

## [PATCH] ATP867X PCI IDE driver

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-04/3039.html>

---

**From:** Eric Uhrhane ([ericu\\_at\\_google.com](mailto:ericu_at_google.com))

**Date:** 04/15/04

To: [marcelo@hera.kernel.org](mailto:marcelo@hera.kernel.org)

Date: Wed, 14 Apr 2004 15:46:06 -0700 (PDT)

Marcelo et al:

This is a new driver for the Acard/Artop PCI ATA/SATA cards (6885[LP]/6896[S]) based on the ATP867{A,B} chips. It supports connectivity only, no fancy ATP867B RAID stuff, for up to 8 drives. The first version of this patch [written for 2.4.18] has been tested on about a dozen x86 SMP machines with several kinds of hard drives, in PIO4 and UDMA5, as well as in a few other configurations. This 2.4.26 port has had minimal testing, but has also had minimal changes from the original.

Acard tell me that they may have seen issues running ATAPI devices using this driver, but that they can't be sure that they weren't seeing problems with higher-level software. I've given it a quick test with my DVD drive, and didn't see anything obvious. YMMV.

Feel free to send me comments directly, but I'll also try to check on the list via archives. Stay tuned for the 2.6.5 port.

Eric Uhrhane  
[ericu@google.com](mailto:ericu@google.com)

```
diff -rNBdU 3 linux-2.4.26.orig/Documentation/Configure.help linux-2.4.26/Documentation/Configure.help
--- linux-2.4.26.orig/Documentation/Configure.help 2004-04-14 06:05:24.000000000 -0700
+++ linux-2.4.26/Documentation/Configure.help 2004-04-14 14:17:34.000000000 -0700
@@ -1167,6 +1167,14 @@
```

Say Y here if you have an ATI IXP chipset IDE controller.

```
+Acard/Artop 867X (867A) Support
+CONFIG_BLK_DEV_ATP867X
+ This driver supports the Acard 6885[LP]/6896[S] PCI IDE RAID cards, based
+ on the Artop 867X chip. It supplies connectivity only, for up to 8 drives--
+ no fancy RAID features.
+
+ If unsure, say N.
+
+CMD64X/CMD680 chipset support
+CONFIG_BLK_DEV_CMD64X
```

Say Y here if you have an IDE controller which uses any of these

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
diff -rNBdU 3 linux-2.4.26.orig/drivers/ide/Config.in linux-2.4.26/drivers/ide/Config.in
--- linux-2.4.26.orig/drivers/ide/Config.in 2004-04-14 06:05:29.000000000 -0700
+++ linux-2.4.26/drivers/ide/Config.in 2004-04-14 14:21:52.000000000 -0700
@@ -50,6 +50,7 @@
     dep_tristate ' AMD and nVidia IDE support' CONFIG_BLK_DEV_AMD74XX
$CONFIG_BLK_DEV_IDEDMA_PCI
     dep_mbool ' AMD Viper ATA-66 Override' CONFIG_AMD74XX_OVERRIDE
$CONFIG_BLK_DEV_AMD74XX
     dep_tristate ' ATI IXP chipset IDE support' CONFIG_BLK_DEV_ATIIXP
$CONFIG_BLK_DEV_IDEDMA_PCI $CONFIG_X86
+ dep_bool ' ACARD 867X support' CONFIG_BLK_DEV_ATP867X
$CONFIG_BLK_DEV_IDEDMA_PCI
     dep_tristate ' CMD64{3|6|8|9} chipset support' CONFIG_BLK_DEV_CMD64X
$CONFIG_BLK_DEV_IDEDMA_PCI
     dep_tristate ' Compaq Triflex IDE support' CONFIG_BLK_DEV_TRIFLEX
$CONFIG_BLK_DEV_IDEDMA_PCI
     dep_tristate ' CY82C693 chipset support' CONFIG_BLK_DEV_CY82C693
$CONFIG_BLK_DEV_IDEDMA_PCI
diff -rNBdU 3 linux-2.4.26.orig/drivers/ide/pci/atp867.c linux-2.4.26/drivers/ide/pci/atp867.c
--- linux-2.4.26.orig/drivers/ide/pci/atp867.c 1969-12-31 16:00:00.000000000 -0800
+++ linux-2.4.26/drivers/ide/pci/atp867.c 2004-04-14 14:17:34.000000000 -0700
@@ -0,0 +1,737 @@
+/*
+ * Copyright 2003-4 by Eric Uhrhane and Google, Inc., with code adapted from
+ * via82cxxx.c, which was
+
+ * Copyright (c) 2000-2001 Vojtech Pavlik
+
+ * Based on the work of:
+ * Michel Aubry
+ * Jeff Garzik
+ * Andre Hedrick
+
+ * Sponsored by SuSE
+
+ * ATP867X IDE driver for Linux: Supports ATP867A chips in PIO 0-4, and UDMA
+ * 0-6. The chips theoretically support MWDMA 0-2 as well. This driver
+ * supplies only connectivity; if you want to hook up a lot of drives, here you
+ * go. I haven't implemented any of the ATP867B RAID stuff [comparison, XOR,
+ * striping, mirroring, etc.], and haven't included the pci id for the ATP867B,
+ * so that this driver won't be activated when you really want Acard's SCSI-IDE
+ * RAID driver. However, if you've got a B, and only need an A, you can add the
+ * id to use this driver if you so choose.
+ */
+
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
+ * This program is distributed in the hope that it will be useful,  
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of  
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
+ * GNU General Public License for more details.  
+ *  
+ * You should have received a copy of the GNU General Public License  
+ * along with this program; if not, write to the Free Software  
+ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
+ *  
+ */  
+  
+#include <linux/config.h>  
+#include <linux/module.h>  
+#include <linux/kernel.h>  
+#include <linux/ioport.h>  
+#include <linux/blkdev.h>  
+#include <linux/pci.h>  
+#include <linux/init.h>  
+#include <linux/ide.h>  
+#include <asm/io.h>  
+  
+#include "ide-timing.h"  
+#include "ide_modes.h"  
+#include "atp867.h"  
+  
+#undef NO_DMA  
+  
+#define ATP867X_NUM_CHANNELS 4  
+  
+/* IO Registers */  
+#define ATP867X_IO_CHANNEL_OFFSET 0x10  
+#define ATP867X_IOBASE(dev,port) (pci_resource_start((dev), 0) + \  
+ (port)*ATP867X_IO_CHANNEL_OFFSET)  
+  
+#define ATP867X_IO_PORTBASE(dev,port) (0x00+ATP867X_IOBASE((dev),(port)))  
+#define ATP867X_IO_STATUS(dev,port) (0x07+ATP867X_IO_PORTBASE((dev),(port)))  
+#define ATP867X_IO_ALTSTATUS(dev,port) (0x0E +ATP867X_IOBASE((dev),(port)))  
+  
+#define ATP867X_IO_DMABASE(dev,port) (0x40+ATP867X_IOBASE((dev),(port)))  
+#define ATP867X_IO_MSTRPIOSPD(dev,port) (0x08+ATP867X_IO_DMABASE((dev),(port)))  
+#define ATP867X_IO_SLAVEPIOSPD(dev,port)(0x09+ATP867X_IO_DMABASE((dev),(port)))  
+#define ATP867X_IO_8BPIOSPD(dev,port) (0x0A+ATP867X_IO_DMABASE((dev),(port)))  
+#define ATP867X_IO_DMAMODE(dev,port) (0x0B+ATP867X_IO_DMABASE((dev),(port)))  
+  
+/* IO Register Bitfields */  
+#define ATP867X_IO_PIOSPD_ACTIVE_SHIFT 4  
+#define ATP867X_IO_PIOSPD_RECOVER_SHIFT 0  
+  
+#define ATP867X_IO_DMAMODE_MSTR_SHIFT 4  
+#define ATP867X_IO_DMAMODE_MSTR_MASK 0x70  
+#define ATP867X_IO_DMAMODE_SLAVE_SHIFT 0
```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
+ #define ATP867X_IO_DMAMODE_SLAVE_MASK 0x07
+
+ #define ATP867X_IO_DMAMODE_UDMA_6 0x07
+ #define ATP867X_IO_DMAMODE_UDMA_5 0x06
+ #define ATP867X_IO_DMAMODE_UDMA_4 0x05
+ #define ATP867X_IO_DMAMODE_UDMA_3 0x04
+ #define ATP867X_IO_DMAMODE_UDMA_2 0x03
+ #define ATP867X_IO_DMAMODE_UDMA_1 0x02
+ #define ATP867X_IO_DMAMODE_UDMA_0 0x01
+ #define ATP867X_IO_DMAMODE_DISABLE 0x00
+
+ /*
+  * ATP867X PCI IDE chips.
+  */
+
+ typedef struct atp867x_controller {
+     char *name;
+     unsigned short id;
+     unsigned char rev_min;
+     unsigned char rev_max;
+ } atp867x_controller;
+
+ /* Beware using very long names in the below structures; it can make
+  * atp867x_get_info blow its buffer.
+  */
+ static struct atp867x_controller atp867x_isa_bridges[] = {
+     { "ATP867A", PCI_DEVICE_ID_ARTOP_ATP867A, 0x00, 0x04 },
+     /* B has hardware XOR, CMP. It also supports UDMA5 writes at 111 and 95
+     * MB/s, and has different characteristics than A when overclocked. So
+     * far, none of the B-only features [or overclocking support] have been
+     * implemented, so A and B are treated alike. */
+     { NULL }
+ };
+
+ static int atp867x_get_active_clocks_shifted(unsigned int ns) {
+     unsigned char clocks = ns / 30;
+
+     switch (clocks) {
+     case 0: /* Too fast */
+         clocks = 1;
+         break;
+     case 1: case 2: case 3: case 4: case 5: case 6: case 7:
+         break;
+     case 8: case 9: case 10: case 11: case 12:
+         clocks = 7;
+         break;
+     default:
+         printk(KERN_WARNING "ATP867X_IDE: active %dns is too slow. "
+            "Using default.\n", ns);
+         clocks = 0;
+     }
+ }
```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ return clocks << ATP867X_IO_PIOSPD_ACTIVE_SHIFT;
+}
+
+static int atp867x_get_recover_clocks_shifted(unsigned int ns) {
+ unsigned char clocks = ns / 30;
+
+ switch (clocks) {
+ case 0: /* Too fast */
+ clocks = 1;
+ break;
+ case 1: case 2: case 3: case 4: case 5: case 6: case 7:
+ case 8: case 9: case 10: case 11:
+ break;
+ case 12:
+ clocks = 0;
+ break;
+ case 13: case 14:
+ --clocks;
+ break;
+ case 15:
+ break;
+ default:
+ printk(KERN_WARNING "ATP867X_IDE: recover %dns is too slow. "
+ "Using default.\n", ns);
+ clocks = 15;
+ }
+ return clocks << ATP867X_IO_PIOSPD_RECOVER_SHIFT;
+}
+
+/*
+ * atp867x_set_drive() computes timing values configures the drive and
+ * the chipset to a desired transfer mode. It also can be called
+ * by upper layers.
+ */
+
+static int atp867x_set_drive(ide_drive_t *drive, unsigned char speed)
+{
+ ide_drive_t *peer = HWIF(drive)->drives + (~drive->dn & 1);
+ int channel = HWIF(drive)->channel;
+ struct ide_timing t, p;
+ int T, UT;
+ unsigned char b;
+ struct pci_dev *dev = HWIF(drive)->pci_dev;
+ unsigned char mode;
+
+ if (speed != XFER_PIO_SLOW && speed != drive->current_speed)
+ if (ide_config_drive_speed(drive, speed))
+ printk(KERN_WARNING "ide%d: Drive %d didn't accept "
+ "speed setting. Oh, well.\n", drive->dn >> 1,
+ drive->dn & 1);
+
+

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
+ T = 1000000 / system_bus_clock(); /* us/cycle */
+ UT = T/4;
+
+ ide_timing_compute(drive, speed, &t, T, UT);
+
+ if (peer->present) {
+ ide_timing_compute(peer, peer->current_speed, &p, T, UT);
+ ide_timing_merge(&p, &t, &t, IDE_TIMING_8BIT);
+ }
+
+ switch (speed) {
+ case XFER_UDMA_6:
+ mode = ATP867X_IO_DMAMODE_UDMA_6;
+ break;
+ case XFER_UDMA_5:
+ mode = ATP867X_IO_DMAMODE_UDMA_5;
+ break;
+ case XFER_UDMA_4:
+ mode = ATP867X_IO_DMAMODE_UDMA_4;
+ break;
+ case XFER_UDMA_3:
+ mode = ATP867X_IO_DMAMODE_UDMA_3;
+ break;
+ case XFER_UDMA_2:
+ mode = ATP867X_IO_DMAMODE_UDMA_2;
+ break;
+ case XFER_UDMA_1:
+ mode = ATP867X_IO_DMAMODE_UDMA_1;
+ break;
+ case XFER_UDMA_0:
+ mode = ATP867X_IO_DMAMODE_UDMA_0;
+ break;
+ case XFER_PIO_4:
+ case XFER_PIO_3:
+ case XFER_PIO_2:
+ case XFER_PIO_1:
+ case XFER_PIO_0:
+ case XFER_PIO_SLOW:
+ mode = ATP867X_IO_DMAMODE_DISABLE;
+ break;
+ case XFER_MW_DMA_2:
+ case XFER_MW_DMA_1:
+ case XFER_MW_DMA_0:
+ /* Dunno what to do about these yet. */
+ default:
+ printk(KERN_WARNING "ATP867X_IDE: Unsupported speed %#x.\n",
+ (unsigned)speed);
+ return -1;
+ }
+
+ pci_read_config_byte(dev, ATP867X_IO_DMAMODE(dev, channel), &b);
```

Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ if (drive->dn & 1) {
+ b = (b & ~ATP867X_IO_DMAMODE_SLAVE_MASK) |
+ (mode << ATP867X_IO_DMAMODE_SLAVE_SHIFT);
+ }
+ else {
+ b = (b & ~ATP867X_IO_DMAMODE_MSTR_MASK) |
+ (mode << ATP867X_IO_DMAMODE_MSTR_SHIFT);
+ }
+ pci_write_config_byte(dev, ATP867X_IO_DMAMODE(dev, channel), b);
+ if (ATP867X_IO_DMAMODE_DISABLE == mode) {
+
+ b = atp867x_get_active_clocks_shifted(t.active) |
+ atp867x_get_recover_clocks_shifted(t.recover);
+
+ /* Set up the PIO speeds. */
+ if (drive->dn & 1) {
+ pci_write_config_byte(dev,
+ ATP867X_IO_SLAVEPIOSPD(dev, channel) , b);
+ }
+ else {
+ pci_write_config_byte(dev,
+ ATP867X_IO_MSTRPIOSPD(dev, channel) , b);
+ }
+
+ b = atp867x_get_active_clocks_shifted(t.act8b) |
+ atp867x_get_recover_clocks_shifted(t.rec8b);
+ pci_write_config_byte(dev, ATP867X_IO_8BPIOSPD(dev, channel),
+ b);
+
+ }
+ if (!drive->init_speed)
+ drive->init_speed = speed;
+
+ return 0;
+}
+
+static void config_chipset_for_pio (ide_drive_t *drive)
+{
+ unsigned short eide_pio_timing[6] = {960, 480, 240, 180, 120, 90};
+ unsigned short xfer_pio = drive->id->eide_pio_modes;
+ byte timing, speed, pio;
+
+ pio = ide_get_best_pio_mode(drive, 255, 5, NULL);
+
+ if (xfer_pio > 4)
+ xfer_pio = 0;
+
+ if (drive->id->eide_pio_iordy > 0) {
+ for (xfer_pio = 5;
+ xfer_pio > 0 &&
+ drive->id->eide_pio_iordy > eide_pio_timing[xfer_pio];

```

```

+ xfer_pio--);
+ } else {
+ xfer_pio = (drive->id->eide_pio_modes & 4) ? 0x05 :
+ (drive->id->eide_pio_modes & 2) ? 0x04 :
+ (drive->id->eide_pio_modes & 1) ? 0x03 :
+ (drive->id->tPIO & 2) ? 0x02 :
+ (drive->id->tPIO & 1) ? 0x01 : xfer_pio;
+ }
+
+ timing = (xfer_pio >= pio) ? xfer_pio : pio;
+
+ switch(timing) {
+ case 4: speed = XFER_PIO_4;break;
+ case 3: speed = XFER_PIO_3;break;
+ case 2: speed = XFER_PIO_2;break;
+ case 1: speed = XFER_PIO_1;break;
+ default:
+ speed = (!drive->id->tPIO) ? XFER_PIO_0 : XFER_PIO_SLOW;
+ break;
+ }
+ (void) atp867x_set_drive(drive, speed);
+}
+
+//#ifdef CONFIG_BLK_DEV_IDEDMA
+/*
+ * This allows the configuration of ide_pci chipset registers
+ * for cards that learn about the drive's UDMA, DMA, PIO capabilities
+ * after the drive is reported by the OS. Initially designed for
+ * HPT366 UDMA chipset by HighPoint|Triones Technologies, Inc.
+ *
+ * Returns 0 if configured for dma.
+ *
+ */
+static int config_chipset_for_dma (ide_drive_t *drive)
+{
+ struct hd_driveid *id = drive->id;
+ byte speed = 0x00;
+ byte ultra66 = 1 /*eighty_ninty_three(drive)*/;
+ int rval;
+
+ config_chipset_for_pio(drive);
+ drive->init_speed = 0;
+
+ if ((drive->media != ide_disk) && (speed < XFER_SW_DMA_0))
+ return 1;
+
+ if ((id->dma_ultra & 0x0040) && ultra66) {
+ speed = XFER_UDMA_6;
+ } else if ((id->dma_ultra & 0x0020) &&
+ (!HWIF(drive)->ide_dma_bad_drive(drive)) && ultra66) {
+ speed = XFER_UDMA_5;

```

Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ } else if ((id->dma_ultra & 0x0010) &&
+ (!HWIF(drive)->ide_dma_bad_drive(drive)) && ultra66) {
+ speed = XFER_UDMA_4;
+ } else if ((id->dma_ultra & 0x0008) &&
+ (!HWIF(drive)->ide_dma_bad_drive(drive)) &&
+ ultra66) {
+ speed = XFER_UDMA_3;
+ } else if (id->dma_ultra && (!HWIF(drive)->ide_dma_bad_drive(drive))) {
+ if (id->dma_ultra & 0x0004) {
+ speed = XFER_UDMA_2;
+ } else if (id->dma_ultra & 0x0002) {
+ speed = XFER_UDMA_1;
+ } else if (id->dma_ultra & 0x0001) {
+ speed = XFER_UDMA_0;
+ }
+ } else if (id->dma_mword & 0x0004) {
+ speed = XFER_MW_DMA_2;
+ } else if (id->dma_mword & 0x0002) {
+ speed = XFER_MW_DMA_1;
+ } else if (id->dma_mword & 0x0001) {
+ speed = XFER_MW_DMA_0;
+ } else {
+ return 1;
+ }
+
+ (void) atp867x_set_drive(drive, speed);
+
+ rval = (int)((id->dma_ultra >> 14) & 3) ? 0 :
+ ((id->dma_ultra >> 11) & 7) ? 0 :
+ ((id->dma_ultra >> 8) & 7) ? 0 :
+ ((id->dma_mword >> 8) & 7) ? 0 :
+ 1);
+ return rval;
+}
+
+/* Returns zero if drive can do DMA. */
+static int atp867x_config_drive(ide_drive_t *drive) {
+ struct hd_driveid *id = drive->id;
+ int rval = 0;
+
+ if (id && (id->capability & 1) && HWIF(drive)->autodma) {
+ /* Consult the list of known "bad" drives */
+ if (HWIF(drive)->ide_dma_bad_drive(drive)) {
+ rval = 1;
+ goto fast_ata_pio;
+ }
+ rval = 1;
+ if (id->field_valid & 4) {
+ if (id->dma_ultra & 0x007F) {
+ /* Force if Capable UltraDMA */
+ rval = config_chipset_for_dma(drive);

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ if ((id->field_valid & 2) && rval)
+ goto try_dma_modes;
+ }
+ } else if (id->field_valid & 2) {
+try_dma_modes:
+ if (id->dma_mword & 0x0007) {
+ /* Force if Capable regular DMA modes */
+ rval = config_chipset_for_dma(drive);
+ if (rval)
+ goto no_dma_set;
+ }
+ } else if (HWIF(drive)->ide_dma_good_drive(drive)) {
+ if (id->eide_dma_time > 150) {
+ goto no_dma_set;
+ }
+ /* Consult the list of known "good" drives */
+ rval = config_chipset_for_dma(drive);
+ if (rval)
+ goto no_dma_set;
+ } else {
+ goto fast_ata_pio;
+ }
+ rval = HWIF(drive)->ide_dma_on(drive);
+ } else if ((id->capability & 8) || (id->field_valid & 2)) {
+fast_ata_pio:
+ rval = 1;
+no_dma_set:
+ config_chipset_for_pio(drive);
+ }
+ return rval;
+ }
+
+int atp867x_dma_host_off(ide_drive_t *drive)
+{
+ outb(inb(HWIF(drive)->dma_base) & ~1, HWIF(drive)->dma_base);
+ return 0;
+ }
+
+int atp867x_dma_host_on(ide_drive_t *drive)
+{
+ return 0;
+ }
+
+int atp867x_dma_check(ide_drive_t *drive)
+{
+ return atp867x_config_drive(drive);
+ }
+
+void atp867x_get_ports(struct pci_dev *dev, unsigned int channel,
+ unsigned long *base, unsigned long *control) {
+ *base = ATP867X_IO_PORTBASE(dev, channel);

```

Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ *control = ATP867X_IO_ALTSTATUS(dev, channel);
+}
+
+unsigned int atp867x_get_dma_base(ide_hwif_t *hwif) {
+ #ifdef NO_DMA
+ return 0;
+ #else
+ return ATP867X_IO_DMABASE(hwif->pci_dev, hwif->channel);
+ #endif
+}
+
+static struct atp867x_controller *atp867x_get_config(struct pci_dev *dev) {
+ struct atp867x_controller *config;
+
+ for (config = atp867x_isa_bridges; config->id; config++)
+ if (dev->device == config->id)
+ break;
+
+ return config->id ? config : NULL;
+}
+
+ #ifdef CONFIG_PROC_FS
+
+ #include <linux/stat.h>
+ #include <linux/proc_fs.h>
+
+ static int atp867x_get_info(char *, char **, off_t, int);
+
+ static ide_pci_host_proc_t atp867x_procs[] __initdata = {
+ {
+ .name = "atp867x",
+ .set = 1,
+ .get_info = atp867x_get_info,
+ .parent = NULL,
+ },
+ };
+ };
+
+ #define ATP867X_MAX_DEVS 8 // Don't increase without changing atp867x_get_info!
+ static struct pci_dev *atp867x_devs[ATP867X_MAX_DEVS];
+ static int atp867x_num_devs = 0;
+
+ /*
+ * ATP867X /proc entry.
+ */
+
+ #define atp867x_print(format, arg...) p += sprintf(p, format "\n" , ## arg)
+
+ static int atp867x_get_info(char *buffer, char **addr, off_t offset, int count)
+ {
+ int i;
+ char *p = buffer;

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+
+ atp867x_print("ATP867X BusMastering IDE Configuration");
+ atp867x_print("Driver Version:\t\t0.1");
+ atp867x_print("Highest DMA rate:\tUDMA133");
+ atp867x_print("PCI clock:\t\t%dmHz\n", system_bus_clock());
+ for (i=0; atp867x_num_devs > i; ++i) {
+ struct pci_dev *dev;
+ struct atp867x_controller *config;
+ unsigned char x;
+ int j;
+
+ dev = atp867x_devs[i];
+ config = atp867x_get_config(dev);
+ if (!config) {
+ printk(KERN_WARNING
+ "ATP867X_IDE: Unknown ATP867X chip\n");
+ continue;
+ }
+
+ atp867x_print("ACARD Controller %d:\t%s", i, config->name);
+ pci_read_config_byte(dev, PCI_REVISION_ID, &x);
+ atp867x_print("PCI Revision:\t\tIDE %#x", (unsigned)x);
+ for(j=0; 4 > j; ++j) {
+ atp867x_print("Channel %d:", j);
+ atp867x_print("\tRegister base\t%#08lx",
+ ATP867X_IO_PORTBASE(dev, j));
+ atp867x_print("\tAlt Status\t%#08lx",
+ ATP867X_IO_ALTSTATUS(dev, j));
+ atp867x_print("\tBM-DMA at\t%#08lx",
+ ATP867X_IO_DMABASE(dev, j));
+ }
+
+ atp867x_print("Enabled:\t\t%s\n",
+ config ? "YES" : "NO");
+ }
+
+ /* Had better be less than 4K. Currently it's just under 500 bytes/dev,
+ * but check it if you change the above, lengthen config->name, or allow
+ * more than 8 devices. */
+ return p - buffer;
+ }
+
+ #endif
+
+ /*
+ * The initialization callback. Here we determine the IDE chip type
+ * and initialize its drive independent registers.
+ */
+
+ unsigned int __init init_chipset_atp867x(struct pci_dev *dev, const char *name)
+ {

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
+ unsigned char t, v;
+ struct atp867x_controller *config;
+
+ /*
+ * Check whether interfaces are enabled.
+ */
+
+ pci_read_config_byte(dev, PCI_COMMAND, &v);
+ if (!(v & PCI_COMMAND_IO)) {
+ printk("ATP867X_IDE: Disabled.\n");
+ return 0;
+ }
+
+ /*
+ * Check system bus clock.
+ */
+
+ if (system_bus_clock() < 20 || system_bus_clock() > 80) {
+ printk(KERN_WARNING "ATP867X_IDE: User given PCI clock speed "
+ "impossible (%d); giving up!\n", system_bus_clock());
+ return 0;
+ }
+
+ /*
+ * Verify version number.
+ */
+
+ config = atp867x_get_config(dev);
+ if (!config) {
+ printk(KERN_WARNING "ATP867X_IDE: Unknown ATP867X chip\n");
+ return 0;
+ }
+
+ pci_read_config_byte(dev, PCI_REVISION_ID, &t);
+ if (t < config->rev_min || t > config->rev_max)
+ printk(KERN_WARNING "%s: Unexpected version 0x%02x.\n",
+ config->name, (unsigned)t);
+
+ /*
+ * Check 80-wire cable presence.
+ */
+
+ /* Just assume it for now. */
+
+ /*
+ * Print the boot message.
+ */
+
+ pci_read_config_byte(dev, PCI_REVISION_ID, &t);
+ printk(KERN_INFO "ATP867X_IDE: %s (rev %02x) IDE %s controller on "
+ "pci%s\n", config->name, t, "UDMA133", dev->slot_name);
```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```
+
+/*
+ * Look up the IO base address.
+ */
+
+ printk(KERN_INFO "%s: IO base address 0x%08lx.\n", config->name,
+ pci_resource_start(dev, 0));
+
+ #ifdef CONFIG_PROC_FS
+ /*
+ * Setup /proc/ide/atp867x entry.
+ */
+
+ if (ATP867X_MAX_DEVS > atp867x_num_devs) {
+ atp867x_devs[atp867x_num_devs++] = dev;
+ if (1 == atp867x_num_devs) {
+ ide_pci_register_host_proc(atp867x_procs);
+ }
+ }
+ #endif
+
+ return dev->irq;
+}
+
+ /* Are we enabled? */
+ unsigned int __init ata66_atp867x(ide_hwif_t *hwif)
+ {
+ unsigned char v;
+
+ pci_read_config_byte(hwif->pci_dev, PCI_COMMAND, &v);
+ return v & PCI_COMMAND_IO;
+ }
+
+ /*
+ * atp867x_tune_drive() is a callback from upper layers for
+ * PIO-only tuning.
+ */
+
+ static void atp867x_tune_drive(ide_drive_t *drive, unsigned char pio)
+ {
+ if (!ata66_atp867x(HWIF(drive)))
+ return;
+
+ if (pio == 255)
+ atp867x_set_drive(drive,
+ ide_find_best_mode(drive, XFER_PIO | XFER_EPIO));
+ else
+ atp867x_set_drive(drive, XFER_PIO_0 + MIN(pio, 4));
+ }
+
+ void __init init_hwif_atp867x(ide_hwif_t *hwif)
```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+{
+ int i;
+
+ hwif->tuneproc = &atp867x_tune_drive;
+ hwif->speedproc = &atp867x_set_drive;
+
+ if (hwif->dma_base) {
+ hwif->ide_dma_host_off = &atp867x_dma_host_off;
+ hwif->ide_dma_host_on = &atp867x_dma_host_on;
+ hwif->ide_dma_check = &atp867x_dma_check;
+ hwif->udma_four = 1;
+ hwif->ultra_mask = 0x7f;
+ }
+ for (i = 0; i < 2; i++) {
+ hwif->drives[i].io_32bit = 1;
+ hwif->drives[i].autotune = 1;
+ hwif->drives[i].dn = hwif->channel * 2 + i;
+ hwif->drives[i].autodma = hwif->autodma;
+ }
+ }
+
+/*
+ * We allow the BM-DMA driver to only work on enabled interfaces.
+ */
+
+void __init init_dma_atp867x(ide_hwif_t *hwif, unsigned long dmabase)
+{
+ #ifndef NO_DMA
+ if (ata66_atp867x(hwif)) {
+ ide_setup_dma(hwif, dmabase, 8);
+ }
+ #endif
+ }
+
+static ide_pci_device_t atp867x_chipsets[] __devinitdata = {
+ { /* 0 */
+ .vendor = PCI_VENDOR_ID_ARTOP,
+ .device = PCI_DEVICE_ID_ARTOP_ATP867A,
+ .name = "ATP867A_IDE",
+ .init_chipset = init_chipset_atp867x,
+ .init_iops = NULL,
+ .init_hwif = init_hwif_atp867x,
+ .init_dma = init_dma_atp867x,
+ .channels = 4,
+ .autodma = AUTODMA,
+ .enablebits = {{0x40,0x01,0x01}, {0x40,0x01,0x01}},
+ .bootable = ON_BOARD,
+ .extra = 0,
+ },{
+ .vendor = 0,
+ .device = 0,

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ .channels = 0,
+ .bootable = EOL,
+ }
+};
+
+static int __devinit atp867x_init_one(struct pci_dev *dev,
+ const struct pci_device_id *id)
+{
+ ide_pci_device_t *d = &atp867x_chipsets[id->driver_data];
+ if (dev->device != d->device)
+ BUG();
+ ide_setup_pci_device(dev, d);
+ MOD_INC_USE_COUNT;
+ return 0;
+}
+
+static struct pci_device_id atp867x_pci_tbl[] __devinitdata = {
+ { PCI_VENDOR_ID_ARTOP, PCI_DEVICE_ID_ARTOP_ATP867A,
+ PCI_ANY_ID, PCI_ANY_ID, 0, 0, 0},
+ { 0, },
+};
+
+static struct pci_driver driver = {
+ .name = "Artop 867x IDE",
+ .id_table = atp867x_pci_tbl,
+ .probe = atp867x_init_one,
+};
+
+static int atp867x_ide_init(void)
+{
+ return ide_pci_register_driver(&driver);
+}
+
+static void atp867x_ide_exit(void)
+{
+ ide_pci_unregister_driver(&driver);
+}
+module_init(atp867x_ide_init);
+module_exit(atp867x_ide_exit);
+
+MODULE_AUTHOR("Eric Uhrhane, Vojtech Pavlik, Michel Aubry, Jeff Garzik, Andre Hedrick");
+MODULE_DESCRIPTION("PCI driver module for Artop/Acard 867x IDE");
+MODULE_LICENSE("GPL");
+
+EXPORT_NO_SYMBOLS;
diff -rNBdU 3 linux-2.4.26.orig/drivers/ide/pci/atp867.h linux-2.4.26/drivers/ide/pci/atp867.h
--- linux-2.4.26.orig/drivers/ide/pci/atp867.h 1969-12-31 16:00:00.000000000 -0800
+++ linux-2.4.26/drivers/ide/pci/atp867.h 2004-04-14 14:17:34.000000000 -0700
@@ -0,0 +1,11 @@
+#ifndef ATP867X_H
+#define ATP867X_H

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+
+#include <linux/pci.h>
+#include <linux/ide.h>
+
+extern unsigned int atp867x_get_dma_base(ide_hwif_t *hwif);
+extern void atp867x_get_ports(struct pci_dev *dev, unsigned int channel,
+ unsigned long *base, unsigned long *control);
+
+#endif /* ATP867X_H */
diff -rNBdU 3 linux-2.4.26.orig/drivers/ide/pci/Makefile linux-2.4.26/drivers/ide/pci/Makefile
--- linux-2.4.26.orig/drivers/ide/pci/Makefile 2004-04-14 06:05:29.000000000 -0700
+++ linux-2.4.26/drivers/ide/pci/Makefile 2004-04-14 14:20:59.000000000 -0700
@@ -9,6 +9,7 @@
obj-$(CONFIG_BLK_DEV_ALI15X3) += alim15x3.o
obj-$(CONFIG_BLK_DEV_AMD74XX) += amd74xx.o
obj-$(CONFIG_BLK_DEV_ATIIXP) += atiixp.o
+obj-$(CONFIG_BLK_DEV_ATP867X) += atp867.o
obj-$(CONFIG_BLK_DEV_CMD640) += cmd640.o
obj-$(CONFIG_BLK_DEV_CMD64X) += cmd64x.o
obj-$(CONFIG_BLK_DEV_CS5530) += cs5530.o
diff -rNBdU 3 linux-2.4.26.orig/drivers/ide/setup-pci.c linux-2.4.26/drivers/ide/setup-pci.c
--- linux-2.4.26.orig/drivers/ide/setup-pci.c 2003-08-25 04:44:41.000000000 -0700
+++ linux-2.4.26/drivers/ide/setup-pci.c 2004-04-14 14:17:35.000000000 -0700
@@ -32,6 +32,9 @@
#include <asm/io.h>
#include <asm/irq.h>

+#ifdef CONFIG_BLK_DEV_ATP867X
+#include "pci/atp867.h"
+#endif

/**
 * ide_match_hwif - match a PCI IDE against an ide_hwif
@@ -183,6 +186,11 @@
second_chance_to_dma:
#endif /* CONFIG_BLK_DEV_IDEDMA_FORCED */

+#ifdef CONFIG_BLK_DEV_ATP867X
+ if ((dev->device == PCI_DEVICE_ID_ARTOP_ATP867A))
+ return atp867x_get_dma_base(hwif);
+#endif
+
+ if (hwif->mmio && hwif->dma_base)
+ return hwif->dma_base;
+ else if (hwif->mate && hwif->mate->dma_base) {
@@ -416,12 +424,23 @@
unsigned long ctl = 0, base = 0;
ide_hwif_t *hwif;

+#ifdef CONFIG_BLK_DEV_ATP867X
+ /* pci_resource_start won't work for this chip. */

```

## Linux-Kernel: [PATCH] ATP867X PCI IDE driver

```

+ if ((dev->device == PCI_DEVICE_ID_ARTOP_ATP867A)) {
+ atp867x_get_ports(dev, port, &base, &ctl);
+ }
+ else {
+ #endif
+
+     /* Possibly we should fail if these checks report true */
+     ide_pci_check_iomem(dev, d, 2*port);
+     ide_pci_check_iomem(dev, d, 2*port+1);
+
+     ctl = pci_resource_start(dev, 2*port+1);
+     base = pci_resource_start(dev, 2*port);
+ #ifdef CONFIG_BLK_DEV_ATP867X
+ }
+ #endif
+     if ((ctl && !base) || (base && !ctl)) {
+         printk(KERN_ERR "%s: inconsistent baseregs (BIOS) "
+             "for port %d, skipping\n", d->name, port);
@@ -627,9 +646,19 @@
+     * Set up the IDE ports
+     */

- for (port = 0; port <= 1; ++port) {
- ide_pci_enablebit_t *e = &(d->enablebits[port]);
+ for (port = 0; port <= d->channels; ++port) {
+ /* eju: HACK to support ATP867X, which really only has one
+ * enablebits entry [entered twice], but has 4 ports. Currently
+ * no other driver has more than 2 ports. */
+ ide_pci_enablebit_t *e = &(d->enablebits[port % 2]);

+ /* eju: It's not clear that we need to do anything here, but
+ * this does no harm, and pairs up the ports neatly. I'm not
+ * sure what proc_ide_write_config will do, though. */
+ if (2 == port) {
+ mate = NULL;
+ }
+
+     /*
+      * If this is a Promise FakeRaid controller,
+      * the 2nd controller will be marked as
diff -rNBdU 3 linux-2.4.26.orig/include/linux/pci_ids.h linux-2.4.26/include/linux/pci_ids.h
--- linux-2.4.26.orig/include/linux/pci_ids.h 2004-04-14 06:05:40.000000000 -0700
+++ linux-2.4.26/include/linux/pci_ids.h 2004-04-14 14:17:35.000000000 -0700
@@ -1292,6 +1292,7 @@
#define PCI_DEVICE_ID_ARTOP_ATP860R 0x0007
#define PCI_DEVICE_ID_ARTOP_ATP865 0x0008
#define PCI_DEVICE_ID_ARTOP_ATP865R 0x0009
+ #define PCI_DEVICE_ID_ARTOP_ATP867A 0x000A
#define PCI_DEVICE_ID_ARTOP_AEC7610 0x8002
#define PCI_DEVICE_ID_ARTOP_AEC7612UW 0x8010
#define PCI_DEVICE_ID_ARTOP_AEC7612U 0x8020

```

Linux-Kernel: [PATCH] ATP867X PCI IDE driver

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@vger.kernel.org](mailto:majordomo@vger.kernel.org)

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>