Introduction

PCIe® Hot-swap is not to be confused with PCIe Hot-plug. PCIe Hot-plug is derived from revision 1.0 of the Standard Hot Plug Controller specification for PCI[1]. This specification describes the methodology by which PCIe endpoint devices may be added to or removed from an operational system without compromising the operational state of the system. The specification defines a number of standard hardware registers and signals, such as Attention Button and Attention Indicator, that allow the development of a common Hot-plug device driver. PCIe Hot-plug is designed as a "no-unexpected" or "graceful" methodology, i.e., the user is not permitted to install or remove a PCIe endpoint device without first notifying the system software.

There is no standard for PCIe Hot-swap; it is system-dependent and usually implemented by the system vendor. PCIe Hot-swap allows an endpoint or one or more PCIe switches with one or more endpoints to be inserted or removed from a PCIe system gracefully or unexpectedly.

This application note discusses some of the issues and firmware considerations as they relate to implementing PCIe Hot-swap on standard PC based systems.

Hot-Plug

The Hot-Plug Specification describes the methodology by which an endpoint may be replaced in a running system without having to turn the system off.

In order to facilitate such a procedure, additional hardware is required to control power to the slot (refer to Figure 1), to relay user requests to the host operating system, and for the operating system to relay status information back to the user. The basic procedure for Hot-plug is as follows:

- User presses attention button to inform operating system that card is to be removed.
- Operating system places card driver in quiescent state, powers down slot, and informs user that card is safe to remove.
- User removes and optionally replaces card.

The procedure for installing a new endpoint is essentially the reverse of the removal procedure. Not only is this a lengthy procedure from the user point of view but the additional hardware required to support Hot-plug also makes it impractical in price sensitive PC markets.
Hot-Swap

PCIe Hot-swap allows an endpoint or one or more PCIe switches with one or more endpoints to be inserted or removed from a PCIe system gracefully or unexpectedly without special consideration. In a Hot-swap environment, PCIe endpoint devices are connected to PCIe switches or to the root complex utilizing PCIe cables. The fact that these cables only carry data and that the PCIe transceivers are Hot-plug safe is what makes it possible for Hot-swap to be realized on standard “off the shelf” PC based architectures without the need for additional hardware.

Firmware Considerations

Although the principles of Hot-swap are fairly straightforward, there are a number of factors that need to be taken into consideration when implementing such a system on PC based architectures, the most important of which is the BIOS.

PC BIOS

It’s important to note that it is the PC BIOS and not the operating system that performs the initial PCIe bus enumeration and device resource allocation. The operating system merely scans the PCIe devices when it boots and reserves the memory and I/O resources specified in the endpoint Base Address Registers (BARs).

In the past, it was only necessary to access 256 bytes of PCI configuration space. PC architectures utilized registers located at I/O ports 0xCF8/0xCFC to first select the device and configuration space address to be accessed and then to read or write data to the specified device. With the introduction of PCIe, the configuration space was extended to 4KB in size. Because only 8 bits were assigned to the PCI configuration space address register and to maintain backward compatibility, a new method of accessing the PCIe extended configuration space was required. The introduction of “Memory Mapped Configuration” (MMCFG) access satisfied this requirement. Instead of using I/O mapped address and data registers, MMCFG maps the actual configuration space for each endpoint into the system address map.

The BIOS sets up the MMCFG space used to access the PCIe configuration space. On a standard PC, the MMCFG is typically a 256MB window that starts at physical address 0xE0000000. This allows access to 4KB of configuration space for each endpoint device up to the maximum number of devices supported by PCIe (8 functions x 32 devices x 256 buses x 4KB of configuration space = 256MB).

The exact location and the bus number range that may be accessed via the MMCFG window is communicated from the BIOS to the OS by means of the ACPI\[2\] MCFG\[3\] structure.
**BIOS Limitations**

In general, the PC BIOS was not written with Hot-swap or even Hot-plug in mind. It is assumed that, once booted, the hardware configuration will remain static. Because of this assumption, a number of complexities are introduced by the BIOS when implementing Hot-swap.

First, as can be seen in Figure 2, only one bus number is assigned to each root port in the system when it boots.

![Figure 2 Bus Number Assignments After Boot](image)

If a PCIe switch were to be connected to root port 02.0 after the system boots, there will be an insufficient number of bus numbers to assign to the newly connected switch. See Figure 3.

![Figure 3 Insufficient Bus Numbers](image)

Also, no memory or I/O resources are allocated to empty slots and no additional resources are allocated to populated slots.

![Figure 4 Insufficient Resources](image)
This causes problems when a device is connected to a root port that was previously unpopulated or when a device is replaced with a device that requires more resources. Finally, the BIOS on some PCs limits the size of the MMCFG window based on the number of buses detected when the system booted[@]. This restricts access to devices on bus numbers higher than the maximum bus number discovered.

All of these limitations must be overcome in order for Hot-swap or even Hot-plug to work on PC based architectures. The ideal situation would be to prevent the BIOS from enumerating the PCIe buses and assigning memory and I/O resources to devices on specified root ports, and instead to allow the operating system to perform these tasks. In practice, this is not possible without modification to the BIOS, so alternate methods must be developed.

**Bus Enumeration and Resource Allocation**

When a new device is added to a running system, there are two methods by which the lack of bus numbers, memory, and I/O resources assigned by the BIOS can be addressed.

The first approach is for the operating system to stop all devices and place the device drivers for all devices sharing the same root port as the device being added into a quiescent state. Once in this state, the entire branch of the PCIe bus, starting at the root port, may be re-enumerated and have new memory and I/O resources allocated before restarting the device drivers. This approach has an advantage in the desktop PC environment where the user may want to connect/disconnect any number of devices at any time. The downside is that it may be undesirable to stop all devices on a branch of the PCIe bus just to add one new device. Furthermore, if for any reason a device cannot be stopped and its driver placed into a quiescent state, it is impossible to install the new device since the bus enumeration and resource allocation procedure is only possible when all devices on the bus have been stopped.

The second approach, depicted in Figure 5, is for the operating system to assign additional bus numbers, memory, and I/O resources to each port on the PCIe bus. Effectively, the operating system must re-enumerate the PCIe buses.

![Figure 5 Additional Bus Number Assignment](image)

When a new device, such as the switch shown in Figure 5, is connected, the operating system will be able to assign the pre-allocated resources to the new device. This approach has the obvious advantage of not having to stop all devices on the branch of the bus to which the new device is being connected. The downside is that the system configuration must be known ahead of time so the operating system can be configured to reserve the required resources to support the maximum system configuration. For this reason, this approach is better suited for systems where the maximum system configuration is known ahead of time.
MMCFG Access

Being able to re-enumerate PCIe buses when new devices are connected to the system is one thing. Being able to access these devices after the buses have been re-enumerated is something else altogether.

Some operating system APIs will restrict access to the MMCFG window based on the bus number being accessed. For example, assume that when the BIOS enumerated the PCIe buses it detected 5 buses, bus #0-4. The operating system then re-enumerated the PCIe buses adding an additional 4 bus numbers, bus #5-8. When an attempt is made to access devices on bus #5-8, because these bus numbers are outside the bus number range specified initially by the BIOS, the operating system may revert to the legacy mode of PCI configuration space access (I/O ports 0xCF8/0xCFC). Using this method to access the PCIe configuration space will limit configuration space access to the first 256 bytes of configuration address space. One operating system known to behave in this manner is Linux (2.6.18 Kernel). In this case, it is important that the operating system knows the highest PCIe bus number in the system in order for the configuration space access API functions to access the extended configuration space.

It is also possible for the BIOS to limit the size of the MMCFG window based on the number of PCIe buses discovered during the initial enumeration process. It is the responsibility of the operating system to resize the MMCFG window to match the actual system configuration as devices are added. Such behavior has been observed on AMD K8 based PCs loaded with the Phoenix AwardBIOS.

Device Removal and Insertion

Since there is no attention button for the user to press to inform the host operating system that a device is to be removed or inserted, automatic device removal/insertion detection must be performed. Detecting the insertion or removal of a PCIe device is accomplished in one of three ways.

1. By checking the Device Link Layer Link Active (DLLLA) bit in the PCIe Link Status register of the down-stream ports [4].
2. By reading a vendor-specific link status register of the down-stream ports[5].
3. By attempting to read the device/vendor IDs of the device connected to the down-stream ports[4].

Since not all devices support DLLLA status reporting, if after checking the Link Capabilities it is determined that the down-stream port does not support DLLLA reporting and if no vendor specific link status register is provided, method #3 should be used to detect the presence/absence of an endpoint device.

Interrupts

Interrupts are another area where care should be taken when re-enumerating PCIe buses. In some cases, operating systems associate specific CPU interrupt sources with PCIe root port bus numbers. If the PCIe root port bus numbers change, as they would in the case of bus re-enumeration, the operating system will no longer be able to associate the correct interrupt with the re-enumerated bus number. Again, with Linux, this is indeed the case. Therefore, after re-enumerating a PCIe root port it is necessary to update the operating system’s PCI interrupt routing table in order to keep it in sync with the updated bus numbers.

Conclusion

Even with the limitations of the PC BIOS, it is possible to implement a system that utilizes PCIe Hot-swap on standard PC architectures. Implementing this same methodology on an embedded system, such as the MPC8548e, is also possible. In fact, it should be simpler due to the lack of a BIOS.

IDT has developed and can make available the source code for a Linux device driver that demonstrates the principles outlined in this document for use on platforms utilizing IDT PCIe switches.

References

Notes


Revision History

September 16, 2008: Initial publication.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard" : Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality" : Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations administered by the governments of any countries asserting jurisdiction over the parties or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters
TOYOSU FORESIA, 3-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information
For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks
Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2020 Renesas Electronics Corporation. All rights reserved.