PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	UEFI related updates
DATE:	7/29/2013
AFFECTED DOCUMENT:	PCI Firmware Specification Ver 3.1
SPONSOR:	Dong Wei, HP

Part I

1. 1. Summary of the Functional Changes

Update the references to the latest UEFI Specification. Make clarifications in 5.1.2 that the pointers to the Device List, the Configuration Utility Code header and the DMTF CLP Entry Point are not applicable to the UEFI Option ROMs.

2. 2. Benefits as a Result of the Changes

Although the Device List, the Configuration Utility Code header, the DMTF CLP Entry Point were designed for the PC Compatible Expansion ROM type, the definitions and some of the behavior descriptions were placed in the 5.1.2 section, which is meant to be a generic section applying to all Code Types. This has caused some confusions to the developers. This ECR is to make the clarifications explicitly.

3. 3. Assessment of the Impact

This is only a clarification of intent.

4. 4. Analysis of the Hardware Implications

None

5. 5. Analysis of the Software Implications

None

Part II

Detailed Description of the change

Modify Section 1.2 as follows:

Unified Extensible Firmware Interface Specification, Version 2.4 ~~2.3, Errata C~~, June~~July 14~~, 2013~~2010~~, http://www.uefi.org

Modify Section 3 as follows:

UEFI stands for Unified Extensible Firmware Interface. The *UEFI Specification, Version 2.4~~2.3~~* or later (http://www.uefi.org) describes an interface between the operating system and the platform firmware. The interface is in the form of data tables that contain platform-related information and boot and run-time services calls that are available to the operating system and its loader. Together, these provide a standard environment for booting an operating system.

The following sections provide an overview of the UEFI Services relevant to PCI (including Conventional PCI, PCI-X, and PCI Express). For details, refer to the UEFI Specification. UEFI is processor-agnostic.

Modify Section 3.5, page 35 as follows (remove the extra blank space):

…

Note: The operating system does not use the ~~-~~state of the Bus Master Enable bit to determine the validity of the BARs. If the BAR ranges are enabled, the device must respond to those addresses. The device may not be able to master a transaction, but enabled BARs shall be configured correctly by firmware.

…

Modify Section 3.5 page 37 as follows:

…

The operating system must not assume that all devices have been configured. Per Section 2.5.6 of UEFI (rev. 2.4~~2.3~~): the presence of an UEFI driver in the system firmware or in an option ROM does not guarantee that the UEFI driver will be loaded, executed, or allowed to manage any devices in a platform. In addition, UEFI drivers are not involved during PCI hot plug.

…

Modify Section 5.1.2 page 77 as follows:

Device List Pointer

The Device List Pointer is a two-byte pointer in little-endian format that points to the list of Device IDs supported by this ROM. The beginning reference point ("offset zero") for this pointer is the beginning of the PCI Data structure (the first byte of the Signature field). This field is only present in Revision 3.0 (and greater) PCI Data structures. A value of 0000h will be present in this field if the Expansion ROM does not support Device List. A value of 0000h must be present in this field if the Code Type field contains a value of 3 (UEFI).

Modify Section 5.1.2 page 78 as follows:

Code Type

The Code Type field is a one-byte field that identifies the type of code contained in this section of the ROM. The code may be executable binary for a specific processor and system architecture or interpretive code. The following code types are assigned:

Type	Description
0	Intel x86, PC-AT compatible
1	Open Firmware standard for PCI[1]
2	Hewlett-Packard PA RISC
3	Unified Extensible Firmware Interface (UEFI)
4-FF	Reserved

Modify Section 5.1.2 page 79 as follows:

Pointer to Configuration Utility Code Header

This pointer is a two-byte pointer in little-endian format that points to the Expansion ROM's Configuration Utility Code Header table at the beginning of the configuration code block described in Section 5.2.1.24. The beginning reference point ("offset zero") for this pointer is the beginning of the Expansion ROM image. This field is only present in Revision 3.0 (and greater) PCI Data structures. A value of 0000h will be present in this field if the Expansion ROM does not support a Configuration Utility Code Header. A value of 0000h must be present in this field if the Code Type field contains a value of 3 (UEFI).

Pointer to DMTF CLP Entry Point	This pointer is a two-byte pointer in little-endian format that points to the execution entry point for the DMTP CLP code supported by this ROM. The beginning reference point ("offset zero") for this pointer is the beginning of the Expansion ROM image. This field is only present in Revision 3.0 (and greater) PCI Data structures. A value of 0000h will be present in this field if the Expansion ROM does not support a DMTF CLP code entry point as described in Section 5.2.1.25. A value of 0000h must be present in this field if the Code Type field contains a value of 3 (UEFI).

Modify Section 5.1.3 as follows (There are also two extra spaces needing to be removed):

Revision 3.0 (or later) defines a Device List Pointer that -points to the list of Device IDs supported by the Expansion ROM image whose Code Type is not 3 (UEFI).

-The beginning reference point ("offset zero") for this pointer is the beginning of the PCI Data structure (the first byte of the Signature field). If this field does not exist (i.e., its contents are zero), then the Expansion ROM only supports the one specific Device ID listed in the Device ID field in the PCI Data structure. However, if this field is non-zero then it must point to a Device List Table. The format of this table is shown in Table 5-1.

Modify Section 5.2 Page 81 as follows:

6. If Vendor ID matches but the Device ID does not, the POST Firmware will examine the Device List Pointer. Assuming the Device List Pointer is not zero, the POST firmware will examine the Device List to find a match to the Device ID in the device. A value of 0000h indicates the end of the Device List.

   Note that Expansion ROMs of Code Type 3 (UEFI) do not support the Device List Pointer. For other Code Types, only Expansion ROMs compliant to this specification version 3.0 or later support the Device List Pointer. POST Firmware should not examine the Device List Pointer field until it has confirmed that the PCI Data Structure Revision Level is 3 or greater.

Modify Section 5.2.1.24 Page 94 as follows (remove the extra space):

Beginning with this version of the specification, the Expansion ROM vendors can isolate the configuration code and the system firmware will execute the configuration code only when specifically requested ~~-~~by the user. The Expansion ROM header (defined in Section 5.1.2) now contains an optional pointer to the header for the configuration code. During POST when the system firmware is preparing the Expansion ROM for the INIT stage, the system firmware will make a copy of the configuration utility code (if present in the ROM image) and leave it in temporary memory for later execution.

Modify Section 5.2.2 as follows:

5.2.2. UEFI Expansion ROM (Type 3)

Section 13.4.2~~12.4.1~~ "PCI Option ROMs" of the *Unified Extensible Firmware Interface Specification, Version 2.~~41.10~~* (http://www.uefi.org~~developer.intel.com/technology/efi~~) describes ways to store UEFI images (e.g., UEFI drivers) in PCI Expansion ROMs. The description in Section 5.2.1 of this document does not apply to UEFI Expansion ROMs.