

[bk patch] PCI MWI updates/fixes

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-08/8194.html>

From: Jeff Garzik (jgarzik_at_pobox.com)

Date: 08/31/03

Date: Sat, 30 Aug 2003 23:00:05 -0400
To: marcelo@parcellfarce.linux.theplanet.co.uk

Marcelo, please do a

bk pull bk://kernel.bkbits.net/jgarzik/misc-2.4

This will update the following files:

```
arch/i386/kernel/pci-i386.c | 11 ++++++++
arch/i386/kernel/pci-i386.h | 2 +
arch/i386/kernel/pci-pc.c | 2 +
arch/i386/kernel/pci-visws.c | 1
drivers/pci/pci.c | 55 ++++++-----
5 files changed, 45 insertions(+), 26 deletions(-)
```

through these ChangeSets:

<ink@jurassic.park.msu.ru> (03/08/30 1.1102)
[PCI] update Memory-Write-Invalidate (MWI) transaction support

Sync up PCI MWI code with 2.5:

- no changes required for drivers or architectures with HAVE_ARCH_PCI_MWI;
- do respect BIOS settings: if the cacheline size is multiple of that we have expected, assume that this is on purpose;
- assume cacheline size of 32 bytes for all x86s except K7/K8 and P4. Actually it's good for 386/486s as quite a few PCI devices do not support smaller values.

```
diff -Nru a/arch/i386/kernel/pci-i386.c b/arch/i386/kernel/pci-i386.c
--- a/arch/i386/kernel/pci-i386.c Sat Aug 30 22:57:53 2003
+++ b/arch/i386/kernel/pci-i386.c Sat Aug 30 22:57:53 2003
@@ -295,6 +295,17 @@
     }
 }

+void __init pcibios_set_cacheline_size(void)
+{
```

Linux-Kernel: [bk patch] PCI MWI updates/fixes

```

+ struct cpuinfo_x86 *c = &boot_cpu_data;
+
+ pci_cache_line_size = 32 >> 2;
+ if (c->x86 >= 6 && c->x86_vendor == X86_VENDOR_AMD)
+ pci_cache_line_size = 64 >> 2; /* K7 & K8 */
+ else if (c->x86 > 6)
+ pci_cache_line_size = 128 >> 2; /* P4 */
+}
+
void __init pcibios_resource_survey(void)
{
    DBG("PCI: Allocating resources\n");
diff -Nru a/arch/i386/kernel/pci-i386.h b/arch/i386/kernel/pci-i386.h
--- a/arch/i386/kernel/pci-i386.h Sat Aug 30 22:57:53 2003
+++ b/arch/i386/kernel/pci-i386.h Sat Aug 30 22:57:53 2003
@@ -27,8 +27,10 @@
/* pci-i386.c */

extern unsigned int pcibios_max_latency;
+extern u8 pci_cache_line_size;

void pcibios_resource_survey(void);
+void pcibios_set_cacheline_size(void);
int pcibios_enable_resources(struct pci_dev *, int);

/* pci-pc.c */
diff -Nru a/arch/i386/kernel/pci-pc.c b/arch/i386/kernel/pci-pc.c
--- a/arch/i386/kernel/pci-pc.c Sat Aug 30 22:57:53 2003
+++ b/arch/i386/kernel/pci-pc.c Sat Aug 30 22:57:53 2003
@@ -1428,6 +1428,8 @@
    return;
}

+ pcibios_set_cacheline_size();
+
    printk(KERN_INFO "PCI: Probing PCI hardware\n");
#ifdef CONFIG_ACPI_PCI
    if (use_acpi_pci && !acpi_pci_irq_init()) {
diff -Nru a/arch/i386/kernel/pci-visws.c b/arch/i386/kernel/pci-visws.c
--- a/arch/i386/kernel/pci-visws.c Sat Aug 30 22:57:53 2003
+++ b/arch/i386/kernel/pci-visws.c Sat Aug 30 22:57:53 2003
@@ -119,6 +119,7 @@
{
    unsigned int sec_bus = li_pcib_read16(LI_PCI_BUSNUM) & 0xff;

+ pcibios_set_cacheline_size();
    printk("PCI: Probing PCI hardware on host buses 00 and %02x\n", sec_bus);
    pci_scan_bus(0, &visws_pci_ops, NULL);
    pci_scan_bus(sec_bus, &visws_pci_ops, NULL);
diff -Nru a/drivers/pci/pci.c b/drivers/pci/pci.c
--- a/drivers/pci/pci.c Sat Aug 30 22:57:53 2003

```

Linux–Kernel: [bk patch] PCI MWI updates/fixes

```
+++ b/drivers/pci/pci.c Sat Aug 30 22:57:53 2003
@@ -907,8 +907,12 @@
     pcibios_set_master(dev);
 }

+#ifndef HAVE_ARCH_PCI_MWI
+/* This can be overridden by arch code. */
+u8 pci_cache_line_size = L1_CACHE_BYTES >> 2;
+
+/**
+ * pdev_set_mwi – arch helper function for pcibios_set_mwi
+ * pci_generic_prep_mwi – helper function for pci_set_mwi
+ * @dev: the PCI device for which MWI is enabled
+ *
+ * Helper function for implementation the arch–specific pcibios_set_mwi
@@ -918,34 +922,33 @@
+ * RETURNS: An appropriate –ERRNO error value on error, or zero for success.
+ */
+int
+pdev_set_mwi(struct pci_dev *dev)
+pci_generic_prep_mwi(struct pci_dev *dev)
+{
+ int rc = 0;
+ u8 cache_size;
+ u8 cacheline_size;
+
+ if (!pci_cache_line_size)
+ return –EINVAL; /* The system doesn't support MWI. */
+
+ /* Validate current setting: the PCI_CACHE_LINE_SIZE must be
+ equal to or multiple of the right value. */
+ pci_read_config_byte(dev, PCI_CACHE_LINE_SIZE, &cacheline_size);
+ if (cacheline_size >= pci_cache_line_size &&
+ (cacheline_size % pci_cache_line_size) == 0)
+ return 0;
+
+ /* Write the correct value. */
+ pci_write_config_byte(dev, PCI_CACHE_LINE_SIZE, pci_cache_line_size);
+ /* Read it back. */
+ pci_read_config_byte(dev, PCI_CACHE_LINE_SIZE, &cacheline_size);
+ if (cacheline_size == pci_cache_line_size)
+ return 0;
+
+ /*
+ * Looks like this is necessary to deal with on all architectures,
+ * even this %$#%$# N440BX Intel based thing doesn't get it right.
+ * Ie. having two NICs in the machine, one will have the cache
+ * line set at boot time, the other will not.
+ */
+ pci_read_config_byte(dev, PCI_CACHE_LINE_SIZE, &cache_size);
+ cache_size <<= 2;
```

Linux-Kernel: [bk patch] PCI MWI updates/fixes

```
- if (cache_size != SMP_CACHE_BYTES) {
- printk(KERN_WARNING "PCI: %s PCI cache line size set incorrectly (%i bytes) by BIOS/FW.\n",
- dev->slot_name, cache_size);
- if (cache_size > SMP_CACHE_BYTES) {
- printk("PCI: %s cache line size too large - expecting %i.\n", dev->slot_name, SMP_CACHE_BYTES);
- rc = -EINVAL;
- } else {
- printk("PCI: %s PCI cache line size corrected to %i.\n", dev->slot_name, SMP_CACHE_BYTES);
- pci_write_config_byte(dev, PCI_CACHE_LINE_SIZE,
- SMP_CACHE_BYTES >> 2);
- }
- }
+ printk(KERN_WARNING "PCI: cache line size of %d is not supported "
+ "by device %s\n", pci_cache_line_size << 2, dev->slot_name);

- return rc;
+ return -EINVAL;
}
+ #endif /* !HAVE_ARCH_PCI_MWI */

/**
 * pci_set_mwi - enables memory-write-invalidate PCI transaction
@@ -966,7 +969,7 @@
# ifdef HAVE_ARCH_PCI_MWI
    rc = pcibios_set_mwi(dev);
# else
- rc = pdev_set_mwi(dev);
+ rc = pci_generic_prep_mwi(dev);
# endif

    if (rc)
-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to 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/>