**PCI-SIG ENGINEERING CHANGE REQUEST**

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th>Ignore PCI Boot Configuration _DSM function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATE:</strong></td>
<td>4/24/2006</td>
</tr>
<tr>
<td><strong>AFFECTED DOCUMENT:</strong></td>
<td>PCI Firmware Specification Ver 3.0</td>
</tr>
<tr>
<td><strong>SPONSOR:</strong></td>
<td>Tony Pierce, Microsoft</td>
</tr>
</tbody>
</table>

### Part I

1. **Summary of the Functional Changes**
   
   Add a function to the _DSM Definitions for PCI to provide an indication to an operating system that it can ignore the PCI boot configuration setup by the firmware during system initialization.

2. **Benefits as a Result of the Changes**

   Provides a mechanism for the platform to indicate to an operating system that boot configurations can be ignored for a device hierarchy.

   Allows firmware to thoroughly boot configure devices for best backward compatibility with current operating systems and provides a smooth transition to future operating systems.

3. **Assessment of the Impact**

   High for new operating systems that have the goal of providing flexible resource management.

4. **Analysis of the Hardware Implications**

   None.

5. **Analysis of the Software Implications**

   Low, _DSM is an optional control method.
Part II
Detailed Description of the change

Add Ignore PCI Boot Configuration function to the _DSM definitions for PCI as function 5 in Table 4-7 on page 56 as follows:

4.6 _DSM Definitions for PCI

_DSMP Device Specific Method is defined in the ACPI 3.0 Specification. This object is a control method that enables devices to provide device specific control functions that are consumed by the device driver. Table 5-1 below lists the UUID, revision, and function definitions.

<table>
<thead>
<tr>
<th>UUID</th>
<th>Revision</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E5C937D0-3553-4d7a-9117-EA4D19C3434D</td>
<td>1</td>
<td>1</td>
<td>PCI Express Slot Information</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>PCI Express Slot Number</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Vendor-specific Token ID</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>PCI Bus Capabilities</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>Ignore PCI Boot Configuration</td>
<td></td>
</tr>
</tbody>
</table>

Insert the following section after 4.6.4 and before 4.7

4.6.5. _DSMP for Ignoring PCI Boot Configurations

This section describes how system firmware can indicate to an operating system that it can ignore the boot configurations of PCI hierarchies and devices. The indication is exposed through the _DSMP ACPI method.

This _DSMP function is optional. If the platform does not provide it, the OS may continue to use the legacy handling regarding the boot configuration.

Note: Function 0 is a generic Query function that is supported by _DSMPs with any UUID and Revision ID. The definition of function 0 is generic to _DSMP and specified in the ACPI Specification, Version 3.0.

Location: This object can be placed under any virtual PCI-to-PCI bridge object representing the PCI Express root port or switch port, PCI Express endpoint, Conventional PCI/PCI-X bridge, or Conventional PCI/PCI-X device. When the object is placed on a bridge, the operating system will assume that the return value applies to the complete hierarchy produced by that bridge.

Note:
The resources for a PCI root bus are described using an _CRS method. If an operating system needs to change the resource assignment for a PCI root bus, it needs to examine the _PRS methods, select an appropriate resource list for the root bus and set them using an _SRS method.

**Arguments:**

Arg0: UUID: E5C937D0-3553-4d7a-9117-EA4D19C3434D

Arg1: Revision ID: 1

Arg2: Function Index: 5

Arg3: Empty Package
**Return:**
An integer whose description is as follows:

**Type:** Integer

**Purpose:** Ignore PCI Boot Configuration

**Description:**

0: No (The operating system shall not ignore the PCI configuration that firmware has done at boot time. However, the operating system is free to configure the devices in this hierarchy that have not been configured by the firmware. There may be a reduced level of hot plug capability support in this hierarchy due to resource constraints. This situation is the same as the legacy situation where this DSM is not provided.)

1: Yes (The operating system may ignore the PCI configuration that the firmware has done at boot time, and reconfigure/rebalance the resources in the hierarchy.)

**IMPLEMENTATION NOTE**

This DSM function provides backwards compatibility on platforms that can run legacy operating systems.

Operating systems for two different architectures (e.g., x86 and x64) can be installed on a platform. The firmware cannot distinguish the operating system in time to change the boot configuration of devices. Say for instance, an x86 operating system in non-PAE mode is installed on a system. The x86 operating system cannot access device resource space above 4GB. So the firmware is required to configure devices at boot time using addresses below 4GB. On the other hand, if an x64 operating system is installed on this system, it can access device resources above the 4GB so it does not want the firmware to constrain the resource assignment below 4GB that the firmware configures at boot time. It is not possible for the firmware to change this by the time it boots the operating system. Ignoring the configurations done by firmware at boot time will allow the operating system to push resource assignment using addresses above 4GB for an x64 operating system while constrain it to addresses below 4GB for an x86 operating system.