

[PATCH]PCI Express Port Bus Driver

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-12/3550.html>

From: long (tlnghuyen_at_snoqualmie.dp.intel.com)

Date: 12/15/04

Date: Wed, 15 Dec 2004 14:27:25 -0800

To: linux-kernel@vger.kernel.org

A PCI Express Port is a logical PCI-PCI Bridge structure. There are two types of PCI Express Port: the Root Port and the Switch Port. A PCI Express Port can provide up to four distinct services, depending on its port type. These services include native hotplug support, power management event support, advanced error reporting support, and virtual channel support.

In existing Linux kernels, the Linux Device Driver Model allows a physical device to be handled by only a single driver. The PCI Express Port is a PCI-PCI Bridge device with multiple distinct services. To maintain a clean and simple solution each service may have its own software service driver. In this case several service drivers will compete for a single PCI-PCI Bridge device. For example, if the PCI Express Root Port native hotplug service driver is loaded first, it claims a PCI-PCI Bridge Root Port. The kernel therefore does not load other service drivers for that Root Port. In other words, it is impossible to have multiple service drivers loaded and run on a PCI-PCI Bridge device simultaneously using the current driver model.

To enable multiple service drivers running simultaneously requires having a PCI Express Port Bus driver, which manages all populated PCI Express Ports and distributes all provided service requests to the corresponding service drivers as required.

This patch, which is based on linux kernel 2.6.10-rc3, includes the followings:

- PCI Express Port Bus driver source
- Modification of PCI Express Native Hot-Plug driver to support PCI Express Port Bus Model
- PCIEBUS-HOWTO.txt

Please send us any suggestions, feedback, comments or alternative designs.

Signed-off-by: T. Long Nguyen <tom.l.nguyen@intel.com>

Linux-Kernel: [PATCH]PCI Express Port Bus Driver

```
diff -urpN linux-2.6.10-rc3/arch/i386/Kconfig patch-2.6.10-rc3-pbdhp/arch/i386/Kconfig
--- linux-2.6.10-rc3/arch/i386/Kconfig 2004-12-03 16:52:47.000000000 -0500
+++ patch-2.6.10-rc3-pbdhp/arch/i386/Kconfig 2004-12-10 14:06:05.000000000 -0500
@@ -1117,6 +1117,8 @@ config PCI_MMCONFIG
    select ACPI_BOOT
    default y

+source "drivers/pcieport/Kconfig"
+
source "drivers/pci/Kconfig"

config ISA
diff -urpN linux-2.6.10-rc3/Documentation/PCIEBUS-HOWTO.txt
patch-2.6.10-rc3-pbdhp/Documentation/PCIEBUS-HOWTO.txt
--- linux-2.6.10-rc3/Documentation/PCIEBUS-HOWTO.txt 1969-12-31 19:00:00.000000000 -0500
+++ patch-2.6.10-rc3-pbdhp
--- linux-2.6.10-rc3/drivers/pcieport/portdrv.h 1969-12-31 19:00:00.000000000 -0500
+++ patch-2.6.10-rc3-pbdhp/drivers/pcieport/portdrv.h 2004-12-10 15:17:31.000000000 -0500
@@ -0,0 +1,42 @@
+/*
+ * File: portdrv.h
+ * Purpose: PCI Express Port Bus Driver's Internal Data Structures
+ *
+ * Copyright (C) 2004 Intel
+ * Copyright (C) Tom Long Nguyen (tom.l.nguyen@intel.com)
+ */
+
+#ifndef _PORTDRV_H_
+#define _PORTDRV_H_
+
+#if !defined(PCI_CAP_ID_PME)
+#define PCI_CAP_ID_PME 1
+#endif
+
+#if !defined(PCI_CAP_ID_EXP)
+#define PCI_CAP_ID_EXP 0x10
+#endif
+
+#define PORT_TYPE_MASK 0xf
+#define PORT_TO_SLOT_MASK 0x100
+#define SLOT_HP_CAPABLE_MASK 0x40
+#define PCIE_CAPABILITIES_REG 0x2
+#define PCIE_SLOT_CAPABILITIES_REG 0x14
+#define PCIE_PORT_DEVICE_MAXSERVICES 4
+#define PCI_CFG_SPACE_SIZE 256
+
+#define get_descriptor_id(type, service) (((type - 4) << 4) | service)
+
+extern struct bus_type pcie_port_bus_type;
+extern struct device_driver pcieport_generic_driver;
```

Linux-Kernel: [PATCH]PCI Express Port Bus Driver

```
+extern int pcie_port_device_probe(struct pci_dev *dev);
+extern int pcie_port_device_register(struct pci_dev *dev);
+#ifdef CONFIG_PM
+extern int pcie_port_device_suspend(struct pcie_device *dev, u32 state);
+extern int pcie_port_device_resume(struct pcie_device *dev);
+#endif
+extern void pcie_port_device_remove(struct pcie_device *dev);
+extern void pcie_port_bus_register(void);
+extern void pcie_port_bus_unregister(void);
+
+#endif /* _PORTDRV_H_ */
diff -urpN linux-2.6.10-rc3/drivers/pcieport/portdrv_pci.c
patch-2.6.10-rc3-pbdhp/drivers/pcieport/portdrv_pci.c
--- linux-2.6.10-rc3/drivers/pcieport/portdrv_pci.c 1969-12-31 19:00:00.000000000 -0500
+++ patch-2.6.10-rc3-pbdhp/drivers/pcieport/portdrv_pci.c 2004-12-10 15:31:35.000000000 -0500
@@ -0,0 +1,138 @@
+/*
+ * File: portdrv_pci.c
+ * Purpose: PCI Express Port Bus Driver
+ *
+ * Copyright (C) 2004 Intel
+ * Copyright (C) Tom Long Nguyen (tom.l.nguyen@intel.com)
+ */
+
+#include <linux/module.h>
+#include <linux/pci.h>
+#include <linux/kernel.h>
+#include <linux/errno.h>
+#include <linux/pm.h>
+#include <linux/init.h>
+#include <linux/pcieport_if.h>
+
+#include "portdrv.h"
+
+/*
+ * Version Information
+ */
+#define DRIVER_VERSION "v1.0"
+#define DRIVER_AUTHOR "tom.l.nguyen@intel.com"
+#define DRIVER_DESC "PCIE Port Bus Driver"
+MODULE_AUTHOR(DRIVER_AUTHOR);
+MODULE_DESCRIPTION(DRIVER_DESC);
+MODULE_LICENSE("GPL");
+
+/* global data */
+static const char device_name[] = "pcieport-driver";
+
+/*
+ * pcie_portdrv_probe - Probe PCI-Express port devices
+ * @dev: PCI-Express port device being probed
+ */
```

Linux-Kernel: [PATCH]PCI Express Port Bus Driver

```
+ * If detected invokes the pcie_port_device_register() method for
+ * this port device.
+ *
+ */
+static int __devinit pcie_portdrv_probe (struct pci_dev *dev,
+ const struct pci_device_id *id )
+{
+ int status;
+
+ status = pcie_port_device_probe(dev);
+ if (status)
+ return status;
+
+ if (pci_enable_device(dev) < 0)
+ return -ENODEV;
+
+ pci_set_master(dev);
+ if (!dev->irq) {
+ printk(KERN_WARNING
+ "%s->Dev[%04x:%04x] has invalid IRQ. Check vendor BIOS\n",
+ __FUNCTION__, dev->device, dev->vendor);
+ }
+ if (pcie_port_device_register(dev))
+ return -ENOMEM;
+
+ return 0;
+}
+
+static void pcie_portdrv_remove (struct pci_dev *dev)
+{
+ struct pcie_device *pciedev;
+
+ pciedev = (struct pcie_device *)pci_get_drvdata(dev);
+ if (pciedev) {
+ pcie_port_
```