```
**********************************************************
    9839 Fri Apr 25 16:07:57 2014
new/usr/src/uts/i86pc/io/pcplusmp/apic_regops.c
4805 apic_mode global should be an enum
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL 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.
 */

#include <sys/cpuvar.h>
#include <sys/psm.h>
#include <sys/archsystm.h>
#include <sys/apic.h>
#include <sys/sunddi.h>
#include <sys/ddi_impldefs.h>
#include <sys/mach_intr.h>
#include <sys/sysmacros.h>
#include <sys/trap.h>
#include <sys/x86_archext.h>
#include <sys/privregs.h>
#include <sys/psm_common.h>

/* Function prototypes of local apic and X2APIC */
static uint64_t local_apic_read(uint32_t reg);
static void local_apic_write(uint32_t reg, uint64_t value);
static int get_local_apic_pri(void);
static void local_apic_write_task_reg(uint64_t value);
static void local_apic_write_int_cmd(uint32_t cpu_id, uint32_t cmd1);
static uint64_t local_x2apic_read(uint32_t msr);
static void local_x2apic_write(uint32_t msr, uint64_t value);
static int get_local_x2apic_pri(void);
static void local_x2apic_write_task_reg(uint64_t value);
static void local_x2apic_write_int_cmd(uint32_t cpu_id, uint32_t cmd1);

/*
 * According to the X2APIC specification:
 *
 *   xAPIC global enable    X2APIC enable         Description
 *   (IA32_APIC_BASE[11])   (IA32_APIC_BASE[10])
 * -----------------------------------------------------------
 *      0                       0       APIC is disabled
 *      0                       1       Invalid
 *      1                       0       APIC is enabled in xAPIC mode
 *      1                       1       APIC is enabled in X2APIC mode
 * -----------------------------------------------------------
```



```
 */
int     x2apic_enable = 1;
apic_mode_t apic_mode = LOCAL_APIC;     /* Default mode is Local APIC */
int     apic_mode = LOCAL_APIC;         /* Default mode is Local APIC */

/* Uses MMIO (Memory Mapped IO) */
static apic_reg_ops_t local_apic_regs_ops = {
        local_apic_read,
        local_apic_write,
        get_local_apic_pri,
        local_apic_write_task_reg,
        local_apic_write_int_cmd,
        apic_send_EOI,
};
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   27561 Fri Apr 25 16:07:57 2014
new/usr/src/uts/i86pc/sys/apic.h
4805 apic_mode global should be an enum
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL 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) 1993, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*
 * Copyright (c) 2010, Intel Corporation.
 * All rights reserved.
 */

#ifndef _SYS_APIC_APIC_H
#define _SYS_APIC_APIC_H

#include <sys/psm_types.h>
#include <sys/avintr.h>
#include <sys/pci.h>

#ifdef __cplusplus
extern "C" {
#endif

#include <sys/psm_common.h>

#define APIC_PCPLUSMP_NAME      "pcplusmp"
#define APIC_APIX_NAME          "apix"

#define APIC_IO_ADDR    0xfec00000
#define APIC_LOCAL_ADDR 0xfee00000
#define APIC_IO_MEMLEN  0xf
#define APIC_LOCAL_MEMLEN       0xfffff

/* Local Unit ID register */
#define APIC_LID_REG            0x8

/* I/o Unit Version Register */
#define APIC_VERS_REG           0xc

/* Task Priority register */
#define APIC_TASK_REG           0x20

/* EOI register */
#define APIC_EOI_REG            0x2c
```



```
/* Remote Read register         */
#define APIC_REMOTE_READ        0x30

/* Logical Destination register */
#define APIC_DEST_REG           0x34

/* Destination Format register */
#define APIC_FORMAT_REG         0x38

/* Spurious Interrupt Vector register */
#define APIC_SPUR_INT_REG       0x3c

/* Error Status Register */
#define APIC_ERROR_STATUS       0xa0

/* Interrupt Command registers */
#define APIC_INT_CMD1           0xc0
#define APIC_INT_CMD2           0xc4

/* Local Interrupt Vector registers */
#define APIC_CMCI_VECT          0xbc
#define APIC_THERM_VECT         0xcc
#define APIC_PCINT_VECT         0xd0
#define APIC_INT_VECT0          0xd4
#define APIC_INT_VECT1          0xd8
#define APIC_ERR_VECT           0xdc

/* IPL for performance counter interrupts */
#define APIC_PCINT_IPL          0xe
#define APIC_LVT_MASK           0x10000         /* Mask bit (16) in LVT */

/* Initial Count register */
#define APIC_INIT_COUNT         0xe0

/* Current Count Register */
#define APIC_CURR_COUNT         0xe4
#define APIC_CURR_ADD           0x39    /* used for remote read command */
#define CURR_COUNT_OFFSET       (sizeof (int32_t) * APIC_CURR_COUNT)

/* Divider Configuration Register */
#define APIC_DIVIDE_REG         0xf8

/* Various mode for local APIC. Modes are mutually exclusive  */
typedef enum apic_mode {
        APIC_IS_DISABLED = 0,
        APIC_MODE_NOTSET,
        LOCAL_APIC,
        LOCAL_X2APIC
} apic_mode_t;
#define APIC_IS_DISABLED        0x0
#define APIC_MODE_NOTSET        0x1
#define LOCAL_APIC              0x2
#define LOCAL_X2APIC            0x3

/* x2APIC SELF IPI Register */
#define X2APIC_SELF_IPI         0xFC

/* General x2APIC constants used at various places */
#define APIC_SVR_SUPPRESS_BROADCAST_EOI         0x1000
#define APIC_DIRECTED_EOI_BIT                   0x1000000

/* IRR register */
#define APIC_IRR_REG            0x80

/* ISR register */
```

```
124 #define APIC_ISR_REG            0x40

126 #define APIC_IO_REG             0x0
127 #define APIC_IO_DATA            0x4
128 #define APIC_IO_EOI             0x10

130 /* Bit offset of APIC ID in LID_REG, INT_CMD and in DEST_REG */
131 #define APIC_ID_BIT_OFFSET      24
132 #define APIC_ICR_ID_BIT_OFFSET  24
133 #define APIC_LDR_ID_BIT_OFFSET  24

135 /*
136  * Choose between flat and clustered models by writing the following to the
137  * FORMAT_REG. 82489 DX documentation seemed to suggest that writing 0 will
138  * disable logical destination mode.
139  * Does not seem to be in the docs for local APICs on the processors.
140  */
141 #define APIC_FLAT_MODEL         0xFFFFFFFFUL
142 #define APIC_CLUSTER_MODEL      0x0FFFFFFF

144 /*
145  * The commands which follow are window selectors written to APIC_IO_REG
146  * before data can be read/written from/to APIC_IO_DATA
147  */

149 #define APIC_ID_CMD             0x0
150 #define APIC_VERS_CMD           0x1
151 #define APIC_RDT_CMD            0x10
152 #define APIC_RDT_CMD2           0x11

154 #define APIC_INTEGRATED_VERS    0x10    /* 0x10 & above indicates integrated */
155 #define IOAPIC_VER_82489DX      0x01    /* Version ID: 82489DX External APIC */

157 #define APIC_INT_SPURIOUS       -1

159 #define APIC_IMCR_P1    0x22            /* int mode conf register port 1 */
160 #define APIC_IMCR_P2    0x23            /* int mode conf register port 2 */
161 #define APIC_IMCR_SELECT 0x70           /* select imcr by writing into P1 */
162 #define APIC_IMCR_PIC   0x0             /* selects PIC mode (8259-> BSP) */
163 #define APIC_IMCR_APIC  0x1             /* selects APIC mode (8259->APIC) */

165 #define APIC_CT_VECT    0x4ac           /* conf table vector            */
166 #define APIC_CT_SIZE    1024            /* conf table size              */

168 #define APIC_ID         'MPAT'          /* conf table signature         */

170 #define VENID_AMD               0x1022
171 #define DEVID_8131_IOAPIC       0x7451
172 #define DEVID_8132_IOAPIC       0x7459

174 #define IOAPICS_NODE_NAME       "ioapics"
175 #define IOAPICS_CHILD_NAME      "ioapic"
176 #define IOAPICS_DEV_TYPE        "ioapic"
177 #define IOAPICS_PROP_VENID      "vendor-id"
178 #define IOAPICS_PROP_DEVID      "device-id"

180 #define IS_CLASS_IOAPIC(b, s, p) \
181         ((b) == PCI_CLASS_PERIPH && (s) == PCI_PERIPH_PIC &&    \
182         ((p) == PCI_PERIPH_PIC_IF_IO_APIC ||                    \
183         (p) == PCI_PERIPH_PIC_IF_IOX_APIC))

185 /*
186  * These macros are used in frequently called routines like
187  * apic_intr_enter().
188  */
189 #define X2APIC_WRITE(reg, v) \
```



```
190         wrmsr((REG_X2APIC_BASE_MSR + (reg >> 2)), v)

192 #define LOCAL_APIC_WRITE_REG(reg, v) \
193         apicadr[reg] = v

195 /*
196  * MP floating pointer structure defined in Intel MP Spec 1.1
197  */
198 struct apic_mpfps_hdr {
199         uint32_t        mpfps_sig;      /* _MP_ (0x5F4D505F)            */
200         uint32_t        mpfps_mpct_paddr; /* paddr of MP configuration tbl */
201         uchar_t mpfps_length;           /* in paragraph (16-bytes units) */
202         uchar_t mpfps_spec_rev;         /* version number of MP spec     */
203         uchar_t mpfps_checksum;         /* checksum of complete structure */
204         uchar_t mpfps_featinfo1;        /* mp feature info bytes 1       */
205         uchar_t mpfps_featinfo2;        /* mp feature info bytes 2       */
206         uchar_t mpfps_featinfo3;        /* mp feature info bytes 3       */
207         uchar_t mpfps_featinfo4;        /* mp feature info bytes 4       */
208         uchar_t mpfps_featinfo5;        /* mp feature info bytes 5       */
209 };
______unchanged_portion_omitted_

769 /* The irq # is implicit in the array index: */
770 extern struct ioapic_reprogram_data apic_reprogram_info[];

772 extern void apic_intr_exit(int ipl, int irq);
773 extern void x2apic_intr_exit(int ipl, int irq);
774 extern int apic_probe_common();
775 extern void apic_init_common();
776 extern void ioapic_init_intr();
777 extern void ioapic_disable_redirection();
778 extern int apic_addspl_common(int irqno, int ipl, int min_ipl, int max_ipl);
779 extern int apic_delspl_common(int irqno, int ipl, int min_ipl, int max_ipl);
780 extern void apic_cleanup_busy();
781 extern void apic_intr_redistribute();
782 extern uchar_t apic_xlate_vector(uchar_t vector);
783 extern uchar_t apic_allocate_vector(int ipl, int irq, int pri);
784 extern void apic_free_vector(uchar_t vector);
785 extern int apic_allocate_irq(int irq);
786 extern uint32_t apic_bind_intr(dev_info_t *dip, int irq, uchar_t ioapicid,
787     uchar_t intin);
788 extern int apic_rebind(apic_irq_t *irq_ptr, int bind_cpu,
789     struct ioapic_reprogram_data *drep);
790 extern int apic_rebind_all(apic_irq_t *irq_ptr, int bind_cpu);
791 extern int apic_introp_xlate(dev_info_t *dip, struct intrspec *ispec, int type);
792 extern int apic_intr_ops(dev_info_t *dip, ddi_intr_handle_impl_t *hdlp,
793     psm_intr_op_t intr_op, int *result);
794 extern int apic_state(psm_state_request_t *);
795 extern boolean_t apic_cpu_in_range(int cpu);
796 extern int apic_check_msi_support();
797 extern apic_irq_t *apic_find_irq(dev_info_t *dip, struct intrspec *ispec,
798     int type);
799 extern int apic_navail_vector(dev_info_t *dip, int pri);
800 extern int apic_alloc_msi_vectors(dev_info_t *dip, int inum, int count,
801     int pri, int behavior);
802 extern int apic_alloc_msix_vectors(dev_info_t *dip, int inum, int count,
803     int pri, int behavior);
804 extern void  apic_free_vectors(dev_info_t *dip, int inum, int count, int pri,
805     int type);
806 extern int apic_get_vector_intr_info(int vecirq,
807     apic_get_intr_t *intr_params_p);
808 extern uchar_t apic_find_multi_vectors(int pri, int count);
809 extern int apic_setup_io_intr(void *p, int irq, boolean_t deferred);
810 extern uint32_t *mapin_apic(uint32_t addr, size_t len, int flags);
811 extern uint32_t *mapin_ioapic(uint32_t addr, size_t len, int flags);
812 extern void mapout_apic(caddr_t addr, size_t len);
```

```
 813 extern void mapout_ioapic(caddr_t addr, size_t len);
 814 extern uchar_t apic_modify_vector(uchar_t vector, int irq);
 815 extern void apic_pci_msi_unconfigure(dev_info_t *rdip, int type, int inum);
 816 extern void apic_pci_msi_disable_mode(dev_info_t *rdip, int type);
 817 extern void apic_pci_msi_enable_mode(dev_info_t *rdip, int type, int inum);
 818 extern void apic_pci_msi_enable_vector(apic_irq_t *, int type, int inum,
 819     int vector, int count, int target_apic_id);
 820 extern char *apic_get_apic_type();
 821 extern uint16_t apic_get_apic_version();
 822 extern void x2apic_send_ipi();
 823 extern void apic_ret();
 824 extern int apic_detect_x2apic();
 825 extern void apic_enable_x2apic();
 826 extern int apic_local_mode();
 827 extern void apic_change_eoi();
 828 extern void apic_send_EOI(uint32_t);
 829 extern void apic_send_directed_EOI(uint32_t);
 830 extern uint_t apic_calibrate(volatile uint32_t *, uint16_t *);

 832 extern volatile uint32_t *apicadr;      /* virtual addr of local APIC   */
 833 extern int apic_forceload;
 834 extern apic_cpus_info_t *apic_cpus;
 835 #ifdef _MACHDEP
 836 extern cpuset_t apic_cpumask;
 837 #endif
 838 extern uint_t apic_picinit_called;
 839 extern uchar_t apic_ipltopri[MAXIPL+1];
 840 extern uchar_t apic_vector_to_irq[APIC_MAX_VECTOR+1];
 841 extern int apic_max_device_irq;
 842 extern int apic_min_device_irq;
 843 extern apic_irq_t *apic_irq_table[APIC_MAX_VECTOR+1];
 844 extern volatile uint32_t *apicioadr[MAX_IO_APIC];
 845 extern uchar_t apic_io_id[MAX_IO_APIC];
 846 extern lock_t apic_ioapic_lock;
 847 extern uint32_t apic_physaddr[MAX_IO_APIC];
 848 extern kmutex_t airq_mutex;
 849 extern int apic_first_avail_irq;
 850 extern uchar_t apic_vectortoipl[APIC_AVAIL_VECTOR / APIC_VECTOR_PER_IPL];
 851 extern int apic_imcrp;
 852 extern int apic_revector_pending;
 853 extern char apic_level_intr[APIC_MAX_VECTOR+1];
 854 extern uchar_t apic_resv_vector[MAXIPL+1];
 855 extern int apic_sample_factor_redistribution;
 856 extern int apic_int_busy_mark;
 857 extern int apic_int_free_mark;
 858 extern int apic_diff_for_redistribution;
 859 extern int apic_poweroff_method;
 860 extern int apic_enable_acpi;
 861 extern int apic_nproc;
 862 extern int apic_max_nproc;
 863 extern int apic_next_bind_cpu;
 864 extern int apic_redistribute_sample_interval;
 865 extern int apic_multi_msi_enable;
 866 extern int apic_sci_vect;
 867 extern int apic_hpet_vect;
 868 extern uchar_t apic_ipls[];
 869 extern apic_reg_ops_t *apic_reg_ops;
 870 extern apic_mode_t apic_mode;
 868 extern int apic_mode;
 871 extern void x2apic_update_psm();
 872 extern void apic_change_ops();
 873 extern void apic_common_send_ipi(int, int);
 874 extern void apic_set_directed_EOI_handler();
 875 extern int apic_directed_EOI_supported();

 877 extern apic_intrmap_ops_t *apic_vt_ops;
```



```
 879 #ifdef  __cplusplus
 880 }
_____unchanged_portion_omitted_
```