_RAID_ACTION_PHYSDISK_ONLINE            (0x0B)
207 #define MPI2_RAID_ACTION_FAIL_PHYSDISK              (0x0F)
208 #define MPI2_RAID_ACTION_ACTIVATE_VOLUME            (0x11)
209 #define MPI2_RAID_ACTION_DEVICE_FW_UPDATE_MODE      (0x15)
210 #define MPI2_RAID_ACTION_CHANGE_VOL_WRITE_CACHE     (0x17)
211 #define MPI2_RAID_ACTION_SET_VOLUME_NAME            (0x18)
212 #define MPI2_RAID_ACTION_SET_RAID_FUNCTION_RATE     (0x19)
213 #define MPI2_RAID_ACTION_ENABLE_FAILED_VOLUME       (0x1C)
214 #define MPI2_RAID_ACTION_CREATE_HOT_SPARE           (0x1D)
215 #define MPI2_RAID_ACTION_DELETE_HOT_SPARE           (0x1E)
216 #define MPI2_RAID_ACTION_SYSTEM_SHUTDOWN_INITIATED  (0x20)
217 #define MPI2_RAID_ACTION_START_RAID_FUNCTION        (0x21)
218 #define MPI2_RAID_ACTION_STOP_RAID_FUNCTION         (0x22)
219 #define MPI2_RAID_ACTION_COMPATIBILITY_CHECK        (0x23)
220 #define MPI2_RAID_ACTION_PHYSDISK_HIDDEN            (0x24)
221 #define MPI2_RAID_ACTION_MIN_PRODUCT_SPECIFIC       (0x80)
222 #define MPI2_RAID_ACTION_MAX_PRODUCT_SPECIFIC       (0xFF)


225 /* RAID Volume Creation Structure */

227 /*
228  * The following define can be customized for the targeted product.
229  */
230 #ifndef MPI2_RAID_VOL_CREATION_NUM_PHYSDISKS
231 #define MPI2_RAID_VOL_CREATION_NUM_PHYSDISKS        (1)
232 #endif

234 typedef struct _MPI2_RAID_VOLUME_PHYSDISK
235 {
236     U8                      RAIDSetNum;                 /* 0x00 */
237     U8                      PhysDiskMap;                /* 0x01 */
238     U16                     PhysDiskDevHandle;          /* 0x02 */
239 } MPI2_RAID_VOLUME_PHYSDISK, MPI2_POINTER PTR_MPI2_RAID_VOLUME_PHYSDISK,
_____unchanged_portion_omitted_
261   MPI2_POINTER PTR_MPI2_RAID_VOLUME_CREATION_STRUCT,
262   Mpi2RaidVolumeCreationStruct_t, MPI2_POINTER pMpi2RaidVolumeCreationStruct_t;

264 /* use MPI2_RAID_VOL_TYPE_ defines from mpi2_cnfg.h for VolumeType */

266 /* defines for the VolumeCreationFlags field */
267 #define MPI2_RAID_VOL_CREATION_DEFAULT_SETTINGS     (0x80000000)
```



```
268 #define MPI2_RAID_VOL_CREATION_BACKGROUND_INIT      (0x00000004) /* MPI 2.0 only
268 #define MPI2_RAID_VOL_CREATION_BACKGROUND_INIT      (0x00000004)
269 #define MPI2_RAID_VOL_CREATION_LOW_LEVEL_INIT       (0x00000002)
270 #define MPI2_RAID_VOL_CREATION_MIGRATE_DATA         (0x00000001)
271 /* The following is an obsolete define.
272  * It must be shifted left 24 bits in order to set the proper bit.
273  */
274 #define MPI2_RAID_VOL_CREATION_USE_DEFAULT_SETTINGS (0x80)


277 /* RAID Online Capacity Expansion Structure */

279 typedef struct _MPI2_RAID_ONLINE_CAPACITY_EXPANSION
280 {
281     U32                     Flags;                      /* 0x00 */
282     U16                     DevHandle0;                 /* 0x04 */
283     U16                     Reserved1;                  /* 0x06 */
284     U16                     DevHandle1;                 /* 0x08 */
285     U16                     Reserved2;                  /* 0x0A */
286 } MPI2_RAID_ONLINE_CAPACITY_EXPANSION,
287   MPI2_POINTER PTR_MPI2_RAID_ONLINE_CAPACITY_EXPANSION,
288   Mpi2RaidOnlineCapacityExpansion_t,
289   MPI2_POINTER pMpi2RaidOnlineCapacityExpansion_t;


292 /* RAID Compatibility Input Structure */

294 typedef struct _MPI2_RAID_COMPATIBILITY_INPUT_STRUCT
295 {
296     U16                     SourceDevHandle;            /* 0x00 */
297     U16                     CandidateDevHandle;         /* 0x02 */
298     U32                     Flags;                      /* 0x04 */
299     U32                     Reserved1;                  /* 0x08 */
300     U32                     Reserved2;                  /* 0x0C */
301 } MPI2_RAID_COMPATIBILITY_INPUT_STRUCT,
302   MPI2_POINTER PTR_MPI2_RAID_COMPATIBILITY_INPUT_STRUCT,
303   Mpi2RaidCompatibilityInputStruct_t,
304   MPI2_POINTER pMpi2RaidCompatibilityInputStruct_t;

306 /* defines for RAID Compatibility Structure Flags field */
307 #define MPI2_RAID_COMPAT_SOURCE_IS_VOLUME_FLAG      (0x00000002)
308 #define MPI2_RAID_COMPAT_REPORT_SOURCE_INFO_FLAG    (0x00000001)


311 /* RAID Volume Indicator Structure */

313 typedef struct _MPI2_RAID_VOL_INDICATOR
314 {
315     U64                     TotalBlocks;                /* 0x00 */
316     U64                     BlocksRemaining;            /* 0x08 */
317     U32                     Flags;                      /* 0x10 */
318     U32                     ElapsedSeconds;             /* 0x14 */
319 } MPI2_RAID_VOL_INDICATOR, MPI2_POINTER PTR_MPI2_RAID_VOL_INDICATOR,
320   Mpi2RaidVolIndicator_t, MPI2_POINTER pMpi2RaidVolIndicator_t;

322 /* defines for RAID Volume Indicator Flags field */
323 #define MPI2_RAID_VOL_FLAGS_ELAPSED_SECONDS_VALID   (0x80000000)

325 #define MPI2_RAID_VOL_FLAGS_OP_MASK                 (0x0000000F)
326 #define MPI2_RAID_VOL_FLAGS_OP_BACKGROUND_INIT      (0x00000000)
327 #define MPI2_RAID_VOL_FLAGS_OP_ONLINE_CAP_EXPANSION (0x00000001)
328 #define MPI2_RAID_VOL_FLAGS_OP_CONSISTENCY_CHECK    (0x00000002)
329 #define MPI2_RAID_VOL_FLAGS_OP_RESYNC               (0x00000003)
330 #define MPI2_RAID_VOL_FLAGS_OP_MDC                  (0x00000004)
```

```
 333 /* RAID Compatibility Result Structure */

 335 typedef struct _MPI2_RAID_COMPATIBILITY_RESULT_STRUCT
 336 {
 337     U8                      State;                          /* 0x00 */
 338     U8                      Reserved1;                      /* 0x01 */
 339     U16                     Reserved2;                      /* 0x02 */
 340     U32                     GenericAttributes;              /* 0x04 */
 341     U32                     OEMSpecificAttributes;          /* 0x08 */
 342     U32                     Reserved3;                      /* 0x0C */
 343     U32                     Reserved4;                      /* 0x10 */
 344 } MPI2_RAID_COMPATIBILITY_RESULT_STRUCT,
 345   MPI2_POINTER PTR_MPI2_RAID_COMPATIBILITY_RESULT_STRUCT,
 346   Mpi2RaidCompatibilityResultStruct_t,
 347   MPI2_POINTER pMpi2RaidCompatibilityResultStruct_t;

 349 /* defines for RAID Compatibility Result Structure State field */
 350 #define MPI2_RAID_COMPAT_STATE_COMPATIBLE           (0x00)
 351 #define MPI2_RAID_COMPAT_STATE_NOT_COMPATIBLE       (0x01)

 353 /* defines for RAID Compatibility Result Structure GenericAttributes field */
 354 #define MPI2_RAID_COMPAT_GENATTRIB_4K_SECTOR            (0x00000010)

 356 #define MPI2_RAID_COMPAT_GENATTRIB_MEDIA_MASK           (0x0000000C)
 357 #define MPI2_RAID_COMPAT_GENATTRIB_SOLID_STATE_DRIVE    (0x00000008)
 358 #define MPI2_RAID_COMPAT_GENATTRIB_HARD_DISK_DRIVE      (0x00000004)

 360 #define MPI2_RAID_COMPAT_GENATTRIB_PROTOCOL_MASK        (0x00000003)
 361 #define MPI2_RAID_COMPAT_GENATTRIB_SAS_PROTOCOL         (0x00000002)
 362 #define MPI2_RAID_COMPAT_GENATTRIB_SATA_PROTOCOL        (0x00000001)


 365 /* RAID Action Reply ActionData union */
 366 typedef union _MPI2_RAID_ACTION_REPLY_DATA
 367 {
 368     U32                                     Word[6];
 313     U32                     Word[5];
 369     MPI2_RAID_VOL_INDICATOR                 RaidVolumeIndicator;
 370     U16                                     VolDevHandle;
 371     U8                                      VolumeState;
 372     U8                                      PhysDiskNum;
 373     MPI2_RAID_COMPATIBILITY_RESULT_STRUCT   RaidCompatibilityResult;
 374 } MPI2_RAID_ACTION_REPLY_DATA, MPI2_POINTER PTR_MPI2_RAID_ACTION_REPLY_DATA,
```

*unchanged_portion_omitted_*

```
**********************************************************
   16626 Mon Jun 16 21:18:07 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_sas.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *           Name:  mpi2_sas.h
 *          Title:  MPI Serial Attached SCSI structures and definitions
 *  Creation Date:  February 9, 2007
 *
 *  mpi2_sas.h Version:  02.00.08
 *  mpi2.h Version:  02.00.02
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  06-26-07  02.00.01  Added Clear All Persistent Operation to SAS IO Unit
 *                      Control Request.
 *  10-02-08  02.00.02  Added Set IOC Parameter Operation to SAS IO Unit Control
 *                      Request.
 *  10-28-09  02.00.03  Changed the type of SGL in MPI2_SATA_PASSTHROUGH_REQUEST
 *                      to MPI2_SGE_IO_UNION since it supports chained SGLs.
 *  05-12-10  02.00.04  Modified some comments.
 *  08-11-10  02.00.05  Added NCQ operations to SAS IO Unit Control.
 *  11-18-11  02.00.06  Incorporating additions for MPI v2.5.
 *  07-10-12  02.00.07  Added MPI2_SATA_PT_SGE_UNION for use in the SATA
 *                      Passthrough Request message.
 *  08-19-13  02.00.08  Made MPI2_SAS_OP_TRANSMIT_PORT_SELECT_SIGNAL obsolete
 *                      for anything newer than MPI v2.0.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_SAS_H
#define MPI2_SAS_H

/*
 * Values for SASStatus.
 */
#define MPI2_SASSTATUS_SUCCESS                          (0x00)
#define MPI2_SASSTATUS_UNKNOWN_ERROR                    (0x01)
#define MPI2_SASSTATUS_INVALID_FRAME                    (0x02)
#define MPI2_SASSTATUS_UTC_BAD_DEST                     (0x03)
#define MPI2_SASSTATUS_UTC_BREAK_RECEIVED               (0x04)
#define MPI2_SASSTATUS_UTC_CONNECT_RATE_NOT_SUPPORTED   (0x05)
#define MPI2_SASSTATUS_UTC_PORT_LAYER_REQUEST           (0x06)
#define MPI2_SASSTATUS_UTC_PROTOCOL_NOT_SUPPORTED       (0x07)
#define MPI2_SASSTATUS_UTC_STP_RESOURCES_BUSY           (0x08)
#define MPI2_SASSTATUS_UTC_WRONG_DESTINATION            (0x09)
#define MPI2_SASSTATUS_SHORT_INFORMATION_UNIT           (0x0A)
#define MPI2_SASSTATUS_LONG_INFORMATION_UNIT            (0x0B)
```

```
 85 #define MPI2_SASSTATUS_XFER_RDY_INCORRECT_WRITE_DATA    (0x0C)
 86 #define MPI2_SASSTATUS_XFER_RDY_REQUEST_OFFSET_ERROR    (0x0D)
 87 #define MPI2_SASSTATUS_XFER_RDY_NOT_EXPECTED            (0x0E)
 88 #define MPI2_SASSTATUS_DATA_INCORRECT_DATA_LENGTH       (0x0F)
 89 #define MPI2_SASSTATUS_DATA_TOO_MUCH_READ_DATA          (0x10)
 90 #define MPI2_SASSTATUS_DATA_OFFSET_ERROR                (0x11)
 91 #define MPI2_SASSTATUS_SDSF_NAK_RECEIVED                (0x12)
 92 #define MPI2_SASSTATUS_SDSF_CONNECTION_FAILED           (0x13)
 93 #define MPI2_SASSTATUS_INITIATOR_RESPONSE_TIMEOUT       (0x14)


 96 /*
 97  * Values for the SAS DeviceInfo field used in SAS Device Status Change Event
 98  * data and SAS Configuration pages.
 99  */
100 #define MPI2_SAS_DEVICE_INFO_SEP                (0x00004000)
101 #define MPI2_SAS_DEVICE_INFO_ATAPI_DEVICE       (0x00002000)
102 #define MPI2_SAS_DEVICE_INFO_LSI_DEVICE         (0x00001000)
103 #define MPI2_SAS_DEVICE_INFO_DIRECT_ATTACH      (0x00000800)
104 #define MPI2_SAS_DEVICE_INFO_SSP_TARGET         (0x00000400)
105 #define MPI2_SAS_DEVICE_INFO_STP_TARGET         (0x00000200)
106 #define MPI2_SAS_DEVICE_INFO_SMP_TARGET         (0x00000100)
107 #define MPI2_SAS_DEVICE_INFO_SATA_DEVICE        (0x00000080)
108 #define MPI2_SAS_DEVICE_INFO_SSP_INITIATOR      (0x00000040)
109 #define MPI2_SAS_DEVICE_INFO_STP_INITIATOR      (0x00000020)
110 #define MPI2_SAS_DEVICE_INFO_SMP_INITIATOR      (0x00000010)
111 #define MPI2_SAS_DEVICE_INFO_SATA_HOST          (0x00000008)

113 #define MPI2_SAS_DEVICE_INFO_MASK_DEVICE_TYPE   (0x00000007)
114 #define MPI2_SAS_DEVICE_INFO_NO_DEVICE          (0x00000000)
115 #define MPI2_SAS_DEVICE_INFO_END_DEVICE         (0x00000001)
116 #define MPI2_SAS_DEVICE_INFO_EDGE_EXPANDER      (0x00000002)
117 #define MPI2_SAS_DEVICE_INFO_FANOUT_EXPANDER    (0x00000003)


120 /*****************************************************************************
121 *
122 *        SAS Messages
123 *
124 *****************************************************************************/

126 /*****************************************************************************
127 *  SMP Passthrough messages
128 *****************************************************************************/

130 /* SMP Passthrough Request Message */
131 typedef struct _MPI2_SMP_PASSTHROUGH_REQUEST
132 {
133     U8                      PassthroughFlags;   /* 0x00 */
134     U8                      PhysicalPort;       /* 0x01 */
135     U8                      ChainOffset;        /* 0x02 */
136     U8                      Function;           /* 0x03 */
137     U16                     RequestDataLength;  /* 0x04 */
138     U8                      SGLFlags;           /* 0x06 */ /* MPI v2.0 only. Res
140     U8                      SGLFlags;           /* 0x06 */
139     U8                      MsgFlags;           /* 0x07 */
140     U8                      VP_ID;              /* 0x08 */
141     U8                      VF_ID;              /* 0x09 */
142     U16                     Reserved1;          /* 0x0A */
143     U32                     Reserved2;          /* 0x0C */
144     U64                     SASAddress;         /* 0x10 */
145     U32                     Reserved3;          /* 0x18 */
146     U32                     Reserved4;          /* 0x1C */
147     MPI2_SIMPLE_SGE_UNION   SGL;                /* 0x20 */ /* MPI v2.5: IEEE Sim
149     MPI2_SIMPLE_SGE_UNION   SGL;                /* 0x20 */
148 } MPI2_SMP_PASSTHROUGH_REQUEST, MPI2_POINTER PTR_MPI2_SMP_PASSTHROUGH_REQUEST,
```



```
149   Mpi2SmpPassthroughRequest_t, MPI2_POINTER pMpi2SmpPassthroughRequest_t;

151 /* values for PassthroughFlags field */
152 #define MPI2_SMP_PT_REQ_PT_FLAGS_IMMEDIATE      (0x80)

154 /* MPI v2.0: use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */
156 /* values for SGLFlags field are in the SGL section of mpi2.h */


157 /* SMP Passthrough Reply Message */
158 typedef struct _MPI2_SMP_PASSTHROUGH_REPLY
159 {
160     U8                      PassthroughFlags;   /* 0x00 */
161     U8                      PhysicalPort;       /* 0x01 */
162     U8                      MsgLength;          /* 0x02 */
163     U8                      Function;           /* 0x03 */
164     U16                     ResponseDataLength; /* 0x04 */
165     U8                      SGLFlags;           /* 0x06 */
166     U8                      MsgFlags;           /* 0x07 */
167     U8                      VP_ID;              /* 0x08 */
168     U8                      VF_ID;              /* 0x09 */
169     U16                     Reserved1;          /* 0x0A */
170     U8                      Reserved2;          /* 0x0C */
171     U8                      SASStatus;          /* 0x0D */
172     U16                     IOCStatus;          /* 0x0E */
173     U32                     IOCLogInfo;         /* 0x10 */
174     U32                     Reserved3;          /* 0x14 */
175     U8                      ResponseData[4];    /* 0x18 */
176 } MPI2_SMP_PASSTHROUGH_REPLY, MPI2_POINTER PTR_MPI2_SMP_PASSTHROUGH_REPLY,
177   Mpi2SmpPassthroughReply_t, MPI2_POINTER pMpi2SmpPassthroughReply_t;

179 /* values for PassthroughFlags field */
180 #define MPI2_SMP_PT_REPLY_PT_FLAGS_IMMEDIATE    (0x80)

182 /* values for SASStatus field are at the top of this file */


185 /*****************************************************************************
186 *  SATA Passthrough messages
187 *****************************************************************************/

189 typedef union _MPI2_SATA_PT_SGE_UNION
190 {
191     MPI2_SGE_SIMPLE_UNION       MpiSimple;      /* MPI v2.0 only */
192     MPI2_SGE_CHAIN_UNION        MpiChain;       /* MPI v2.0 only */
193     MPI2_IEEE_SGE_SIMPLE_UNION  IeeeSimple;
194     MPI2_IEEE_SGE_CHAIN_UNION   IeeeChain;      /* MPI v2.0 only */
195     MPI25_IEEE_SGE_CHAIN64      IeeeChain64;    /* MPI v2.5 only */
196 } MPI2_SATA_PT_SGE_UNION, MPI2_POINTER PTR_MPI2_SATA_PT_SGE_UNION,
197   Mpi2SataPTSGEUnion_t, MPI2_POINTER pMpi2SataPTSGEUnion_t;


200 /* SATA Passthrough Request Message */
201 typedef struct _MPI2_SATA_PASSTHROUGH_REQUEST
202 {
203     U16                     DevHandle;          /* 0x00 */
204     U8                      ChainOffset;        /* 0x02 */
205     U8                      Function;           /* 0x03 */
206     U16                     PassthroughFlags;   /* 0x04 */
207     U8                      SGLFlags;           /* 0x06 */ /* MPI v2.0 only. Res
198     U8                      SGLFlags;           /* 0x06 */
208     U8                      MsgFlags;           /* 0x07 */
209     U8                      VP_ID;              /* 0x08 */
210     U8                      VF_ID;              /* 0x09 */
211     U16                     Reserved1;          /* 0x0A */
212     U32                     Reserved2;          /* 0x0C */
```

```
213     U32                     Reserved3;          /* 0x10 */
214     U32                     Reserved4;          /* 0x14 */
215     U32                     DataLength;         /* 0x18 */
216     U8                      CommandFIS[20];     /* 0x1C */
217     MPI2_SATA_PT_SGE_UNION  SGL;                /* 0x30 */ /* MPI v2.5: IEEE 64
208     MPI2_SIMPLE_SGE_UNION   SGL;                /* 0x20 */
218 } MPI2_SATA_PASSTHROUGH_REQUEST, MPI2_POINTER PTR_MPI2_SATA_PASSTHROUGH_REQUEST,
219   Mpi2SataPassthroughRequest_t, MPI2_POINTER pMpi2SataPassthroughRequest_t;

221 /* values for PassthroughFlags field */
222 #define MPI2_SATA_PT_REQ_PT_FLAGS_EXECUTE_DIAG      (0x0100)
223 #define MPI2_SATA_PT_REQ_PT_FLAGS_DMA               (0x0020)
224 #define MPI2_SATA_PT_REQ_PT_FLAGS_PIO               (0x0010)
225 #define MPI2_SATA_PT_REQ_PT_FLAGS_UNSPECIFIED_VU    (0x0004)
226 #define MPI2_SATA_PT_REQ_PT_FLAGS_WRITE             (0x0002)
227 #define MPI2_SATA_PT_REQ_PT_FLAGS_READ              (0x0001)

229 /* MPI v2.0: use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */
220 /* values for SGLFlags field are in the SGL section of mpi2.h */


232 /* SATA Passthrough Reply Message */
233 typedef struct _MPI2_SATA_PASSTHROUGH_REPLY
234 {
235     U16                     DevHandle;          /* 0x00 */
236     U8                      MsgLength;          /* 0x02 */
237     U8                      Function;           /* 0x03 */
238     U16                     PassthroughFlags;   /* 0x04 */
239     U8                      SGLFlags;           /* 0x06 */
240     U8                      MsgFlags;           /* 0x07 */
241     U8                      VP_ID;              /* 0x08 */
242     U8                      VF_ID;              /* 0x09 */
243     U16                     Reserved1;          /* 0x0A */
244     U8                      Reserved2;          /* 0x0C */
245     U8                      SASStatus;          /* 0x0D */
246     U16                     IOCStatus;          /* 0x0E */
247     U32                     IOCLogInfo;         /* 0x10 */
248     U8                      StatusFIS[20];      /* 0x14 */
249     U32                     StatusControlRegisters; /* 0x28 */
250     U32                     TransferCount;      /* 0x2C */
251 } MPI2_SATA_PASSTHROUGH_REPLY, MPI2_POINTER PTR_MPI2_SATA_PASSTHROUGH_REPLY,
______unchanged_portion_omitted_
286   MPI2_POINTER PTR_MPI2_SAS_IOUNIT_CONTROL_REQUEST,
287   Mpi2SasIoUnitControlRequest_t, MPI2_POINTER pMpi2SasIoUnitControlRequest_t;

289 /* values for the Operation field */
290 #define MPI2_SAS_OP_CLEAR_ALL_PERSISTENT        (0x02)
291 #define MPI2_SAS_OP_PHY_LINK_RESET              (0x06)
292 #define MPI2_SAS_OP_PHY_HARD_RESET              (0x07)
293 #define MPI2_SAS_OP_PHY_CLEAR_ERROR_LOG         (0x08)
294 #define MPI2_SAS_OP_SEND_PRIMITIVE              (0x0A)
295 #define MPI2_SAS_OP_FORCE_FULL_DISCOVERY        (0x0B)
296 #define MPI2_SAS_OP_TRANSMIT_PORT_SELECT_SIGNAL (0x0C) /* MPI v2.0 only */
287 #define MPI2_SAS_OP_TRANSMIT_PORT_SELECT_SIGNAL (0x0C)
297 #define MPI2_SAS_OP_REMOVE_DEVICE               (0x0D)
298 #define MPI2_SAS_OP_LOOKUP_MAPPING              (0x0E)
299 #define MPI2_SAS_OP_SET_IOC_PARAMETER           (0x0F)
300 #define MPI25_SAS_OP_ENABLE_FP_DEVICE           (0x10)
301 #define MPI25_SAS_OP_DISABLE_FP_DEVICE          (0x11)
302 #define MPI25_SAS_OP_ENABLE_FP_ALL              (0x12)
303 #define MPI25_SAS_OP_DISABLE_FP_ALL             (0x13)
304 #define MPI2_SAS_OP_DEV_ENABLE_NCQ              (0x14)
305 #define MPI2_SAS_OP_DEV_DISABLE_NCQ             (0x15)
306 #define MPI2_SAS_OP_PRODUCT_SPECIFIC_MIN        (0x80)

308 /* values for the PrimFlags field */
```



```
309 #define MPI2_SAS_PRIMFLAGS_SINGLE               (0x08)
310 #define MPI2_SAS_PRIMFLAGS_TRIPLE               (0x02)
311 #define MPI2_SAS_PRIMFLAGS_REDUNDANT            (0x01)

313 /* values for the LookupMethod field */
314 #define MPI2_SAS_LOOKUP_METHOD_SAS_ADDRESS          (0x01)
315 #define MPI2_SAS_LOOKUP_METHOD_SAS_ENCLOSURE_SLOT   (0x02)
316 #define MPI2_SAS_LOOKUP_METHOD_SAS_DEVICE_NAME      (0x03)


319 /* SAS IO Unit Control Reply Message */
320 typedef struct _MPI2_SAS_IOUNIT_CONTROL_REPLY
321 {
322     U8                      Operation;          /* 0x00 */
323     U8                      Reserved1;          /* 0x01 */
324     U8                      MsgLength;          /* 0x02 */
325     U8                      Function;           /* 0x03 */
326     U16                     DevHandle;          /* 0x04 */
327     U8                      IOCParameter;       /* 0x06 */
328     U8                      MsgFlags;           /* 0x07 */
329     U8                      VP_ID;              /* 0x08 */
330     U8                      VF_ID;              /* 0x09 */
331     U16                     Reserved3;          /* 0x0A */
332     U16                     Reserved4;          /* 0x0C */
333     U16                     IOCStatus;          /* 0x0E */
334     U32                     IOCLogInfo;         /* 0x10 */
335 } MPI2_SAS_IOUNIT_CONTROL_REPLY,
______unchanged_portion_omitted_
```

```
**********************************************************
   31281 Mon Jun 16 21:18:08 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_targ.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */

/*
 *  Copyright (c) 2000-2012 LSI Corporation.
 *
 *
 *          Name:  mpi2_targ.h
 *         Title:  MPI Target mode messages and structures
 *  Creation Date:  September 8, 2006
 *
 *  mpi2_targ.h Version: 02.00.06
 *
 *  NOTE: Names (typedefs, defines, etc.) beginning with an MPI25 or Mpi25
 *        prefix are for use only on MPI v2.5 products, and must not be used
 *        with MPI v2.0 products. Unless otherwise noted, names beginning with
 *        MPI2 or Mpi2 are for use with both MPI v2.0 and MPI v2.5 products.
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  08-31-07  02.00.01  Added Command Buffer Data Location Address Space bits to
 *                      BufferPostFlags field of CommandBufferPostBase Request.
 *  02-29-08  02.00.02  Modified various names to make them 32-character unique.
 *  10-02-08  02.00.03  Removed NextCmdBufferOffset from
 *                      MPI2_TARGET_CMD_BUF_POST_BASE_REQUEST.
 *                      Target Status Send Request only takes a single SGE for
 *                      response data.
 *  02-10-10  02.00.04  Added comment to MPI2_TARGET_SSP_RSP_IU structure.
 *  11-18-11  02.00.05  Incorporating additions for MPI v2.5.
 *  11-27-12  02.00.06  Added InitiatorDevHandle field to MPI2_TARGET_MODE_ABORT
 *                      request message structure.
```



```
 *                      Added AbortType MPI2_TARGET_MODE_ABORT_DEVHANDLE and
 *                      MPI2_TARGET_MODE_ABORT_ALL_COMMANDS.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_TARG_H
#define MPI2_TARG_H


/*****************************************************************************
*
*        SCSI Target Messages
*
*****************************************************************************/

/****************************************************************************
*  Target Command Buffer Post Base Request
****************************************************************************/

typedef struct _MPI2_TARGET_CMD_BUF_POST_BASE_REQUEST
{
    U8                      BufferPostFlags;        /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     TotalCmdBuffers;        /* 0x04 */
    U8                      Reserved;               /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved2;              /* 0x0A */
    U32                     Reserved3;              /* 0x0C */
    U16                     CmdBufferLength;        /* 0x10 */
    U16                     Reserved4;              /* 0x12 */
    U32                     BaseAddressLow;         /* 0x14 */
    U32                     BaseAddressHigh;        /* 0x18 */
} MPI2_TARGET_CMD_BUF_POST_BASE_REQUEST,
  MPI2_POINTER PTR_MPI2_TARGET_CMD_BUF_POST_BASE_REQUEST,
  Mpi2TargetCmdBufferPostBaseRequest_t,
  MPI2_POINTER pMpi2TargetCmdBufferPostBaseRequest_t;

/* values for the BufferPostflags field */
#define MPI2_CMD_BUF_POST_BASE_ADDRESS_SPACE_MASK           (0x0C)
#define MPI2_CMD_BUF_POST_BASE_SYSTEM_ADDRESS_SPACE         (0x00)
#define MPI2_CMD_BUF_POST_BASE_IOCDDR_ADDRESS_SPACE         (0x04)
#define MPI2_CMD_BUF_POST_BASE_IOCPLB_ADDRESS_SPACE         (0x08)
#define MPI2_CMD_BUF_POST_BASE_IOCPLBNTA_ADDRESS_SPACE      (0x0C)

#define MPI2_CMD_BUF_POST_BASE_FLAGS_AUTO_POST_ALL          (0x01)


/****************************************************************************
*  Target Command Buffer Post List Request
****************************************************************************/

typedef struct _MPI2_TARGET_CMD_BUF_POST_LIST_REQUEST
{
    U16                     Reserved;               /* 0x00 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     CmdBufferCount;         /* 0x04 */
    U8                      Reserved1;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved2;              /* 0x0A */
```

```
    U32                     Reserved3;              /* 0x0C */
    U16                     IoIndex[2];             /* 0x10 */
} MPI2_TARGET_CMD_BUF_POST_LIST_REQUEST,
  MPI2_POINTER PTR_MPI2_TARGET_CMD_BUF_POST_LIST_REQUEST,
  Mpi2TargetCmdBufferPostListRequest_t,
  MPI2_POINTER pMpi2TargetCmdBufferPostListRequest_t;

/*****************************************************************************
*  Target Command Buffer Post Base List Reply
*****************************************************************************/

typedef struct _MPI2_TARGET_BUF_POST_BASE_LIST_REPLY
{
    U8                      Flags;                  /* 0x00 */
    U8                      Reserved;               /* 0x01 */
    U8                      MsgLength;              /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved1;              /* 0x04 */
    U8                      Reserved2;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved3;              /* 0x0A */
    U16                     Reserved4;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
    U16                     IoIndex;                /* 0x14 */
    U16                     Reserved5;              /* 0x16 */
    U32                     Reserved6;              /* 0x18 */
} MPI2_TARGET_BUF_POST_BASE_LIST_REPLY,
  MPI2_POINTER PTR_MPI2_TARGET_BUF_POST_BASE_LIST_REPLY,
  Mpi2TargetCmdBufferPostBaseListReply_t,
  MPI2_POINTER pMpi2TargetCmdBufferPostBaseListReply_t;

/* Flags defines */
#define MPI2_CMD_BUF_POST_REPLY_IOINDEX_VALID       (0x01)


/*****************************************************************************
*  Command Buffer Formats (with 16 byte CDB)
*****************************************************************************/

typedef struct _MPI2_TARGET_SSP_CMD_BUFFER
{
    U8      FrameType;                              /* 0x00 */
    U8      Reserved1;                              /* 0x01 */
    U16     InitiatorConnectionTag;                 /* 0x02 */
    U32     HashedSourceSASAddress;                 /* 0x04 */
    U16     Reserved2;                              /* 0x08 */
    U16     Flags;                                  /* 0x0A */
    U32     Reserved3;                              /* 0x0C */
    U16     Tag;                                    /* 0x10 */
    U16     TargetPortTransferTag;                  /* 0x12 */
    U32     DataOffset;                             /* 0x14 */
    /* COMMAND information unit starts here */
    U8      LogicalUnitNumber[8];                   /* 0x18 */
    U8      Reserved4;                              /* 0x20 */
    U8      TaskAttribute; /* lower 3 bits */       /* 0x21 */
    U8      Reserved5;                              /* 0x22 */
    U8      AdditionalCDBLength; /* upper 5 bits */ /* 0x23 */
    U8      CDB[16];                                /* 0x24 */
    /* Additional CDB bytes extend past the CDB field */
} MPI2_TARGET_SSP_CMD_BUFFER, MPI2_POINTER PTR_MPI2_TARGET_SSP_CMD_BUFFER,
  Mpi2TargetSspCmdBuffer, MPI2_POINTER pMp2iTargetSspCmdBuffer;

typedef struct _MPI2_TARGET_SSP_TASK_BUFFER
```



```
{
    U8      FrameType;                              /* 0x00 */
    U8      Reserved1;                              /* 0x01 */
    U16     InitiatorConnectionTag;                 /* 0x02 */
    U32     HashedSourceSASAddress;                 /* 0x04 */
    U16     Reserved2;                              /* 0x08 */
    U16     Flags;                                  /* 0x0A */
    U32     Reserved3;                              /* 0x0C */
    U16     Tag;                                    /* 0x10 */
    U16     TargetPortTransferTag;                  /* 0x12 */
    U32     DataOffset;                             /* 0x14 */
    /* TASK information unit starts here */
    U8      LogicalUnitNumber[8];                   /* 0x18 */
    U16     Reserved4;                              /* 0x20 */
    U8      TaskManagementFunction;                 /* 0x22 */
    U8      Reserved5;                              /* 0x23 */
    U16     ManagedTaskTag;                         /* 0x24 */
    U16     Reserved6;                              /* 0x26 */
    U32     Reserved7;                              /* 0x28 */
    U32     Reserved8;                              /* 0x2C */
    U32     Reserved9;                              /* 0x30 */
} MPI2_TARGET_SSP_TASK_BUFFER, MPI2_POINTER PTR_MPI2_TARGET_SSP_TASK_BUFFER,
  Mpi2TargetSspTaskBuffer, MPI2_POINTER pMpi2TargetSspTaskBuffer;

/* mask and shift for HashedSourceSASAddress field */
#define MPI2_TARGET_HASHED_SAS_ADDRESS_MASK     (0xFFFFFF00)
#define MPI2_TARGET_HASHED_SAS_ADDRESS_SHIFT    (8)


/*****************************************************************************
*   MPI v2.0 Target Assist Request
*****************************************************************************/

typedef struct _MPI2_TARGET_ASSIST_REQUEST
{
    U8                  Reserved1;                  /* 0x00 */
    U8                  TargetAssistFlags;          /* 0x01 */
    U8                  ChainOffset;                /* 0x02 */
    U8                  Function;                   /* 0x03 */
    U16                 QueueTag;                   /* 0x04 */
    U8                  Reserved2;                  /* 0x06 */
    U8                  MsgFlags;                   /* 0x07 */
    U8                  VP_ID;                      /* 0x08 */
    U8                  VF_ID;                      /* 0x09 */
    U16                 Reserved3;                  /* 0x0A */
    U16                 IoIndex;                    /* 0x0C */
    U16                 InitiatorConnectionTag;     /* 0x0E */
    U16                 SGLFlags;                   /* 0x10 */
    U8                  SequenceNumber;             /* 0x12 */
    U8                  Reserved4;                  /* 0x13 */
    U8                  SGLOffset0;                 /* 0x14 */
    U8                  SGLOffset1;                 /* 0x15 */
    U8                  SGLOffset2;                 /* 0x16 */
    U8                  SGLOffset3;                 /* 0x17 */
    U32                 SkipCount;                  /* 0x18 */
    U32                 DataLength;                 /* 0x1C */
    U32                 BidirectionalDataLength;    /* 0x20 */
    U16                 IoFlags;                    /* 0x24 */
    U16                 EEDPFlags;                  /* 0x26 */
    U32                 EEDPBlockSize;              /* 0x28 */
    U32                 SecondaryReferenceTag;      /* 0x2C */
    U16                 SecondaryApplicationTag;    /* 0x30 */
    U16                 ApplicationTagTranslationMask;  /* 0x32 */
    U32                 PrimaryReferenceTag;        /* 0x34 */
    U16                 PrimaryApplicationTag;      /* 0x38 */
    U16                 PrimaryApplicationTagMask;  /* 0x3A */
```

```
260     U32                 RelativeOffset;                 /* 0x3C */
261     U32                 Reserved5;                      /* 0x40 */
262     U32                 Reserved6;                      /* 0x44 */
263     U32                 Reserved7;                      /* 0x48 */
264     U32                 Reserved8;                      /* 0x4C */
265     MPI2_SGE_IO_UNION   SGL[1];                         /* 0x50 */
266 } MPI2_TARGET_ASSIST_REQUEST, MPI2_POINTER PTR_MPI2_TARGET_ASSIST_REQUEST,
267   Mpi2TargetAssistRequest_t, MPI2_POINTER pMpi2TargetAssistRequest_t;

269 /* Target Assist TargetAssistFlags bits */

271 #define MPI2_TARGET_ASSIST_FLAGS_REPOST_CMD_BUFFER      (0x80)
272 #define MPI2_TARGET_ASSIST_FLAGS_TLR                    (0x10)
273 #define MPI2_TARGET_ASSIST_FLAGS_RETRANSMIT             (0x04)
274 #define MPI2_TARGET_ASSIST_FLAGS_AUTO_STATUS            (0x02)
275 #define MPI2_TARGET_ASSIST_FLAGS_DATA_DIRECTION         (0x01)

277 /* Target Assist SGLFlags bits */

279 /* base values for Data Location Address Space */
280 #define MPI2_TARGET_ASSIST_SGLFLAGS_ADDR_MASK           (0x0C)
281 #define MPI2_TARGET_ASSIST_SGLFLAGS_SYSTEM_ADDR         (0x00)
282 #define MPI2_TARGET_ASSIST_SGLFLAGS_IOCDDR_ADDR         (0x04)
283 #define MPI2_TARGET_ASSIST_SGLFLAGS_IOCPLB_ADDR         (0x08)
284 #define MPI2_TARGET_ASSIST_SGLFLAGS_PLBNTA_ADDR         (0x0C)

286 /* base values for Type */
287 #define MPI2_TARGET_ASSIST_SGLFLAGS_TYPE_MASK           (0x03)
288 #define MPI2_TARGET_ASSIST_SGLFLAGS_MPI_TYPE            (0x00)
289 #define MPI2_TARGET_ASSIST_SGLFLAGS_32IEEE_TYPE         (0x01)
290 #define MPI2_TARGET_ASSIST_SGLFLAGS_64IEEE_TYPE         (0x02)

292 /* shift values for each sub-field */
293 #define MPI2_TARGET_ASSIST_SGLFLAGS_SGL3_SHIFT          (12)
294 #define MPI2_TARGET_ASSIST_SGLFLAGS_SGL2_SHIFT          (8)
295 #define MPI2_TARGET_ASSIST_SGLFLAGS_SGL1_SHIFT          (4)
296 #define MPI2_TARGET_ASSIST_SGLFLAGS_SGL0_SHIFT          (0)

298 /* Target Assist IoFlags bits */

300 #define MPI2_TARGET_ASSIST_IOFLAGS_BIDIRECTIONAL        (0x0800)
301 #define MPI2_TARGET_ASSIST_IOFLAGS_MULTICAST            (0x0400)
302 #define MPI2_TARGET_ASSIST_IOFLAGS_RECEIVE_FIRST        (0x0200)

304 /* Target Assist EEDPFlags bits */

306 #define MPI2_TA_EEDPFLAGS_INC_PRI_REFTAG            (0x8000)
307 #define MPI2_TA_EEDPFLAGS_INC_SEC_REFTAG            (0x4000)
308 #define MPI2_TA_EEDPFLAGS_INC_PRI_APPTAG            (0x2000)
309 #define MPI2_TA_EEDPFLAGS_INC_SEC_APPTAG            (0x1000)

311 #define MPI2_TA_EEDPFLAGS_CHECK_REFTAG              (0x0400)
312 #define MPI2_TA_EEDPFLAGS_CHECK_APPTAG              (0x0200)
313 #define MPI2_TA_EEDPFLAGS_CHECK_GUARD               (0x0100)

315 #define MPI2_TA_EEDPFLAGS_PASSTHRU_REFTAG           (0x0008)

317 #define MPI2_TA_EEDPFLAGS_MASK_OP                   (0x0007)
318 #define MPI2_TA_EEDPFLAGS_NOOP_OP                   (0x0000)
319 #define MPI2_TA_EEDPFLAGS_CHECK_OP                  (0x0001)
320 #define MPI2_TA_EEDPFLAGS_STRIP_OP                  (0x0002)
321 #define MPI2_TA_EEDPFLAGS_CHECK_REMOVE_OP           (0x0003)
322 #define MPI2_TA_EEDPFLAGS_INSERT_OP                 (0x0004)
323 #define MPI2_TA_EEDPFLAGS_REPLACE_OP                (0x0006)
324 #define MPI2_TA_EEDPFLAGS_CHECK_REGEN_OP            (0x0007)
```



```
327 /****************************************************************************
328 *   MPI v2.5 Target Assist Request
329 ****************************************************************************/

331 typedef struct _MPI25_TARGET_ASSIST_REQUEST
332 {
333     U8                  Reserved1;                      /* 0x00 */
334     U8                  TargetAssistFlags;              /* 0x01 */
335     U8                  ChainOffset;                    /* 0x02 */
336     U8                  Function;                       /* 0x03 */
337     U16                 QueueTag;                       /* 0x04 */
338     U8                  Reserved2;                      /* 0x06 */
339     U8                  MsgFlags;                       /* 0x07 */
340     U8                  VP_ID;                          /* 0x08 */
341     U8                  VF_ID;                          /* 0x09 */
342     U16                 Reserved3;                      /* 0x0A */
343     U16                 IoIndex;                        /* 0x0C */
344     U16                 InitiatorConnectionTag;         /* 0x0E */
345     U8                  DMAFlags;                       /* 0x10 */
346     U8                  Reserved9;                      /* 0x11 */
347     U8                  SequenceNumber;                 /* 0x12 */
348     U8                  Reserved4;                      /* 0x13 */
349     U8                  SGLOffset0;                     /* 0x14 */
350     U8                  SGLOffset1;                     /* 0x15 */
351     U8                  SGLOffset2;                     /* 0x16 */
352     U8                  SGLOffset3;                     /* 0x17 */
353     U32                 SkipCount;                      /* 0x18 */
354     U32                 DataLength;                     /* 0x1C */
355     U32                 BidirectionalDataLength;        /* 0x20 */
356     U16                 IoFlags;                        /* 0x24 */
357     U16                 EEDPFlags;                      /* 0x26 */
358     U16                 EEDPBlockSize;                  /* 0x28 */
359     U16                 Reserved10;                     /* 0x2A */
360     U32                 SecondaryReferenceTag;          /* 0x2C */
361     U16                 SecondaryApplicationTag;        /* 0x30 */
362     U16                 ApplicationTagTranslationMask;  /* 0x32 */
363     U32                 PrimaryReferenceTag;            /* 0x34 */
364     U16                 PrimaryApplicationTag;          /* 0x38 */
365     U16                 PrimaryApplicationTagMask;      /* 0x3A */
366     U32                 RelativeOffset;                 /* 0x3C */
367     U32                 Reserved5;                      /* 0x40 */
368     U32                 Reserved6;                      /* 0x44 */
369     U32                 Reserved7;                      /* 0x48 */
370     U32                 Reserved8;                      /* 0x4C */
371     MPI25_SGE_IO_UNION  SGL;                            /* 0x50 */
372 } MPI25_TARGET_ASSIST_REQUEST, MPI2_POINTER PTR_MPI25_TARGET_ASSIST_REQUEST,
373   Mpi25TargetAssistRequest_t, MPI2_POINTER pMpi25TargetAssistRequest_t;

375 /* use MPI2_TARGET_ASSIST_FLAGS_ defines for the Flags field */

377 /* Defines for the DMAFlags field
378  *  Each setting affects 4 SGLS, from SGL0 to SGL3.
379  *      D = Data
380  *      C = Cache DIF
381  *      I = Interleaved
382  *      H = Host DIF
383  */
384 #define MPI25_TA_DMAFLAGS_OP_MASK                   (0x0F)
385 #define MPI25_TA_DMAFLAGS_OP_D_D_D_D                (0x00)
386 #define MPI25_TA_DMAFLAGS_OP_D_D_D_C                (0x01)
387 #define MPI25_TA_DMAFLAGS_OP_D_D_D_I                (0x02)
388 #define MPI25_TA_DMAFLAGS_OP_D_D_C_C                (0x03)
389 #define MPI25_TA_DMAFLAGS_OP_D_D_C_I                (0x04)
390 #define MPI25_TA_DMAFLAGS_OP_D_D_I_I                (0x05)
391 #define MPI25_TA_DMAFLAGS_OP_D_C_C_C                (0x06)
```

```
392 #define MPI25_TA_DMAFLAGS_OP_D_C_C_I                (0x07)
393 #define MPI25_TA_DMAFLAGS_OP_D_C_I_I                (0x08)
394 #define MPI25_TA_DMAFLAGS_OP_D_I_I_I                (0x09)
395 #define MPI25_TA_DMAFLAGS_OP_D_H_D_D                (0x0A)
396 #define MPI25_TA_DMAFLAGS_OP_D_H_D_C                (0x0B)
397 #define MPI25_TA_DMAFLAGS_OP_D_H_D_I                (0x0C)
398 #define MPI25_TA_DMAFLAGS_OP_D_H_C_C                (0x0D)
399 #define MPI25_TA_DMAFLAGS_OP_D_H_C_I                (0x0E)
400 #define MPI25_TA_DMAFLAGS_OP_D_H_I_I                (0x0F)

402 /* defines for the IoFlags field */
403 #define MPI25_TARGET_ASSIST_IOFLAGS_BIDIRECTIONAL       (0x0800)
404 #define MPI25_TARGET_ASSIST_IOFLAGS_RECEIVE_FIRST       (0x0200)

406 /* defines for the EEDPFlags field */
407 #define MPI25_TA_EEDPFLAGS_INC_PRI_REFTAG               (0x8000)
408 #define MPI25_TA_EEDPFLAGS_INC_SEC_REFTAG               (0x4000)
409 #define MPI25_TA_EEDPFLAGS_INC_PRI_APPTAG               (0x2000)
410 #define MPI25_TA_EEDPFLAGS_INC_SEC_APPTAG               (0x1000)

412 #define MPI25_TA_EEDPFLAGS_CHECK_REFTAG                 (0x0400)
413 #define MPI25_TA_EEDPFLAGS_CHECK_APPTAG                 (0x0200)
414 #define MPI25_TA_EEDPFLAGS_CHECK_GUARD                  (0x0100)

416 #define MPI25_TA_EEDPFLAGS_ESCAPE_MODE_MASK             (0x00C0)
417 #define MPI25_TA_EEDPFLAGS_COMPATIBLE_MODE              (0x0000)
418 #define MPI25_TA_EEDPFLAGS_DO_NOT_DISABLE_MODE          (0x0040)
419 #define MPI25_TA_EEDPFLAGS_APPTAG_DISABLE_MODE          (0x0080)
420 #define MPI25_TA_EEDPFLAGS_APPTAG_REFTAG_DISABLE_MODE   (0x00C0)

422 #define MPI25_TA_EEDPFLAGS_HOST_GUARD_METHOD_MASK       (0x0030)
423 #define MPI25_TA_EEDPFLAGS_T10_CRC_HOST_GUARD           (0x0000)
424 #define MPI25_TA_EEDPFLAGS_IP_CHKSUM_HOST_GUARD         (0x0010)

426 #define MPI25_TA_EEDPFLAGS_PASSTHRU_REFTAG              (0x0008)

428 #define MPI25_TA_EEDPFLAGS_MASK_OP                      (0x0007)
429 #define MPI25_TA_EEDPFLAGS_NOOP_OP                      (0x0000)
430 #define MPI25_TA_EEDPFLAGS_CHECK_OP                     (0x0001)
431 #define MPI25_TA_EEDPFLAGS_STRIP_OP                     (0x0002)
432 #define MPI25_TA_EEDPFLAGS_CHECK_REMOVE_OP              (0x0003)
433 #define MPI25_TA_EEDPFLAGS_INSERT_OP                    (0x0004)
434 #define MPI25_TA_EEDPFLAGS_REPLACE_OP                   (0x0006)
435 #define MPI25_TA_EEDPFLAGS_CHECK_REGEN_OP               (0x0007)


438 /*****************************************************************************
439 *  Target Status Send Request
440 *****************************************************************************/

442 typedef struct _MPI2_TARGET_STATUS_SEND_REQUEST
443 {
444     U8                      Reserved1;              /* 0x00 */
445     U8                      StatusFlags;            /* 0x01 */
446     U8                      ChainOffset;            /* 0x02 */
447     U8                      Function;               /* 0x03 */
448     U16                     QueueTag;               /* 0x04 */
449     U8                      Reserved2;              /* 0x06 */
450     U8                      MsgFlags;               /* 0x07 */
451     U8                      VP_ID;                  /* 0x08 */
452     U8                      VF_ID;                  /* 0x09 */
453     U16                     Reserved3;              /* 0x0A */
454     U16                     IoIndex;                /* 0x0C */
455     U16                     InitiatorConnectionTag; /* 0x0E */
456     U16                     SGLFlags;               /* 0x10 */ /* MPI v2.0 o
457     U16                     Reserved4;              /* 0x12 */
```



```
458     U8                      SGLOffset0;             /* 0x14 */
459     U8                      Reserved5;              /* 0x15 */
460     U16                     Reserved6;              /* 0x16 */
461     U32                     Reserved7;              /* 0x18 */
462     U32                     Reserved8;              /* 0x1C */
463     MPI2_SIMPLE_SGE_UNION   StatusDataSGE;          /* 0x20 */ /* MPI v2.5:
464 } MPI2_TARGET_STATUS_SEND_REQUEST,
465   MPI2_POINTER PTR_MPI2_TARGET_STATUS_SEND_REQUEST,
466   Mpi2TargetStatusSendRequest_t, MPI2_POINTER pMpi2TargetStatusSendRequest_t;

468 /* Target Status Send StatusFlags bits */

470 #define MPI2_TSS_FLAGS_REPOST_CMD_BUFFER            (0x80)
471 #define MPI2_TSS_FLAGS_RETRANSMIT                   (0x04)
472 #define MPI2_TSS_FLAGS_AUTO_GOOD_STATUS             (0x01)

474 /* Target Status Send SGLFlags bits - MPI v2.0 only */
475 /* Data Location Address Space */
476 #define MPI2_TSS_SGLFLAGS_ADDR_MASK                 (0x0C)
477 #define MPI2_TSS_SGLFLAGS_SYSTEM_ADDR               (0x00)
478 #define MPI2_TSS_SGLFLAGS_IOCDDR_ADDR               (0x04)
479 #define MPI2_TSS_SGLFLAGS_IOCPLB_ADDR               (0x08)
480 #define MPI2_TSS_SGLFLAGS_IOCPLBNTA_ADDR            (0x0C)
481 /* Type */
482 #define MPI2_TSS_SGLFLAGS_TYPE_MASK                 (0x03)
483 #define MPI2_TSS_SGLFLAGS_MPI_TYPE                  (0x00)
484 #define MPI2_TSS_SGLFLAGS_IEEE32_TYPE               (0x01)
485 #define MPI2_TSS_SGLFLAGS_IEEE64_TYPE               (0x02)



489 /*
490  * NOTE: The SSP status IU is big-endian. When used on a little-endian system,
491  * this structure properly orders the bytes.
492  */
493 typedef struct _MPI2_TARGET_SSP_RSP_IU
494 {
495     U32     Reserved0[6]; /* reserved for SSP header */ /* 0x00 */

497     /* start of RESPONSE information unit */
498     U32     Reserved1;                                  /* 0x18 */
499     U32     Reserved2;                                  /* 0x1C */
500     U16     Reserved3;                                  /* 0x20 */
501     U8      DataPres; /* lower 2 bits */                /* 0x22 */
502     U8      Status;                                     /* 0x23 */
503     U32     Reserved4;                                  /* 0x24 */
504     U32     SenseDataLength;                            /* 0x28 */
505     U32     ResponseDataLength;                         /* 0x2C */

507     /* start of Response or Sense Data (size may vary dynamically) */
508     U8      ResponseSenseData[4];                       /* 0x30 */
509 } MPI2_TARGET_SSP_RSP_IU, MPI2_POINTER PTR_MPI2_TARGET_SSP_RSP_IU,
510   Mpi2TargetSspRspIu_t, MPI2_POINTER pMpi2TargetSspRspIu_t;


513 /*****************************************************************************
514 *  Target Standard Reply - used with Target Assist or Target Status Send
515 *****************************************************************************/

517 typedef struct _MPI2_TARGET_STANDARD_REPLY
518 {
519     U16                     Reserved;               /* 0x00 */
520     U8                      MsgLength;              /* 0x02 */
521     U8                      Function;               /* 0x03 */
522     U16                     Reserved1;              /* 0x04 */
523     U8                      Reserved2;              /* 0x06 */
```

```
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved3;              /* 0x0A */
    U16                     Reserved4;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
    U16                     IoIndex;                /* 0x14 */
    U16                     Reserved5;              /* 0x16 */
    U32                     TransferCount;          /* 0x18 */
    U32                     BidirectionalTransferCount; /* 0x1C */
} MPI2_TARGET_STANDARD_REPLY, MPI2_POINTER PTR_MPI2_TARGET_STANDARD_REPLY,
  Mpi2TargetErrorReply_t, MPI2_POINTER pMpi2TargetErrorReply_t;


/*****************************************************************************
*  Target Mode Abort Request
*****************************************************************************/

typedef struct _MPI2_TARGET_MODE_ABORT_REQUEST
{
    U8                      AbortType;              /* 0x00 */
    U8                      Reserved1;              /* 0x01 */
    U8                      ChainOffset;            /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved2;              /* 0x04 */
    U8                      Reserved3;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved4;              /* 0x0A */
    U16                     IoIndexToAbort;         /* 0x0C */
    U16                     InitiatorDevHandle;     /* 0x0E */
    U32                     MidToAbort;             /* 0x10 */
} MPI2_TARGET_MODE_ABORT, MPI2_POINTER PTR_MPI2_TARGET_MODE_ABORT,
  Mpi2TargetModeAbort_t, MPI2_POINTER pMpi2TargetModeAbort_t;

/* Target Mode Abort AbortType values */

#define MPI2_TARGET_MODE_ABORT_ALL_CMD_BUFFERS      (0x00)
#define MPI2_TARGET_MODE_ABORT_ALL_IO               (0x01)
#define MPI2_TARGET_MODE_ABORT_EXACT_IO             (0x02)
#define MPI2_TARGET_MODE_ABORT_EXACT_IO_REQUEST     (0x03)
#define MPI2_TARGET_MODE_ABORT_IO_REQUEST_AND_IO    (0x04)
#define MPI2_TARGET_MODE_ABORT_DEVHANDLE            (0x05)
#define MPI2_TARGET_MODE_ABORT_ALL_COMMANDS         (0x06)


/*****************************************************************************
*  Target Mode Abort Reply
*****************************************************************************/

typedef struct _MPI2_TARGET_MODE_ABORT_REPLY
{
    U16                     Reserved;               /* 0x00 */
    U8                      MsgLength;              /* 0x02 */
    U8                      Function;               /* 0x03 */
    U16                     Reserved1;              /* 0x04 */
    U8                      Reserved2;              /* 0x06 */
    U8                      MsgFlags;               /* 0x07 */
    U8                      VP_ID;                  /* 0x08 */
    U8                      VF_ID;                  /* 0x09 */
    U16                     Reserved3;              /* 0x0A */
    U16                     Reserved4;              /* 0x0C */
    U16                     IOCStatus;              /* 0x0E */
    U32                     IOCLogInfo;             /* 0x10 */
```



```
    U32                     AbortCount;             /* 0x14 */
} MPI2_TARGET_MODE_ABORT_REPLY, MPI2_POINTER PTR_MPI2_TARGET_MODE_ABORT_REPLY,
  Mpi2TargetModeAbortReply_t, MPI2_POINTER pMpi2TargetModeAbortReply_t;


#endif
```

```
**********************************************************
   27717 Mon Jun 16 21:18:08 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_tool.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2013 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_tool.h
 *         Title:  MPI diagnostic tool structures and definitions
 *  Creation Date:  March 26, 2007
 *
 *    mpi2_tool.h Version:  02.00.11
 *    mpi2_tool.h Version:  02.00.04
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  12-18-07  02.00.01  Added Diagnostic Buffer Post and Diagnostic Release
 *                      structures and defines.
 *  02-29-08  02.00.02  Modified various names to make them 32-character unique.
 *  05-06-09  02.00.03  Added ISTWI Read Write Tool and Diagnostic CLI Tool.
 *  07-30-09  02.00.04  Added ExtendedType field to DiagnosticBufferPost request
 *                      and reply messages.
 *                      Added MPI2_DIAG_BUF_TYPE_EXTENDED.
 *                      Incremented MPI2_DIAG_BUF_TYPE_COUNT.
 *  05-12-10  02.00.05  Added Diagnostic Data Upload tool.
 *  08-11-10  02.00.06  Added defines that were missing for Diagnostic Buffer
 *                      Post Request.
 *  05-25-11  02.00.07  Added Flags field and related defines to
 *                      MPI2_TOOLBOX_ISTWI_READ_WRITE_REQUEST.
 *  11-18-11  02.00.08  Incorporating additions for MPI v2.5.
 *  07-10-12  02.00.09  Add MPI v2.5 Toolbox Diagnostic CLI Tool Request
 *                      message.
 *  07-26-12  02.00.10  Modified MPI2_TOOLBOX_DIAGNOSTIC_CLI_REQUEST so that
 *                      it uses MPI Chain SGE as well as MPI Simple SGE.
 *  08-19-13  02.00.11  Added MPI2_TOOLBOX_TEXT_DISPLAY_TOOL and related info.
 *  --------------------------------------------------------------------------
 */

#ifndef MPI2_TOOL_H
#define MPI2_TOOL_H

/*****************************************************************************
 *
 *               Toolbox Messages
 *
 *****************************************************************************/

/* defines for the Tools */
#define MPI2_TOOLBOX_CLEAN_TOOL                     (0x00)
#define MPI2_TOOLBOX_MEMORY_MOVE_TOOL               (0x01)
#define MPI2_TOOLBOX_DIAG_DATA_UPLOAD_TOOL          (0x02)
#define MPI2_TOOLBOX_ISTWI_READ_WRITE_TOOL          (0x03)
#define MPI2_TOOLBOX_BEACON_TOOL                    (0x05)
#define MPI2_TOOLBOX_DIAGNOSTIC_CLI_TOOL            (0x06)
#define MPI2_TOOLBOX_TEXT_DISPLAY_TOOL              (0x07)
```

```
 87 /*****************************************************************************
 88 *  Toolbox reply
 89 *****************************************************************************/

 91 typedef struct _MPI2_TOOLBOX_REPLY
 92 {
 93     U8                      Tool;                   /* 0x00 */
 94     U8                      Reserved1;              /* 0x01 */
 95     U8                      MsgLength;              /* 0x02 */
 96     U8                      Function;               /* 0x03 */
 97     U16                     Reserved2;              /* 0x04 */
 98     U8                      Reserved3;              /* 0x06 */
 99     U8                      MsgFlags;               /* 0x07 */
100     U8                      VP_ID;                  /* 0x08 */
101     U8                      VF_ID;                  /* 0x09 */
102     U16                     Reserved4;              /* 0x0A */
103     U16                     Reserved5;              /* 0x0C */
104     U16                     IOCStatus;              /* 0x0E */
105     U32                     IOCLogInfo;             /* 0x10 */
106 } MPI2_TOOLBOX_REPLY, MPI2_POINTER PTR_MPI2_TOOLBOX_REPLY,
_____unchanged_portion_omitted_
161   Mpi2ToolboxMemMoveRequest_t, MPI2_POINTER pMpi2ToolboxMemMoveRequest_t;


164 /*****************************************************************************
165 *  Toolbox Diagnostic Data Upload request
166 *****************************************************************************/

168 typedef struct _MPI2_TOOLBOX_DIAG_DATA_UPLOAD_REQUEST
169 {
170     U8                      Tool;                   /* 0x00 */
171     U8                      Reserved1;              /* 0x01 */
172     U8                      ChainOffset;            /* 0x02 */
173     U8                      Function;               /* 0x03 */
174     U16                     Reserved2;              /* 0x04 */
175     U8                      Reserved3;              /* 0x06 */
176     U8                      MsgFlags;               /* 0x07 */
177     U8                      VP_ID;                  /* 0x08 */
178     U8                      VF_ID;                  /* 0x09 */
179     U16                     Reserved4;              /* 0x0A */
180     U8                      SGLFlags;               /* 0x0C */
181     U8                      Reserved5;              /* 0x0D */
182     U16                     Reserved6;              /* 0x0E */
183     U32                     Flags;                  /* 0x10 */
184     U32                     DataLength;             /* 0x14 */
185     MPI2_SGE_SIMPLE_UNION   SGL;                    /* 0x18 */
186 } MPI2_TOOLBOX_DIAG_DATA_UPLOAD_REQUEST,
187   MPI2_POINTER PTR_MPI2_TOOLBOX_DIAG_DATA_UPLOAD_REQUEST,
188   Mpi2ToolboxDiagDataUploadRequest_t,
189   MPI2_POINTER pMpi2ToolboxDiagDataUploadRequest_t;

191 /* use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */


194 typedef struct _MPI2_DIAG_DATA_UPLOAD_HEADER
195 {
196     U32                     DiagDataLength;         /* 00h */
197     U8                      FormatCode;             /* 04h */
198     U8                      Reserved1;              /* 05h */
199     U16                     Reserved2;              /* 06h */
200 } MPI2_DIAG_DATA_UPLOAD_HEADER, MPI2_POINTER PTR_MPI2_DIAG_DATA_UPLOAD_HEADER,
201   Mpi2DiagDataUploadHeader_t, MPI2_POINTER pMpi2DiagDataUploadHeader_t;
```



```
204 /*****************************************************************************
205 *  Toolbox ISTWI Read Write Tool
206 *****************************************************************************/

208 /* Toolbox ISTWI Read Write Tool request message */
209 typedef struct _MPI2_TOOLBOX_ISTWI_READ_WRITE_REQUEST
210 {
211     U8                      Tool;                   /* 0x00 */
212     U8                      Reserved1;              /* 0x01 */
213     U8                      ChainOffset;            /* 0x02 */
214     U8                      Function;               /* 0x03 */
215     U16                     Reserved2;              /* 0x04 */
216     U8                      Reserved3;              /* 0x06 */
217     U8                      MsgFlags;               /* 0x07 */
218     U8                      VP_ID;                  /* 0x08 */
219     U8                      VF_ID;                  /* 0x09 */
220     U16                     Reserved4;              /* 0x0A */
221     U32                     Reserved5;              /* 0x0C */
222     U32                     Reserved6;              /* 0x10 */
223     U8                      DevIndex;               /* 0x14 */
224     U8                      Action;                 /* 0x15 */
225     U8                      SGLFlags;               /* 0x16 */
226     U8                      Flags;                  /* 0x17 */
189     U8                      Reserved7;              /* 0x17 */
227     U16                     TxDataLength;           /* 0x18 */
228     U16                     RxDataLength;           /* 0x1A */
229     U32                     Reserved8;              /* 0x1C */
230     U32                     Reserved9;              /* 0x20 */
231     U32                     Reserved10;             /* 0x24 */
232     U32                     Reserved11;             /* 0x28 */
233     U32                     Reserved12;             /* 0x2C */
234     MPI2_SGE_SIMPLE_UNION   SGL;                    /* 0x30 */
235 } MPI2_TOOLBOX_ISTWI_READ_WRITE_REQUEST,
236   MPI2_POINTER PTR_MPI2_TOOLBOX_ISTWI_READ_WRITE_REQUEST,
237   Mpi2ToolboxIstwiReadWriteRequest_t,
238   MPI2_POINTER pMpi2ToolboxIstwiReadWriteRequest_t;

240 /* values for the Action field */
241 #define MPI2_TOOL_ISTWI_ACTION_READ_DATA            (0x01)
242 #define MPI2_TOOL_ISTWI_ACTION_WRITE_DATA           (0x02)
243 #define MPI2_TOOL_ISTWI_ACTION_SEQUENCE             (0x03)
244 #define MPI2_TOOL_ISTWI_ACTION_RESERVE_BUS          (0x10)
245 #define MPI2_TOOL_ISTWI_ACTION_RELEASE_BUS          (0x11)
246 #define MPI2_TOOL_ISTWI_ACTION_RESET                (0x12)

248 /* use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */
211 /* values for SGLFlags field are in the SGL section of mpi2.h */

250 /* values for the Flags field */
251 #define MPI2_TOOL_ISTWI_FLAG_AUTO_RESERVE_RELEASE   (0x80)
252 #define MPI2_TOOL_ISTWI_FLAG_PAGE_ADDR_MASK         (0x07)


255 /* Toolbox ISTWI Read Write Tool reply message */
256 typedef struct _MPI2_TOOLBOX_ISTWI_REPLY
257 {
258     U8                      Tool;                   /* 0x00 */
259     U8                      Reserved1;              /* 0x01 */
260     U8                      MsgLength;              /* 0x02 */
261     U8                      Function;               /* 0x03 */
262     U16                     Reserved2;              /* 0x04 */
263     U8                      Reserved3;              /* 0x06 */
264     U8                      MsgFlags;               /* 0x07 */
265     U8                      VP_ID;                  /* 0x08 */
266     U8                      VF_ID;                  /* 0x09 */
267     U16                     Reserved4;              /* 0x0A */
```

```
268     U16                     Reserved5;                  /* 0x0C */
269     U16                     IOCStatus;                  /* 0x0E */
270     U32                     IOCLogInfo;                 /* 0x10 */
271     U8                      DevIndex;                   /* 0x14 */
272     U8                      Action;                     /* 0x15 */
273     U8                      IstwiStatus;                /* 0x16 */
274     U8                      Reserved6;                  /* 0x17 */
275     U16                     TxDataCount;                /* 0x18 */
276     U16                     RxDataCount;                /* 0x1A */
277 } MPI2_TOOLBOX_ISTWI_REPLY, MPI2_POINTER PTR_MPI2_TOOLBOX_ISTWI_REPLY,
      unchanged_portion_omitted_
302   Mpi2ToolboxBeaconRequest_t, MPI2_POINTER pMpi2ToolboxBeaconRequest_t;

304 /* values for the Flags field */
305 #define MPI2_TOOLBOX_FLAGS_BEACONMODE_OFF       (0x00)
306 #define MPI2_TOOLBOX_FLAGS_BEACONMODE_ON        (0x01)


309 /*****************************************************************************
310 *  Toolbox Diagnostic CLI Tool
311 *****************************************************************************/

313 #define MPI2_TOOLBOX_DIAG_CLI_CMD_LENGTH    (0x5C)

315 /* MPI v2.0 Toolbox Diagnostic CLI Tool request message */
274 /* Toolbox Diagnostic CLI Tool request message */
316 typedef struct _MPI2_TOOLBOX_DIAGNOSTIC_CLI_REQUEST
317 {
318     U8                      Tool;                       /* 0x00 */
319     U8                      Reserved1;                  /* 0x01 */
320     U8                      ChainOffset;                /* 0x02 */
321     U8                      Function;                   /* 0x03 */
322     U16                     Reserved2;                  /* 0x04 */
323     U8                      Reserved3;                  /* 0x06 */
324     U8                      MsgFlags;                   /* 0x07 */
325     U8                      VP_ID;                      /* 0x08 */
326     U8                      VF_ID;                      /* 0x09 */
327     U16                     Reserved4;                  /* 0x0A */
328     U8                      SGLFlags;                   /* 0x0C */
329     U8                      Reserved5;                  /* 0x0D */
330     U16                     Reserved6;                  /* 0x0E */
331     U32                     DataLength;                 /* 0x10 */
332     U8                      DiagnosticCliCommand[MPI2_TOOLBOX_DIAG_CLI_CMD_LENGT
333     MPI2_MPI_SGE_IO_UNION   SGL;                        /* 0x70 */
292     MPI2_SGE_SIMPLE_UNION   SGL;                        /* 0x70 */
334 } MPI2_TOOLBOX_DIAGNOSTIC_CLI_REQUEST,
335   MPI2_POINTER PTR_MPI2_TOOLBOX_DIAGNOSTIC_CLI_REQUEST,
336   Mpi2ToolboxDiagnosticCliRequest_t,
337   MPI2_POINTER pMpi2ToolboxDiagnosticCliRequest_t;

339 /* use MPI2_SGLFLAGS_ defines from mpi2.h for the SGLFlags field */
298 /* values for SGLFlags field are in the SGL section of mpi2.h */


342 /* MPI v2.5 Toolbox Diagnostic CLI Tool request message */
343 typedef struct _MPI25_TOOLBOX_DIAGNOSTIC_CLI_REQUEST
344 {
345     U8                      Tool;                       /* 0x00 */
346     U8                      Reserved1;                  /* 0x01 */
347     U8                      ChainOffset;                /* 0x02 */
348     U8                      Function;                   /* 0x03 */
349     U16                     Reserved2;                  /* 0x04 */
350     U8                      Reserved3;                  /* 0x06 */
351     U8                      MsgFlags;                   /* 0x07 */
352     U8                      VP_ID;                      /* 0x08 */
353     U8                      VF_ID;                      /* 0x09 */
```



```
354     U16                     Reserved4;                  /* 0x0A */
355     U32                     Reserved5;                  /* 0x0C */
356     U32                     DataLength;                 /* 0x10 */
357     U8                      DiagnosticCliCommand[MPI2_TOOLBOX_DIAG_CLI_CMD_LENGT
358     MPI25_SGE_IO_UNION      SGL;                        /* 0x70 */
359 } MPI25_TOOLBOX_DIAGNOSTIC_CLI_REQUEST,
360   MPI2_POINTER PTR_MPI25_TOOLBOX_DIAGNOSTIC_CLI_REQUEST,
361   Mpi25ToolboxDiagnosticCliRequest_t,
362   MPI2_POINTER pMpi25ToolboxDiagnosticCliRequest_t;


365 /* Toolbox Diagnostic CLI Tool reply message */
366 typedef struct _MPI2_TOOLBOX_DIAGNOSTIC_CLI_REPLY
367 {
368     U8                      Tool;                       /* 0x00 */
369     U8                      Reserved1;                  /* 0x01 */
370     U8                      MsgLength;                  /* 0x02 */
371     U8                      Function;                   /* 0x03 */
372     U16                     Reserved2;                  /* 0x04 */
373     U8                      Reserved3;                  /* 0x06 */
374     U8                      MsgFlags;                   /* 0x07 */
375     U8                      VP_ID;                      /* 0x08 */
376     U8                      VF_ID;                      /* 0x09 */
377     U16                     Reserved4;                  /* 0x0A */
378     U16                     Reserved5;                  /* 0x0C */
379     U16                     IOCStatus;                  /* 0x0E */
380     U32                     IOCLogInfo;                 /* 0x10 */
381     U32                     ReturnedDataLength;         /* 0x14 */
382 } MPI2_TOOLBOX_DIAGNOSTIC_CLI_REPLY,
383   MPI2_POINTER PTR_MPI2_TOOLBOX_DIAG_CLI_REPLY,
384   Mpi2ToolboxDiagnosticCliReply_t,
385   MPI2_POINTER pMpi2ToolboxDiagnosticCliReply_t;


388 /*****************************************************************************
389 *  Toolbox Console Text Display Tool
390 *****************************************************************************/

392 /* Toolbox Console Text Display Tool request message */
393 typedef struct _MPI2_TOOLBOX_TEXT_DISPLAY_REQUEST
394 {
395     U8                      Tool;               /* 0x00 */
396     U8                      Reserved1;          /* 0x01 */
397     U8                      ChainOffset;        /* 0x02 */
398     U8                      Function;           /* 0x03 */
399     U16                     Reserved2;          /* 0x04 */
400     U8                      Reserved3;          /* 0x06 */
401     U8                      MsgFlags;           /* 0x07 */
402     U8                      VP_ID;              /* 0x08 */
403     U8                      VF_ID;              /* 0x09 */
404     U16                     Reserved4;          /* 0x0A */
405     U8                      Console;            /* 0x0C */
406     U8                      Flags;              /* 0x0D */
407     U16                     Reserved6;          /* 0x0E */
408     U8                      TextToDisplay[4];   /* 0x10 */ /* actual length dete
409 } MPI2_TOOLBOX_TEXT_DISPLAY_REQUEST,
410   MPI2_POINTER PTR_MPI2_TOOLBOX_TEXT_DISPLAY_REQUEST,
411   Mpi2ToolboxTextDisplayRequest_t,
412   MPI2_POINTER pMpi2ToolboxTextDisplayRequest_t;

414 /* defines for the Console field */
415 #define MPI2_TOOLBOX_CONSOLE_TYPE_MASK          (0xF0)
416 #define MPI2_TOOLBOX_CONSOLE_TYPE_DEFAULT       (0x00)
417 #define MPI2_TOOLBOX_CONSOLE_TYPE_UART          (0x10)
418 #define MPI2_TOOLBOX_CONSOLE_TYPE_ETHERNET      (0x20)
```

```
#define MPI2_TOOLBOX_CONSOLE_NUMBER_MASK        (0x0F)

/* defines for the Flags field */
#define MPI2_TOOLBOX_CONSOLE_FLAG_TIMESTAMP     (0x01)



/*****************************************************************************
*
*       Diagnostic Buffer Messages
*
*****************************************************************************/


/*****************************************************************************
*  Diagnostic Buffer Post request
*****************************************************************************/

typedef struct _MPI2_DIAG_BUFFER_POST_REQUEST
{
    U8                      ExtendedType;               /* 0x00 */
    U8                      BufferType;                 /* 0x01 */
    U8                      ChainOffset;                /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved4;                  /* 0x0A */
    U64                     BufferAddress;              /* 0x0C */
    U32                     BufferLength;               /* 0x14 */
    U32                     Reserved5;                  /* 0x18 */
    U32                     Reserved6;                  /* 0x1C */
    U32                     Flags;                      /* 0x20 */
    U32                     ProductSpecific[23];        /* 0x24 */
} MPI2_DIAG_BUFFER_POST_REQUEST, MPI2_POINTER PTR_MPI2_DIAG_BUFFER_POST_REQUEST,
  Mpi2DiagBufferPostRequest_t, MPI2_POINTER pMpi2DiagBufferPostRequest_t;

/* values for the ExtendedType field */
#define MPI2_DIAG_EXTENDED_TYPE_UTILIZATION         (0x02)

/* values for the BufferType field */
#define MPI2_DIAG_BUF_TYPE_TRACE                    (0x00)
#define MPI2_DIAG_BUF_TYPE_SNAPSHOT                 (0x01)
#define MPI2_DIAG_BUF_TYPE_EXTENDED                 (0x02)
/* count of the number of buffer types */
#define MPI2_DIAG_BUF_TYPE_COUNT                    (0x03)

/* values for the Flags field */
#define MPI2_DIAG_BUF_FLAG_RELEASE_ON_FULL          (0x00000002) /* for MPI v2.0
#define MPI2_DIAG_BUF_FLAG_IMMEDIATE_RELEASE        (0x00000001)


/*****************************************************************************
*  Diagnostic Buffer Post reply
*****************************************************************************/

typedef struct _MPI2_DIAG_BUFFER_POST_REPLY
{
    U8                      ExtendedType;               /* 0x00 */
    U8                      BufferType;                 /* 0x01 */
    U8                      MsgLength;                  /* 0x02 */
    U8                      Function;                   /* 0x03 */
    U16                     Reserved2;                  /* 0x04 */
    U8                      Reserved3;                  /* 0x06 */
```



```
    U8                      MsgFlags;                   /* 0x07 */
    U8                      VP_ID;                      /* 0x08 */
    U8                      VF_ID;                      /* 0x09 */
    U16                     Reserved4;                  /* 0x0A */
    U16                     Reserved5;                  /* 0x0C */
    U16                     IOCStatus;                  /* 0x0E */
    U32                     IOCLogInfo;                 /* 0x10 */
    U32                     TransferLength;             /* 0x14 */
} MPI2_DIAG_BUFFER_POST_REPLY, MPI2_POINTER PTR_MPI2_DIAG_BUFFER_POST_REPLY,
_____unchanged_portion_omitted_
```

```
**********************************************************
    3757 Mon Jun 16 21:18:08 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mpi/mpi2_type.h
NEX-1888 upstream
**********************************************************
```

```
/*-
 * Copyright (c) 2013 LSI Corp.
 * All rights reserved.
/*
 * CDDL HEADER START
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the author nor the names of any co-contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 *  Copyright (c) 2000-2007 LSI Corporation.
 * Copyright (c) 2000 to 2009, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```



```
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

/*
 *          Name:  mpi2_type.h
 *         Title:  MPI basic type definitions
 *  Creation Date:  August 16, 2006
 *
 *    mpi2_type.h Version:  02.00.00
 *
 *  Version History
 *  ---------------
 *
 *  Date      Version   Description
 *  --------  --------  ------------------------------------------------------
 *  04-30-07  02.00.00  Corresponds to Fusion-MPT MPI Specification Rev A.
 *  ------------------------------------------------------------------------
 */

#ifndef MPI2_TYPE_H
#define MPI2_TYPE_H


/*****************************************************************************
 * Define MPI2_POINTER if it hasn't already been defined. By default
 * MPI2_POINTER is defined to be a near pointer. MPI2_POINTER can be defined as
 * a far pointer by defining MPI2_POINTER as "far *" before this header file is
 * included.
 */
#ifndef MPI2_POINTER
#define MPI2_POINTER     *
#endif

/* the basic types may have already been included by mpi_type.h */
#ifndef MPI_TYPE_H
/*****************************************************************************
*
*               Basic Types
*
*****************************************************************************/

typedef signed   char   S8;
typedef unsigned char   U8;
typedef signed   short  S16;
typedef unsigned short  U16;

#ifdef  __FreeBSD__

typedef int32_t  S32;
typedef uint32_t U32;

#else

#if defined(unix) || defined(__arm) || defined(ALPHA) || defined(__PPC__) || def

    typedef signed   int   S32;
```

```
  86     typedef unsigned int   U32;

  88 #else

  90     typedef signed   long  S32;
  91     typedef unsigned long  U32;

  93 #endif
  94 #endif


  96 typedef struct _S64
  97 {
  98     U32          Low;
  99     S32          High;
 100 } S64;
```

*unchanged_portion_omitted_*

```
**********************************************************
    9514 Mon Jun 16 21:18:08 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mptsas_ioctl.h
NEX-1889 upstream
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
 */

/*
 * Copyright (c) 2000 to 2010, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

#ifndef _MPTSAS_IOCTL_H
#define _MPTSAS_IOCTL_H

#ifdef  __cplusplus
```



```
extern "C" {
#endif

#include <sys/types.h>

#define MPTIOCTL                        ('I' << 8)
#define MPTIOCTL_GET_ADAPTER_DATA       (MPTIOCTL | 1)
#define MPTIOCTL_UPDATE_FLASH           (MPTIOCTL | 2)
#define MPTIOCTL_RESET_ADAPTER          (MPTIOCTL | 3)
#define MPTIOCTL_PASS_THRU              (MPTIOCTL | 4)
#define MPTIOCTL_EVENT_QUERY            (MPTIOCTL | 5)
#define MPTIOCTL_EVENT_ENABLE           (MPTIOCTL | 6)
#define MPTIOCTL_EVENT_REPORT           (MPTIOCTL | 7)
#define MPTIOCTL_GET_PCI_INFO           (MPTIOCTL | 8)
#define MPTIOCTL_DIAG_ACTION            (MPTIOCTL | 9)
#define MPTIOCTL_REG_ACCESS             (MPTIOCTL | 10)
#define MPTIOCTL_GET_DISK_INFO          (MPTIOCTL | 11)
#define MPTIOCTL_LED_CONTROL            (MPTIOCTL | 12)

/*
 *  The following are our ioctl() return status values.  If everything went
 *  well, we return good status.  If the buffer length sent to us is too short
 *  we return a status to tell the user.
 */
#define MPTIOCTL_STATUS_GOOD            0
#define MPTIOCTL_STATUS_LEN_TOO_SHORT   1

typedef struct mptsas_pci_bits
{
        union {
                struct {
                        uint32_t        DeviceNumber    :5;
                        uint32_t        FunctionNumber  :3;
                        uint32_t        BusNumber       :24;
                } bits;
                uint32_t        AsDWORD;
        } u;
        uint32_t        PciSegmentId;
} mptsas_pci_bits_t;
/*
 *  The following is the MPTIOCTL_GET_ADAPTER_DATA data structure.  This data
 *  structure is setup so that we hopefully are properly aligned for both
 *  32-bit and 64-bit mode applications.
 *
 *  Adapter Type - Value = 4 = SCSI Protocol through SAS-2 adapter
 *                 Value = 6 = SCSI Protocol through SAS-3 adapter
 *
 *  MPI Port Number - The PCI Function number for this device
 *
 *  PCI Device HW Id - The PCI device number for this device
 *
 */
#define MPTIOCTL_ADAPTER_TYPE_SAS2      4
#define MPTIOCTL_ADAPTER_TYPE_SAS3      6

typedef struct mptsas_adapter_data
{
        uint32_t                StructureLength;
        uint32_t                AdapterType;
        uint32_t                MpiPortNumber;
        uint32_t                PCIDeviceHwId;
        uint32_t                PCIDeviceHwRev;
        uint32_t                SubSystemId;
        uint32_t                SubsystemVendorId;
        uint32_t                Reserved1;
        uint32_t                MpiFirmwareVersion;
```

```
128         uint32_t                BiosVersion;
129         uint8_t                 DriverVersion[32];
130         uint8_t                 Reserved2;
131         uint8_t                 ScsiId;
132         uint16_t                Reserved3;
133         mptsas_pci_bits_t       PciInformation;
134 } mptsas_adapter_data_t;
```

unchanged_portion_omitted_

```
**********************************************************
   45097 Mon Jun 16 21:18:08 2014
new/usr/src/uts/common/sys/scsi/adapters/mpt_sas/mptsas_var.h
NEX-1889 upstream
**********************************************************
```

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

/*
 * Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
 * Copyright (c) 2014, Tegile Systems Inc. All rights reserved.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 */

/*
 * Copyright (c) 2000 to 2010, LSI Corporation.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms of all code within
 * this file that is exclusively owned by LSI, with or without
 * modification, is permitted provided that, in addition to the CDDL 1.0
 * License requirements, the following conditions are met:
 *
 *    Neither the name of the author nor the names of its contributors may be
 *    used to endorse or promote products derived from this software without
 *    specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
 * DAMAGE.
 */

#ifndef _SYS_SCSI_ADAPTERS_MPTVAR_H
#define _SYS_SCSI_ADAPTERS_MPTVAR_H

#include <sys/byteorder.h>
#include <sys/queue.h>
```



```
#include <sys/isa_defs.h>
#include <sys/sunmdi.h>
#include <sys/mdi_impldefs.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_hash.h>
#include <sys/scsi/adapters/mpt_sas/mptsas_ioctl.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_tool.h>
#include <sys/scsi/adapters/mpt_sas/mpi/mpi2_cnfg.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * Compile options
 */
#ifdef DEBUG
#define MPTSAS_DEBUG            /* turn on debugging code */
#endif  /* DEBUG */

#define MPTSAS_INITIAL_SOFT_SPACE       4

#define MAX_MPI_PORTS           16

/*
 * Note below macro definition and data type definition
 * are used for phy mask handling, it should be changed
 * simultaneously.
 */
#define MPTSAS_MAX_PHYS         16
typedef uint16_t                mptsas_phymask_t;

#define MPTSAS_INVALID_DEVHDL   0xffff
#define MPTSAS_SATA_GUID        "sata-guid"

/*
 * Hash table sizes for SMP targets (i.e., expanders) and ordinary SSP/STP
 * targets.  There's no need to go overboard here, as the ordinary paths for
 * I/O do not normally require hashed target lookups.  These should be good
 * enough and then some for any fabric within the hardware's capabilities.
 */
#define MPTSAS_SMP_BUCKET_COUNT         23
#define MPTSAS_TARGET_BUCKET_COUNT      97

/*
 * MPT HW defines
 */
#define MPTSAS_MAX_DISKS_IN_CONFIG      14
#define MPTSAS_MAX_DISKS_IN_VOL         10
#define MPTSAS_MAX_HOTSPARES            2
#define MPTSAS_MAX_RAIDVOLS             2
#define MPTSAS_MAX_RAIDCONFIGS          5

/*
 * 64-bit SAS WWN is displayed as 16 characters as HEX characters,
 * plus two means the prefix 'w' and end of the string '\0'.
 */
#define MPTSAS_WWN_STRLEN       (16 + 2)
#define MPTSAS_MAX_GUID_LEN     64

/*
 * DMA routine flags
 */
#define MPTSAS_DMA_HANDLE_ALLOCD        0x2
#define MPTSAS_DMA_MEMORY_ALLOCD        0x4
#define MPTSAS_DMA_HANDLE_BOUND 0x8
```

```
 127 /*
 128  * If the HBA supports DMA or bus-mastering, you may have your own
 129  * scatter-gather list for physically non-contiguous memory in one
 130  * I/O operation; if so, there's probably a size for that list.
 131  * It must be placed in the ddi_dma_lim_t structure, so that the system
 132  * DMA-support routines can use it to break up the I/O request, so we
 133  * define it here.
 134  */
 135 #if defined(__sparc)
 136 #define MPTSAS_MAX_DMA_SEGS     1
 137 #define MPTSAS_MAX_CMD_SEGS     1
 138 #else
 139 #define MPTSAS_MAX_DMA_SEGS     256
 140 #define MPTSAS_MAX_CMD_SEGS     257
 141 #endif
 142 #define MPTSAS_MAX_FRAME_SGES(mpt) \
 143         (((mpt->m_req_frame_size - (sizeof (MPI2_SCSI_IO_REQUEST))) / 8) + 1)

 145 /*
 146  * Calculating how many 64-bit DMA simple elements can be stored in the first
 145  * Caculating how many 64-bit DMA simple elements can be stored in the first
 147  * frame. Note that msg_scsi_io_request contains 2 double-words (8 bytes) for
 148  * element storage.  And 64-bit dma element is 3 double-words (12 bytes) in
 149  * size. IEEE 64-bit dma element used for SAS3 controllers is 4 double-words
 150  * (16 bytes).
 148  * size.
 151  */
 152 #define MPTSAS_MAX_FRAME_SGES64(mpt) \
 153         ((mpt->m_req_frame_size - \
 154         sizeof (MPI2_SCSI_IO_REQUEST) + sizeof (MPI2_SGE_IO_UNION)) / \
 155         (mpt->m_MPI25 ? sizeof (MPI2_IEEE_SGE_SIMPLE64) : \
 156         sizeof (MPI2_SGE_SIMPLE64)))
 152         (sizeof (MPI2_SCSI_IO_REQUEST)) + sizeof (MPI2_SGE_IO_UNION)) / 12)

 158 /*
 159  * Scatter-gather list structure defined by HBA hardware
 160  */
 161 typedef struct NcrTableIndirect {       /* Table Indirect entries */
 162         uint32_t count;         /* 24 bit count */
 163         union {
 164                 uint32_t address32;     /* 32 bit address */
 165                 struct {
 166                         uint32_t Low;
 167                         uint32_t High;
 168                 } address64;            /* 64 bit address */
 169         } addr;
 170 } mptti_t;
_____unchanged_portion_omitted_

 208 TAILQ_HEAD(mptsas_active_cmdq, mptsas_cmd);
 209 typedef struct mptsas_active_cmdq mptsas_active_cmdq_t;

 211 typedef struct mptsas_target {
 212                 mptsas_target_addr_t    m_addr;
 213                 refhash_link_t          m_link;
 214                 uint8_t                 m_dr_flag;
 215                 uint16_t                m_devhdl;
 216                 uint32_t                m_deviceinfo;
 217                 uint8_t                 m_phynum;
 218                 uint32_t                m_dups;
 219                 mptsas_active_cmdq_t    m_active_cmdq;
 220                 int32_t                 m_t_throttle;
 221                 int32_t                 m_t_ncmds;
 222                 int32_t                 m_reset_delay;
 223                 int32_t                 m_t_nwait;
```



```
 225                 uint16_t                m_qfull_retry_interval;
 226                 uint8_t                 m_qfull_retries;
 227                 uint16_t                m_enclosure;
 228                 uint16_t                m_slot_num;
 229                 uint32_t                m_tgt_unconfigured;
 230                 uint8_t                 m_led_status;
 231                 uint8_t                 m_scsi_req_desc_type;

 233 } mptsas_target_t;
_____unchanged_portion_omitted_

 248 typedef struct mptsas_cache_frames {
 249         ddi_dma_handle_t m_dma_hdl;
 250         ddi_acc_handle_t m_acc_hdl;
 251         caddr_t m_frames_addr;
 252         uint64_t m_phys_addr;
 247         uint32_t m_phys_addr;
 253 } mptsas_cache_frames_t;
_____unchanged_portion_omitted_

 370 /*
 371  * passthrough request structure
 372  */
 373 typedef struct mptsas_pt_request {
 374         uint8_t *request;
 375         uint32_t request_size;
 376         uint32_t data_size;
 377         uint32_t dataout_size;
 378         uint8_t direction;
 379         uint8_t simple;
 380         uint16_t sgl_offset;
 373         uint32_t direction;
 381         ddi_dma_cookie_t data_cookie;
 382         ddi_dma_cookie_t dataout_cookie;
 383 } mptsas_pt_request_t;
_____unchanged_portion_omitted_

 679 typedef struct mptsas {
 680         int             m_instance;

 682         struct mptsas *m_next;

 684         scsi_hba_tran_t         *m_tran;
 685         smp_hba_tran_t          *m_smptran;
 686         kmutex_t                m_mutex;
 687         kmutex_t                m_passthru_mutex;
 688         kcondvar_t              m_cv;
 689         kcondvar_t              m_passthru_cv;
 690         kcondvar_t              m_fw_cv;
 691         kcondvar_t              m_config_cv;
 692         kcondvar_t              m_fw_diag_cv;
 693         dev_info_t              *m_dip;

 695         /*
 696          * soft state flags
 697          */
 698         uint_t          m_softstate;

 700         refhash_t       *m_targets;
 701         refhash_t       *m_smp_targets;

 703         m_raidconfig_t  m_raidconfig[MPTSAS_MAX_RAIDCONFIGS];
 704         uint8_t         m_num_raid_configs;

 706         struct mptsas_slots *m_active;  /* outstanding cmds */
```

```
708         mptsas_cmd_t    *m_waitq;       /* cmd queue for active request */
709         mptsas_cmd_t    **m_waitqtail;  /* wait queue tail ptr */

711         kmutex_t        m_tx_waitq_mutex;
712         mptsas_cmd_t    *m_tx_waitq;    /* TX cmd queue for active request */
713         mptsas_cmd_t    **m_tx_waitqtail;       /* tx_wait queue tail ptr */
714         int             m_tx_draining;  /* TX queue draining flag */

716         mptsas_cmd_t    *m_doneq;       /* queue of completed commands */
717         mptsas_cmd_t    **m_donetail;   /* queue tail ptr */

719         /*
720          * variables for helper threads (fan-out interrupts)
721          */
722         mptsas_doneq_thread_list_t      *m_doneq_thread_id;
723         uint32_t                m_doneq_thread_n;
724         uint32_t                m_doneq_thread_threshold;
725         uint32_t                m_doneq_length_threshold;
726         uint32_t                m_doneq_len;
727         kcondvar_t              m_doneq_thread_cv;
728         kmutex_t                m_doneq_mutex;

730         int             m_ncmds;        /* number of outstanding commands */
731         m_event_struct_t *m_ioc_event_cmdq;     /* cmd queue for ioc event */
732         m_event_struct_t **m_ioc_event_cmdtail; /* ioc cmd queue tail */

734         ddi_acc_handle_t m_datap;       /* operating regs data access handle */

736         struct _MPI2_SYSTEM_INTERFACE_REGS      *m_reg;

738         ushort_t        m_devid;        /* device id of chip. */
739         uchar_t         m_revid;        /* revision of chip. */
740         uint16_t        m_svid;         /* subsystem Vendor ID of chip */
741         uint16_t        m_ssid;         /* subsystem Device ID of chip */

743         uchar_t         m_sync_offset;  /* default offset for this chip. */

745         timeout_id_t    m_quiesce_timeid;

747         ddi_dma_handle_t m_dma_req_frame_hdl;
748         ddi_acc_handle_t m_acc_req_frame_hdl;
749         ddi_dma_handle_t m_dma_reply_frame_hdl;
750         ddi_acc_handle_t m_acc_reply_frame_hdl;
751         ddi_dma_handle_t m_dma_free_queue_hdl;
752         ddi_acc_handle_t m_acc_free_queue_hdl;
753         ddi_dma_handle_t m_dma_post_queue_hdl;
754         ddi_acc_handle_t m_acc_post_queue_hdl;

756         /*
757          * list of reset notification requests
758          */
759         struct scsi_reset_notify_entry  *m_reset_notify_listf;

761         /*
762          * qfull handling
763          */
764         timeout_id_t    m_restart_cmd_timeid;

766         /*
767          * scsi reset delay per bus
768          */
769         uint_t          m_scsi_reset_delay;

771         int             m_pm_idle_delay;

773         uchar_t         m_polled_intr;  /* intr was polled. */
```



```
774         uchar_t         m_suspended;    /* true if driver is suspended */

776         struct kmem_cache *m_kmem_cache;
777         struct kmem_cache *m_cache_frames;

779         /*
780          * hba options.
781          */
782         uint_t          m_options;

784         int             m_in_callback;

786         int             m_power_level;  /* current power level */

788         int             m_busy;         /* power management busy state */

790         off_t           m_pmcsr_offset; /* PMCSR offset */

792         ddi_acc_handle_t m_config_handle;

794         ddi_dma_attr_t          m_io_dma_attr;  /* Used for data I/O */
795         ddi_dma_attr_t          m_msg_dma_attr; /* Used for message frames */
796         ddi_device_acc_attr_t   m_dev_acc_attr;
797         ddi_device_acc_attr_t   m_reg_acc_attr;

799         /*
800          * request/reply variables
801          */
802         caddr_t         m_req_frame;
803         uint64_t        m_req_frame_dma_addr;
804         caddr_t         m_reply_frame;
805         uint64_t        m_reply_frame_dma_addr;
806         caddr_t         m_free_queue;
807         uint64_t        m_free_queue_dma_addr;
808         caddr_t         m_post_queue;
809         uint64_t        m_post_queue_dma_addr;

811         m_replyh_arg_t *m_replyh_args;

813         uint16_t        m_max_requests;
814         uint16_t        m_req_frame_size;

816         /*
817          * Max frames per request reprted in IOC Facts
818          */
819         uint8_t         m_max_chain_depth;
820         /*
821          * Max frames per request which is used in reality. It's adjusted
822          * according DMA SG length attribute, and shall not exceed the
823          * m_max_chain_depth.
824          */
825         uint8_t         m_max_request_frames;

827         uint16_t        m_free_queue_depth;
828         uint16_t        m_post_queue_depth;
829         uint16_t        m_max_replies;
830         uint32_t        m_free_index;
831         uint32_t        m_post_index;
832         uint8_t         m_reply_frame_size;
833         uint32_t        m_ioc_capabilities;

835         /*
836          * indicates if the firmware was upload by the driver
837          * at boot time
838          */
839         ushort_t        m_fwupload;
```

```
841         uint16_t        m_productid;

843         /*
844          * per instance data structures for dma memory resources for
845          * MPI handshake protocol. only one handshake cmd can run at a time.
846          */
847         ddi_dma_handle_t        m_hshk_dma_hdl;
848         ddi_acc_handle_t        m_hshk_acc_hdl;
849         caddr_t                 m_hshk_memp;
850         size_t                  m_hshk_dma_size;

852         /* Firmware version on the card at boot time */
853         uint32_t                m_fwversion;

855         /* MSI specific fields */
856         ddi_intr_handle_t       *m_htable;      /* For array of interrupts */
857         int                     m_intr_type;    /* What type of interrupt */
858         int                     m_intr_cnt;     /* # of intrs count returned */
859         size_t                  m_intr_size;    /* Size of intr array */
860         uint_t                  m_intr_pri;     /* Interrupt priority   */
861         int                     m_intr_cap;     /* Interrupt capabilities */
862         ddi_taskq_t             *m_event_taskq;

864         /* SAS specific information */

866         union {
867                 uint64_t        m_base_wwid;    /* Base WWID */
868                 struct {
869 #ifdef _BIG_ENDIAN
870                         uint32_t        m_base_wwid_hi;
871                         uint32_t        m_base_wwid_lo;
872 #else
873                         uint32_t        m_base_wwid_lo;
874                         uint32_t        m_base_wwid_hi;
875 #endif
876                 } sasaddr;
877         } un;

879         uint8_t                 m_num_phys;             /* # of PHYs */
880         mptsas_phy_info_t       m_phy_info[MPTSAS_MAX_PHYS];
881         uint8_t                 m_port_chng;    /* initiator port changes */
882         MPI2_CONFIG_PAGE_MAN_0   m_MANU_page0;   /* Manufactor page 0 info */
883         MPI2_CONFIG_PAGE_MAN_1   m_MANU_page1;   /* Manufactor page 1 info */

885         /* FMA Capabilities */
886         int                     m_fm_capabilities;
887         ddi_taskq_t             *m_dr_taskq;
888         int                     m_mpxio_enable;
889         uint8_t                 m_done_traverse_dev;
890         uint8_t                 m_done_traverse_smp;
891         int                     m_diag_action_in_progress;
892         uint16_t                m_dev_handle;
893         uint16_t                m_smp_devhdl;

895         /*
896          * Event recording
897          */
898         uint8_t                 m_event_index;
899         uint32_t                m_event_number;
900         uint32_t                m_event_mask[4];
901         mptsas_event_entry_t    m_events[MPTSAS_EVENT_QUEUE_SIZE];

903         /*
904          * FW diag Buffer List
905          */
```



```
906         mptsas_fw_diagnostic_buffer_t
907                 m_fw_diag_buffer_list[MPI2_DIAG_BUF_TYPE_COUNT];

909         /* GEN3 support */
910         uint8_t                 m_MPI25;

912         /*
913          * Event Replay flag (MUR support)
914          */
915         uint8_t                 m_event_replay;

917         /*
918          * IR Capable flag
919          */
920         uint8_t                 m_ir_capable;

922         /*
923          * Is HBA processing a diag reset?
924          */
925         uint8_t                 m_in_reset;

927         /*
928          * per instance cmd data structures for task management cmds
929          */
930         m_event_struct_t        m_event_task_mgmt;      /* must be last */
931                                                         /* ... scsi_pkt_size */
932 } mptsas_t;
```

*unchanged_portion_omitted_*

```
1049 /*
1050  * inq_dtype:
1051  * Bits 5 through 7 are the Peripheral Device Qualifier
1052  * 001b: device not connected to the LUN
1053  * Bits 0 through 4 are the Peripheral Device Type
1054  * 1fh: Unknown or no device type
1055  *
1056  * Although the inquiry may return success, the following value
1057  * means no valid LUN connected.
1058  */
1059 #define MPTSAS_VALID_LUN(sd_inq) \
1060         (((sd_inq->inq_dtype & 0xe0) != 0x20) && \
1061         ((sd_inq->inq_dtype & 0x1f) != 0x1f))

1063 /*
1064  * Default is to have 10 retries on receiving QFULL status and
1065  * each retry to be after 100 ms.
1066  */
1067 #define QFULL_RETRIES           10
1068 #define QFULL_RETRY_INTERVAL    100

1070 /*
1071  * Handy macros
1072  */
1073 #define Tgt(sp) ((sp)->cmd_pkt->pkt_address.a_target)
1074 #define Lun(sp) ((sp)->cmd_pkt->pkt_address.a_lun)

1076 #define IS_HEX_DIGIT(n) (((n) >= '0' && (n) <= '9') || \
1077         ((n) >= 'a' && (n) <= 'f') || ((n) >= 'A' && (n) <= 'F'))

1079 /*
1080  * poll time for mptsas_pollret() and mptsas_wait_intr()
1081  */
1082 #define MPTSAS_POLL_TIME        30000   /* 30 seconds */

1084 /*
1085  * default time for mptsas_do_passthru
```

```
 */
#define MPTSAS_PASS_THRU_TIME_DEFAULT   60      /* 60 seconds */

/*
 * macro to return the effective address of a given per-target field
 */
#define EFF_ADDR(start, offset)         ((start) + (offset))

#define SDEV2ADDR(devp)         (&((devp)->sd_address))
#define SDEV2TRAN(devp)         ((devp)->sd_address.a_hba_tran)
#define PKT2TRAN(pkt)           ((pkt)->pkt_address.a_hba_tran)
#define ADDR2TRAN(ap)           ((ap)->a_hba_tran)
#define DIP2TRAN(dip)           (ddi_get_driver_private(dip))


#define TRAN2MPT(hba)           ((mptsas_t *)(hba)->tran_hba_private)
#define DIP2MPT(dip)            (TRAN2MPT((scsi_hba_tran_t *)DIP2TRAN(dip)))
#define SDEV2MPT(sd)            (TRAN2MPT(SDEV2TRAN(sd)))
#define PKT2MPT(pkt)            (TRAN2MPT(PKT2TRAN(pkt)))

#define ADDR2MPT(ap)            (TRAN2MPT(ADDR2TRAN(ap)))

#define POLL_TIMEOUT            (2 * SCSI_POLL_TIMEOUT * 1000000)
#define SHORT_POLL_TIMEOUT      (1000000)       /* in usec, about 1 secs */
#define MPTSAS_QUIESCE_TIMEOUT  1               /* 1 sec */
#define MPTSAS_PM_IDLE_TIMEOUT  60              /* 60 seconds */

#define MPTSAS_GET_ISTAT(mpt)  (ddi_get32((mpt)->m_datap, \
                        &(mpt)->m_reg->HostInterruptStatus))

#define MPTSAS_SET_SIGP(P) \
                ClrSetBits(mpt->m_devaddr + NREG_ISTAT, 0, NB_ISTAT_SIGP)

#define MPTSAS_RESET_SIGP(P) (void) ddi_get8(mpt->m_datap, \
                        (uint8_t *)(mpt->m_devaddr + NREG_CTEST2))

#define MPTSAS_GET_INTCODE(P) (ddi_get32(mpt->m_datap, \
                        (uint32_t *)(mpt->m_devaddr + NREG_DSPS)))


#define MPTSAS_START_CMD(mpt, req_desc_lo, req_desc_hi) \
        ddi_put32(mpt->m_datap, &mpt->m_reg->RequestDescriptorPostLow,\
            req_desc_lo);\
        ddi_put32(mpt->m_datap, &mpt->m_reg->RequestDescriptorPostHigh,\
            req_desc_hi);

#define INTPENDING(mpt) \
        (MPTSAS_GET_ISTAT(mpt) & MPI2_HIS_REPLY_DESCRIPTOR_INTERRUPT)

/*
 * Mask all interrupts to disable
 */
#define MPTSAS_DISABLE_INTR(mpt)        \
        ddi_put32((mpt)->m_datap, &(mpt)->m_reg->HostInterruptMask, \
            (MPI2_HIM_RIM | MPI2_HIM_DIM | MPI2_HIM_RESET_IRQ_MASK))

/*
 * Mask Doorbell and Reset interrupts to enable reply desc int.
 */
#define MPTSAS_ENABLE_INTR(mpt) \
        ddi_put32(mpt->m_datap, &mpt->m_reg->HostInterruptMask, \
        (MPI2_HIM_DIM | MPI2_HIM_RESET_IRQ_MASK))

#define MPTSAS_GET_NEXT_REPLY(mpt, index)  \
        &((uint64_t *)(void *)mpt->m_post_queue)[index]
```



```
#define MPTSAS_GET_NEXT_FRAME(mpt, SMID) \
        (mpt->m_req_frame + (mpt->m_req_frame_size * SMID))

#define ClrSetBits32(hdl, reg, clr, set) \
        ddi_put32(hdl, (reg), \
            ((ddi_get32(mpt->m_datap, (reg)) & ~(clr)) | (set)))

#define ClrSetBits(reg, clr, set) \
        ddi_put8(mpt->m_datap, (uint8_t *)(reg), \
                ((ddi_get8(mpt->m_datap, (uint8_t *)(reg)) & ~(clr)) | (set)))

#define MPTSAS_WAITQ_RM(mpt, cmdp)      \
        if ((cmdp = mpt->m_waitq) != NULL) { \
                /* If the queue is now empty fix the tail pointer */    \
                if ((mpt->m_waitq = cmdp->cmd_linkp) == NULL) \
                        mpt->m_waitqtail = &mpt->m_waitq; \
                cmdp->cmd_linkp = NULL; \
                cmdp->cmd_queued = FALSE; \
        }

#define MPTSAS_TX_WAITQ_RM(mpt, cmdp)   \
        if ((cmdp = mpt->m_tx_waitq) != NULL) { \
                /* If the queue is now empty fix the tail pointer */    \
                if ((mpt->m_tx_waitq = cmdp->cmd_linkp) == NULL) \
                        mpt->m_tx_waitqtail = &mpt->m_tx_waitq; \
                cmdp->cmd_linkp = NULL; \
                cmdp->cmd_queued = FALSE; \
        }

/*
 * defaults for the global properties
 */
#define DEFAULT_SCSI_OPTIONS    SCSI_OPTIONS_DR
#define DEFAULT_TAG_AGE_LIMIT   2
#define DEFAULT_WD_TICK         1

/*
 * invalid hostid.
 */
#define MPTSAS_INVALID_HOSTID  -1

/*
 * Get/Set hostid from SCSI port configuration page
 */
#define MPTSAS_GET_HOST_ID(configuration) (configuration & 0xFF)
#define MPTSAS_SET_HOST_ID(hostid) (hostid | ((1 << hostid) << 16))

/*
 * Config space.
 */
#define MPTSAS_LATENCY_TIMER    0x40

/*
 * Offset to firmware version
 */
#define MPTSAS_FW_VERSION_OFFSET        9

/*
 * Offset and masks to get at the ProductId field
 */
#define MPTSAS_FW_PRODUCTID_OFFSET      8
#define MPTSAS_FW_PRODUCTID_MASK        0xFFFF0000
#define MPTSAS_FW_PRODUCTID_SHIFT       16

/*
 * Subsystem ID for HBAs.
```

```
 */
#define MPTSAS_HBA_SUBSYSTEM_ID    0x10C0
#define MPTSAS_RHEA_SUBSYSTEM_ID        0x10B0

/*
 * reset delay tick
 */
#define MPTSAS_WATCH_RESET_DELAY_TICK 50        /* specified in milli seconds */

/*
 * Ioc reset return values
 */
#define MPTSAS_RESET_FAIL       -1
#define MPTSAS_NO_RESET         0
#define MPTSAS_SUCCESS_HARDRESET        1
#define MPTSAS_SUCCESS_MUR      2

/*
 * throttle support.
 */
#define MAX_THROTTLE    32
#define HOLD_THROTTLE   0
#define DRAIN_THROTTLE  -1
#define QFULL_THROTTLE  -2

/*
 * Passthrough/config request flags
 */
#define MPTSAS_DATA_ALLOCATED           0x0001
#define MPTSAS_DATAOUT_ALLOCATED        0x0002
#define MPTSAS_REQUEST_POOL_CMD         0x0004
#define MPTSAS_ADDRESS_REPLY            0x0008
#define MPTSAS_CMD_TIMEOUT              0x0010

/*
 * response code tlr flag
 */
#define MPTSAS_SCSI_RESPONSE_CODE_TLR_OFF       0x02

/*
 * System Events
 */
#ifndef DDI_VENDOR_LSI
#define DDI_VENDOR_LSI  "LSI"
#endif  /* DDI_VENDOR_LSI */

/*
 * Shared functions
 */
int mptsas_save_cmd(struct mptsas *mpt, struct mptsas_cmd *cmd);
void mptsas_remove_cmd(mptsas_t *mpt, mptsas_cmd_t *cmd);
void mptsas_waitq_add(mptsas_t *mpt, mptsas_cmd_t *cmd);
void mptsas_log(struct mptsas *mpt, int level, char *fmt, ...);
int mptsas_poll(mptsas_t *mpt, mptsas_cmd_t *poll_cmd, int polltime);
int mptsas_do_dma(mptsas_t *mpt, uint32_t size, int var, int (*callback)());
int mptsas_send_config_request_msg(mptsas_t *mpt, uint8_t action,
        uint8_t pagetype, uint32_t pageaddress, uint8_t pagenumber,
        uint8_t pageversion, uint8_t pagelength, uint32_t
        SGEflagslength, uint32_t SGEaddress32);
int mptsas_send_extended_config_request_msg(mptsas_t *mpt, uint8_t action,
        uint8_t extpagetype, uint32_t pageaddress, uint8_t pagenumber,
        uint8_t pageversion, uint16_t extpagelength,
        uint32_t SGEflagslength, uint32_t SGEaddress32);
int mptsas_update_flash(mptsas_t *mpt, caddr_t ptrbuffer, uint32_t size,
        uint8_t type, int mode);
int mptsas_check_flash(mptsas_t *mpt, caddr_t origfile, uint32_t size,
```



```
        uint8_t type, int mode);
int mptsas_download_firmware();
int mptsas_can_download_firmware();
int mptsas_dma_alloc(mptsas_t *mpt, mptsas_dma_alloc_state_t *dma_statep);
void mptsas_dma_free(mptsas_dma_alloc_state_t *dma_statep);
mptsas_phymask_t mptsas_physport_to_phymask(mptsas_t *mpt, uint8_t physport);
void mptsas_fma_check(mptsas_t *mpt, mptsas_cmd_t *cmd);
int mptsas_check_acc_handle(ddi_acc_handle_t handle);
int mptsas_check_dma_handle(ddi_dma_handle_t handle);
void mptsas_fm_ereport(mptsas_t *mpt, char *detail);
int mptsas_dma_addr_create(mptsas_t *mpt, ddi_dma_attr_t dma_attr,
    ddi_dma_handle_t *dma_hdp, ddi_acc_handle_t *acc_hdp, caddr_t *dma_memp,
    uint32_t alloc_size, ddi_dma_cookie_t *cookiep);
void mptsas_dma_addr_destroy(ddi_dma_handle_t *, ddi_acc_handle_t *);

/*
 * impl functions
 */
int mptsas_ioc_wait_for_response(mptsas_t *mpt);
int mptsas_ioc_wait_for_doorbell(mptsas_t *mpt);
int mptsas_ioc_reset(mptsas_t *mpt, int);
int mptsas_send_handshake_msg(mptsas_t *mpt, caddr_t memp, int numbytes,
    ddi_acc_handle_t accessp);
int mptsas_get_handshake_msg(mptsas_t *mpt, caddr_t memp, int numbytes,
    ddi_acc_handle_t accessp);
int mptsas_send_config_request_msg(mptsas_t *mpt, uint8_t action,
    uint8_t pagetype, uint32_t pageaddress, uint8_t pagenumber,
    uint8_t pageversion, uint8_t pagelength, uint32_t SGEflagslength,
    uint32_t SGEaddress32);
int mptsas_send_extended_config_request_msg(mptsas_t *mpt, uint8_t action,
    uint8_t extpagetype, uint32_t pageaddress, uint8_t pagenumber,
    uint8_t pageversion, uint16_t extpagelength,
    uint32_t SGEflagslength, uint32_t SGEaddress32);

int mptsas_request_from_pool(mptsas_t *mpt, mptsas_cmd_t **cmd,
    struct scsi_pkt **pkt);
void mptsas_return_to_pool(mptsas_t *mpt, mptsas_cmd_t *cmd);
void mptsas_destroy_ioc_event_cmd(mptsas_t *mpt);
void mptsas_start_config_page_access(mptsas_t *mpt, mptsas_cmd_t *cmd);
int mptsas_access_config_page(mptsas_t *mpt, uint8_t action, uint8_t page_type,
    uint8_t page_number, uint32_t page_address, int (*callback) (mptsas_t *,
    caddr_t, ddi_acc_handle_t, uint16_t, uint32_t, va_list), ...);

int mptsas_ioc_task_management(mptsas_t *mpt, int task_type,
    uint16_t dev_handle, int lun, uint8_t *reply, uint32_t reply_size,
    int mode);
int mptsas_send_event_ack(mptsas_t *mpt, uint32_t event, uint32_t eventcntx);
void mptsas_send_pending_event_ack(mptsas_t *mpt);
void mptsas_set_throttle(struct mptsas *mpt, mptsas_target_t *ptgt, int what);
int mptsas_restart_ioc(mptsas_t *mpt);
void mptsas_update_driver_data(struct mptsas *mpt);
uint64_t mptsas_get_sata_guid(mptsas_t *mpt, mptsas_target_t *ptgt, int lun);

/*
 * init functions
 */
int mptsas_ioc_get_facts(mptsas_t *mpt);
int mptsas_ioc_get_port_facts(mptsas_t *mpt, int port);
int mptsas_ioc_enable_port(mptsas_t *mpt);
int mptsas_ioc_enable_event_notification(mptsas_t *mpt);
int mptsas_ioc_init(mptsas_t *mpt);

/*
 * configuration pages operation
 */
int mptsas_get_sas_device_page0(mptsas_t *mpt, uint32_t page_address,
```

```
1350     uint16_t *dev_handle, uint64_t *sas_wwn, uint32_t *dev_info,
1351     uint8_t *physport, uint8_t *phynum, uint16_t *pdevhandle,
1352     uint16_t *slot_num, uint16_t *enclosure);
1353 int mptsas_get_sas_io_unit_page(mptsas_t *mpt);
1354 int mptsas_get_sas_io_unit_page_hndshk(mptsas_t *mpt);
1355 int mptsas_get_sas_expander_page0(mptsas_t *mpt, uint32_t page_address,
1356     mptsas_smp_t *info);
1357 int mptsas_set_ioc_params(mptsas_t *mpt);
1358 int mptsas_get_manufacture_page5(mptsas_t *mpt);
1359 int mptsas_get_sas_port_page0(mptsas_t *mpt, uint32_t page_address,
1360     uint64_t *sas_wwn, uint8_t *portwidth);
1361 int mptsas_get_bios_page3(mptsas_t *mpt,  uint32_t *bios_version);
1362 int
1363 mptsas_get_sas_phy_page0(mptsas_t *mpt, uint32_t page_address,
1364     smhba_info_t *info);
1365 int
1366 mptsas_get_sas_phy_page1(mptsas_t *mpt, uint32_t page_address,
1367     smhba_info_t *info);
1368 int
1369 mptsas_get_manufacture_page0(mptsas_t *mpt);
1370 void
1371 mptsas_create_phy_stats(mptsas_t *mpt, char *iport, dev_info_t *dip);
1372 void mptsas_destroy_phy_stats(mptsas_t *mpt);
1373 int mptsas_smhba_phy_init(mptsas_t *mpt);
1374 /*
1375  * RAID functions
1376  */
1377 int mptsas_get_raid_settings(mptsas_t *mpt, mptsas_raidvol_t *raidvol);
1378 int mptsas_get_raid_info(mptsas_t *mpt);
1379 int mptsas_get_physdisk_settings(mptsas_t *mpt, mptsas_raidvol_t *raidvol,
1380     uint8_t physdisknum);
1381 int mptsas_delete_volume(mptsas_t *mpt, uint16_t volid);
1382 void mptsas_raid_action_system_shutdown(mptsas_t *mpt);

1384 #define MPTSAS_IOCSTATUS(status) (status & MPI2_IOCSTATUS_MASK)
1385 /*
1386  * debugging.
1387  */
1388 #if defined(MPTSAS_DEBUG)

1390 void mptsas_printf(char *fmt, ...);

1392 #define MPTSAS_DBGPR(m, args)   \
1393         if (mptsas_debug_flags & (m)) \
1394                 mptsas_printf args
1395 #else   /* ! defined(MPTSAS_DEBUG) */
1396 #define MPTSAS_DBGPR(m, args)
1397 #endif  /* defined(MPTSAS_DEBUG) */

1399 #define NDBG0(args)     MPTSAS_DBGPR(0x01, args)        /* init */
1400 #define NDBG1(args)     MPTSAS_DBGPR(0x02, args)        /* normal running */
1401 #define NDBG2(args)     MPTSAS_DBGPR(0x04, args)        /* property handling */
1402 #define NDBG3(args)     MPTSAS_DBGPR(0x08, args)        /* pkt handling */

1404 #define NDBG4(args)     MPTSAS_DBGPR(0x10, args)        /* kmem alloc/free */
1405 #define NDBG5(args)     MPTSAS_DBGPR(0x20, args)        /* polled cmds */
1406 #define NDBG6(args)     MPTSAS_DBGPR(0x40, args)        /* interrupts */
1407 #define NDBG7(args)     MPTSAS_DBGPR(0x80, args)        /* queue handling */

1409 #define NDBG8(args)     MPTSAS_DBGPR(0x0100, args)      /* arq */
1410 #define NDBG9(args)     MPTSAS_DBGPR(0x0200, args)      /* Tagged Q'ing */
1411 #define NDBG10(args)    MPTSAS_DBGPR(0x0400, args)      /* halting chip */
1412 #define NDBG11(args)    MPTSAS_DBGPR(0x0800, args)      /* power management */

1414 #define NDBG12(args)    MPTSAS_DBGPR(0x1000, args)      /* enumeration */
1415 #define NDBG13(args)    MPTSAS_DBGPR(0x2000, args)      /* configuration page */
```



```
1416 #define NDBG14(args)    MPTSAS_DBGPR(0x4000, args)      /* LED control */
1417 #define NDBG15(args)    MPTSAS_DBGPR(0x8000, args)      /* Passthrough */
1407 #define NDBG15(args)    MPTSAS_DBGPR(0x8000, args)

1419 #define NDBG16(args)    MPTSAS_DBGPR(0x010000, args)
1420 #define NDBG17(args)    MPTSAS_DBGPR(0x020000, args)    /* scatter/gather */
1421 #define NDBG18(args)    MPTSAS_DBGPR(0x040000, args)
1422 #define NDBG19(args)    MPTSAS_DBGPR(0x080000, args)    /* handshaking */

1424 #define NDBG20(args)    MPTSAS_DBGPR(0x100000, args)    /* events */
1425 #define NDBG21(args)    MPTSAS_DBGPR(0x200000, args)    /* dma */
1426 #define NDBG22(args)    MPTSAS_DBGPR(0x400000, args)    /* reset */
1427 #define NDBG23(args)    MPTSAS_DBGPR(0x800000, args)    /* abort */

1429 #define NDBG24(args)    MPTSAS_DBGPR(0x1000000, args)   /* capabilities */
1430 #define NDBG25(args)    MPTSAS_DBGPR(0x2000000, args)   /* flushing */
1431 #define NDBG26(args)    MPTSAS_DBGPR(0x4000000, args)
1432 #define NDBG27(args)    MPTSAS_DBGPR(0x8000000, args)

1434 #define NDBG28(args)    MPTSAS_DBGPR(0x10000000, args)  /* hotplug */
1435 #define NDBG29(args)    MPTSAS_DBGPR(0x20000000, args)  /* timeouts */
1436 #define NDBG30(args)    MPTSAS_DBGPR(0x40000000, args)  /* mptsas_watch */
1437 #define NDBG31(args)    MPTSAS_DBGPR(0x80000000, args)  /* negotations */

1439 /*
1440  * auto request sense
1441  */
1442 #define RQ_MAKECOM_COMMON(pkt, flag, cmd) \
1443         (pkt)->pkt_flags = (flag), \
1444         ((union scsi_cdb *)(pkt)->pkt_cdbp)->scc_cmd = (cmd), \
1445         ((union scsi_cdb *)(pkt)->pkt_cdbp)->scc_lun = \
1446             (pkt)->pkt_address.a_lun

1448 #define RQ_MAKECOM_G0(pkt, flag, cmd, addr, cnt) \
1449         RQ_MAKECOM_COMMON((pkt), (flag), (cmd)), \
1450         FORMG0ADDR(((union scsi_cdb *)(pkt)->pkt_cdbp), (addr)), \
1451         FORMG0COUNT(((union scsi_cdb *)(pkt)->pkt_cdbp), (cnt))


1454 #ifdef  __cplusplus
1455 }
```

_____***unchanged_portion_omitted_***