



## PCI-SIG ENGINEERING CHANGE NOTICE

<b>TITLE:</b>	CRS Software Visibility + RCRB Header
<b>DATE:</b>	January 21, 2004
<b>AFFECTED DOCUMENTS:</b>	PCI Express Base Specification, Rev 1.0a Root Complex Topology Discovery ECN
<b>SPONSOR:</b>	Carl Jackson; Hewlett-Packard Company Sridhar Muthrasanallur; Intel Corporation

### **Part I**

#### **1. Summary of the Functional Changes**

This ECN describes a standardized means for a Root Complex to handle Configuration Requests that return the Configuration Request Retry Status (CRS) Completion Status and provides a standardized means to make the CRS Completion Status visible to software.

Additionally, this ECN defines an optional Root Complex Register Block (RCRB) Header Capability containing Vendor ID & Device ID fields applicable to the RCRB plus Capabilities, Control, and Status registers for management of features such as CRS Software Visibility.

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

Some devices, Intelligent I/O sub-systems for example, can take a comparatively long time to initialize locally before presenting themselves to host enumeration software. This ECN provides a method for software to identify a device that is slow to initialize so that it may skip the device temporarily to work on other tasks rather than being stalled waiting for the device to become ready. This method can provide system latency improvements during boot-time, system wake-up and hot-plug events.

The RCRB Header Capability provides an RCRB with standardized vendor and device identification plus a feature management register set useable by CRS Software Visibility and potentially other Root Complex functionality.

#### **3. Assessment of the Impact**

This ECN adds optional functionality; thus, there is no impact to current software or hardware conforming to the PCI Express 1.0a specification.

#### **4. Analysis of the Hardware Implications**

Root Complexes (RCs) may optionally support the mechanism defined in this ECN to expose the PCI Express CRS Completion Status to software. Such Root Complexes must support the appropriate configuration register bits outlined in the ECN. This ECN also more tightly constrains when a device is permitted to return CRS Completion Status in response to a Configuration Request that follows a reset of the device.

## **5. Analysis of the Software Implications**

Software may optionally support the mechanism defined in this ECN to expose the PCI Express CRS Completion Status to software. Such software must comprehend the configuration register bits and the specific Configuration Retry read data value identified in this ECN. The CRS Software Visibility mechanism is turned off by default to maintain backwards compatibility with software that does not support this mechanism.

Existing software will not recognize the new RCRB Header Capability and thus will ignore it.

## Part II

### Detailed Description of the change

Section 2.3.1, page 80 – Modify the bullet beginning on line 7 as follows:

- For Configuration Requests only, following reset it is possible for a device to terminate the request but indicate that it is temporarily unable to process the Request, but will be able to process the Request in the future – in this case, the Configuration Request Retry Status (CRS) Completion Status is used (see Section 6.6). Valid reset conditions after which a device is permitted to return CRS are Cold, Warm and Hot Link Resets as well as reset initiated in response to a D3<sub>hot</sub> to D0<sub>uninitialized</sub> device state transition. A device is explicitly not permitted to return CRS following a software-initiated reset of the device, e.g., by the device’s software driver writing to a device-specific reset bit. Additionally, a device is not permitted to return CRS after having previously returned a Successful Completion without an intervening valid reset condition.

Section 2.3.1, p.82 – Modify the Implementation Note as follows:



## IMPLEMENTATION NOTE

### Configuration Request Retry Status

Some devices require a lengthy self-initialization sequence to complete before they are able to service Configuration Requests (common with intelligent I/O solutions on PCI). PCI/PCI-X architecture has specified a 2<sup>25</sup> (PCI) or 2<sup>26</sup> (PCI-X) clock “recovery time” ( $T_{rhfa}$ ) following reset to provide the required self-initialization time for such devices. PCI Express “softens” the need for this time based recovery period by implementing a Configuration Request Retry ~~Completion~~ Status (CRS) Completion Status. A device in receipt of a Configuration Request following a valid reset condition may respond with a ~~Configuration Request Retry~~ S Completion Status to terminate the Request, and thus effectively stall the Configuration Request until such time that the subsystem has completed local initialization and is ready to communicate with the host. Note that it is only legal to respond with a ~~configuration retry~~ CRS e Completion ~~s~~ Status in response to a Configuration Request. Sending this Completion Status in response to any other Request type can result in the generation of an error condition (Malformed TLP – see Section 6.2).

Receipt by the Requestor of a Completion with ~~Configuration Request Retry~~ S Completion Status terminates the Configuration Request on PCI Express. Further action by the Root Complex regarding the original Configuration Request is specified in Section 2.3.2, implementation specific, except during the period immediately following system reset (see Section 6.6). ~~A Root Complex may choose to re-issue the Configuration Request as a new Request on PCI Express or complete the Request to the host as a failed transaction. Root Complex implementations may further choose to only allow a fixed number of Configuration Request/Retry Completion Status loops before determining that something is wrong with the target of the Request and taking appropriate action.~~ Root Complexes that implement CRS Software Visibility have the ability to report the receipt of CRS Completion Status to software, enabling software to attend to other tasks rather than being

stalled while the device completes its self-initialization. Software that intends to take advantage of this mechanism must ensure that the first access made to a device following a valid reset condition is a Configuration Read Request accessing both bytes of the Vendor ID field in the device's configuration space header. For this case only, the Root Complex, if enabled, will synthesize a special read-data value for the Vendor ID field to indicate to software that CRS Completion Status has been returned by the device. For other Configuration Requests, or when CRS Software Visibility is not enabled, the Root Complex will generally re-issue the Configuration Request until it completes with a status other than CRS as described in Section 2.3.2.

When used in systems including PCI Express to PCI/PCI-X Bridges, system software and/or the Root Complex must comprehend the limit  $T_{\text{rthfa}}$  for PCI/PCI-X agents as described in Sections 2.8 and 6.6. Similarly, systems using PCI Express components which require additional self initialization time beyond the minimum guaranteed must provide some mechanism for re-issuing Configuration Requests terminated with CRS status. In systems running legacy PCI/PCI-X based software, the Root Complex must re-issue the Configuration Request using a hardware mechanism to ensure proper enumeration of the system.

See Section 6.6 for more information on reset.

---

*Section 2.3.2, p.89 – Modify the bullet beginning on line 17 as follows:*

- ❑ Root Complex handling of a Completion with Configuration Request Retry Status for a Configuration Request is implementation specific, except for the period following system reset (see Section 6.6). For Root Complexes that support CRS Software Visibility, the following rules apply:
  - If CRS Software Visibility is not enabled, the Root Complex must re-issue the Configuration Request as a new Request.
  - If CRS Software Visibility is enabled (see below):
    - ◆ For a Configuration Read Request that includes both bytes of the Vendor ID field of a device's Configuration Space Header, the Root Complex must complete the Request to the host by returning a read-data value of 0001h for the Vendor ID field and all '1's for any additional bytes included in the request. This read-data value has been reserved specifically for this use by the PCI-SIG and does not correspond to any assigned Vendor ID.
    - ◆ For a Configuration Write Request or for any other Configuration Read Request, the Root Complex must re-issue the Configuration Request as a new Request.

A Root Complex implementation may choose to limit the number of Configuration Request/CRS Completion Status loops before determining that something is wrong with the target of the Request and taking appropriate action, e.g., complete the Request to the host as a failed transaction.

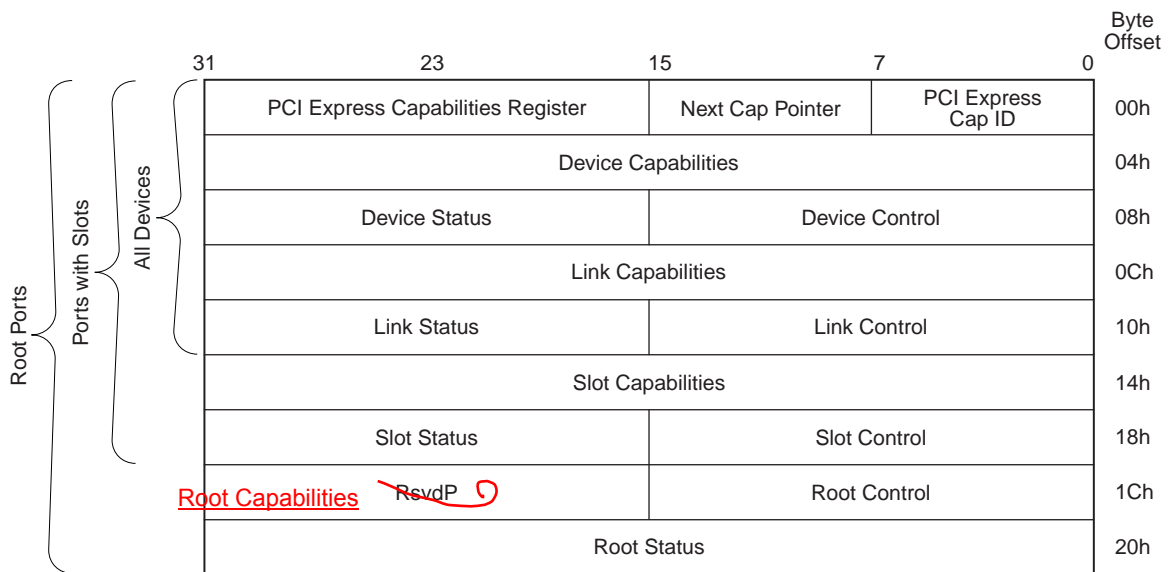
CRS Software Visibility may be enabled through the CRS Software Visibility Enable bit in the Root Control Register (see Section 7.8.12) to control Root Complex behavior on an individual Root Port basis. Alternatively, Root Complex behavior may be managed through the CRS Software Visibility Enable bit in an RCRB Header Capability as described in Section 7.y,

permitting the behavior of one or more Root Ports or Root Complex Integrated Endpoints to be controlled by a single Enable bit. For this alternate case, each Root Port or Integrated Endpoint declares its association with a particular Enable bit via an RCRB header association in a Root Complex Link Declaration Capability (see Section 7.13 of the Root Complex Topology Discovery ECN as updated by this ECN). Each Root Port or Integrated Endpoint is permitted to be controlled by at most one Enable bit. Thus, for example, it is prohibited for a Root Port whose Root Control Register contains an Enable bit to declare an RCRB header association to an RCRB that also includes an Enable bit in its RCRB Header Capability. The presence of an Enable bit in a Root Port or RCRB Header Capability is indicated by the corresponding CRS Software Visibility bit (see Sections 7.8.x and 7.y.3, respectively).

Section 2.8, p. 116 – Modify the last paragraph as follows:

~~Completion timeouts for Configuration Requests have special requirements for the support of PCI Express to PCI/PCI-X bridges (see Sections 2.3.4 and 6.6). Because of these special requirements, the support and timer values for a Completion Timeout for Configuration Requests are implementation specific. Furthermore, PCI Express to PCI/PCI-X Bridges, by default, are not enabled to return Configuration Request Retry Status (CRS) for Configuration Requests to a PCI/PCI-X device behind the Bridge. This may result in lengthy completion delays that must be comprehended by the Completion Timeout value in the Root Complex. System software may enable PCI Express to PCI/PCI-X Bridges to return Configuration Request Retry Status by setting the Bridge Configuration Retry Enable bit in the Device Control register, subject to the restrictions noted in the PCI Express to PCI/PCI-X Bridge Specification, Rev. 1.0.~~

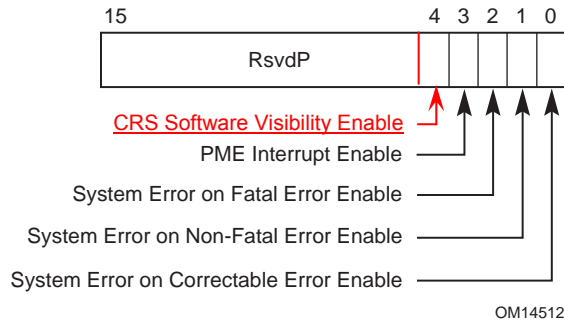
Section 7.8, p. 334 – Modify Figure 7-10 as shown:



OM14318

Figure 7-10: PCI Express Capability Structure

Section 7.8.12, p. 360 – Modify Figure 7-22 as shown:



**Figure 7-22: Root Control Register**

Section 7.8.12, p. 361 – Add a new entry to Table 7-20 as shown:

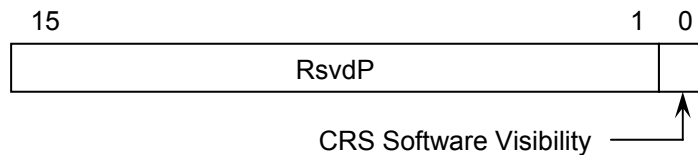
**Table 7-20: Root Control Register**

Bit Location	Register Description	Attributes
<u>4</u>	<u><b>CRS Software Visibility Enable</b> – This bit, when set, enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software (see Section 2.3.1). Default value of this field is 0. Root Ports that do not implement this capability must hardwire this bit to 0b.</u>	<u>RW</u>

Add the following new Section 7.8.x between existing Sections 7.8.12 & 7.8.13. Adjust affected Section, Figure, and Table numbering as appropriate:

**7.8.x. Root Capabilities Register (Offset 1Eh)**

The Root Capabilities register identifies PCI Express Root Port specific capabilities. Figure 7-a details allocation of register fields in the Root Capabilities register; Table 7-a provides the respective bit definitions.



**Figure 7-a: Root Capabilities Register**

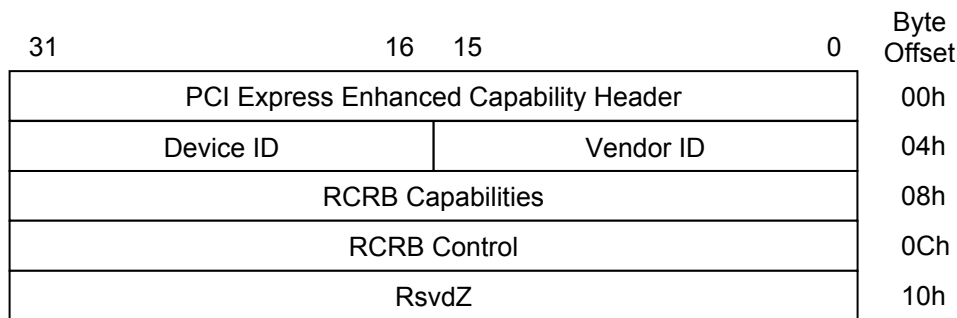
**Table 7-a: Root Capabilities Register**

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>0</u>	<b>CRS Software Visibility</b> – This bit, when set, indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software (see Section 2.3.1).	<u>RO</u>

Add the following new Section to the end of Chapter 7:

## **7.y. RCRB Header Capability**

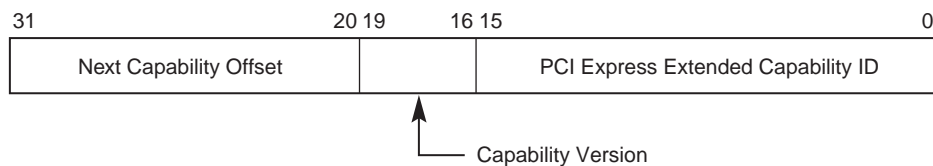
The PCI Express RCRB Header Capability is an optional extended capability that may be implemented in an RCRB to provide a Vendor ID and Device ID for the RCRB and to permit the management of parameters that affect the behavior of Root Complex functionality associated with the RCRB.



**Figure 7-b: Root Complex Features Capability Structure**

### **7.y.1. RCRB Header Enhanced Capability Header (Offset 00h)**

Figure 7-c details allocation of register fields in the RCRB Header Enhanced Capability header. Table 7-c provides the respective bit definitions. See Section 7.9.3 for a description of the PCI Express Enhanced Capabilities header. The Extended Capability ID for the RCRB Header Capability is 000Ah.



OM14533

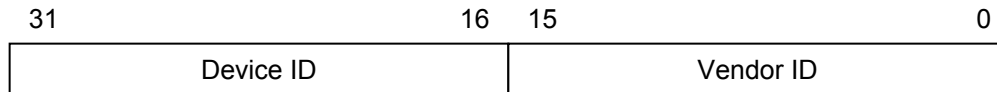
**Figure 7-c: RCRB Header Enhanced Capability Header**

**Table 7-c: RCRB Header Enhanced Capability Header**

<b><u>Bit Location</u></b>	<b><u>Register Description</u></b>	<b><u>Attributes</u></b>
<u>15:0</u>	<b><u>PCI Express Extended Capability ID</u></b> – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability.  <u>Extended Capability ID for the RCRB Header Capability is 000Ah.</u>	<u>RO</u>
<u>19:16</u>	<b><u>Capability Version</u></b> – This field is a PCI-SIG defined version number that indicates the version of the capability structure present.  <u>Must be 1h for this version of the specification.</u>	<u>RO</u>
<u>31:20</u>	<b><u>Next Capability Offset</u></b> – This field contains the offset to the next PCI Express capability structure or 000h if no other items exist in the linked list of capabilities.  <u>For Extended Capabilities implemented in device configuration space, this offset is relative to the beginning of PCI compatible configuration space and thus must always be either 000h (for terminating list of capabilities) or greater than 0FFh.</u>	<u>RO</u>

**7.y.2. Vendor ID (Offset 04h) and Device ID (Offset 06h)**

Figure 7-d details allocation of register fields in the RCRB Capabilities register; Table 7-d provides the respective bit definitions.



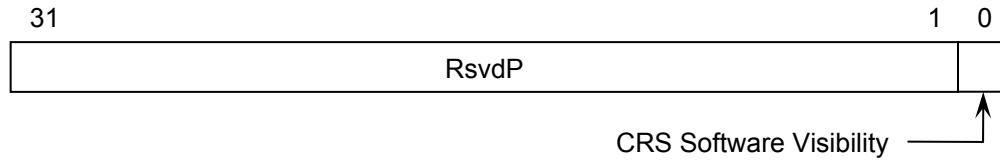
**Figure 7-d: Vendor ID and Device ID**

**Table 7-d: Vendor ID and Device ID**

<b><u>Bit Location</u></b>	<b><u>Register Description</u></b>	<b><u>Attributes</u></b>
<u>15:0</u>	<b><u>Vendor ID</u></b> – PCI-SIG assigned. Analogous to the equivalent field in PCI-compatible Configuration Space. This field provides a means to uniquely associate an RCRB with a particular vendor.	<u>RO</u>
<u>31:16</u>	<b><u>Device ID</u></b> – Vendor assigned. Analogous to the equivalent field in PCI-compatible Configuration Space. This field provides a means for a vendor to further classify a particular RCRB.	<u>RO</u>

### 7.y.3. RCRB Capabilities (Offset 08h)

Figure 7-e details allocation of register fields in the RCRB Capabilities register; Table 7-e provides the respective bit definitions.



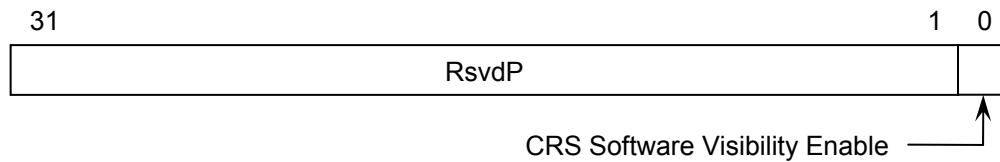
**Figure 7-e: RCRB Capabilities**

**Table 7-e: RCRB Capabilities**

<b>Bit Location</b>	<b>Register Description</b>	<b>Attributes</b>
<b>0</b>	<b>CRS Software Visibility</b> – This bit, when set, indicates that the Root Complex is capable of returning Configuration Request Retry Status (CRS) Completion Status to software for all Root Ports and integrated devices associated with this RCRB (see Section 2.3.1).	<b>RO</b>

### 7.y.4. RCRB Control (Offset 0Ch)

Figure 7-f details allocation of register fields in the RCRB Control register; Table 7-f provides the respective bit definitions.



**Figure 7-f: RCRB Control**

**Table 7-f: RCRB Control**

<b>Bit Location</b>	<b>Register Description</b>	<b>Attributes</b>
<b>0</b>	<b>CRS Software Visibility Enable</b> – This bit, when set, enables the Root Complex to return Configuration Request Retry Status (CRS) Completion Status to software for all Root Ports and integrated devices associated with this RCRB (see Section 2.3.1). Default value of this field is 0. RCRBs that do not implement this capability must hardwire this bit to 0b.	<b>RW</b>

In Section 7.13 on page 5 of the RC Topology Discovery ECN, append to the paragraph before the Implementation Note as follows:

The PCI Express Root Complex Link Declaration Capability may be present in a Root Complex element’s configuration space or RCRB. It declares links from the respective element to other elements of the same Root Complex Component or to an element in another Root Complex Component. The links are required to be declared bidirectional such that each valid data path from one element to another has corresponding link entries in the configuration space (or RCRB) of both elements.

The PCI Express Root Complex Link Declaration Capability may also declare an association between a configuration space element (Root Port or Root Complex Integrated Endpoint) and an RCRB Header Capability (see Section 7.y) contained in an RCRB that affects the behavior of the configuration space element. Note that an RCRB Header association is not declared bidirectional; the association is only declared by the configuration space element and not by the target RCRB.

In Section 7.13.3.1 on page 9 of the RC Topology Discovery ECN, modify Figure 7-60 and Table 7-52 as follows:

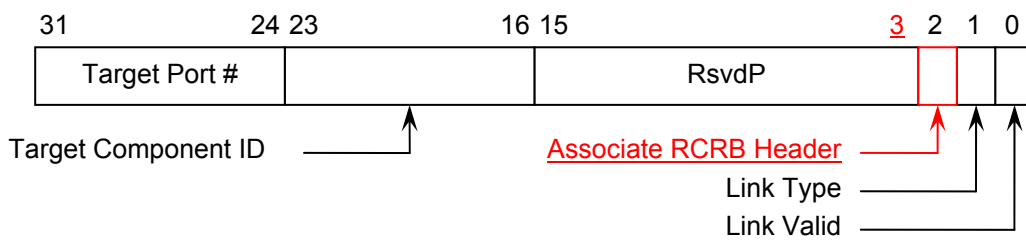


Figure 7-60: Link Description Register

Table 7-52: Link Description Register

Bit Location	Register Description	Attributes
0	<b>Link Valid</b> – This field when set to 1 indicates that the Link Entry specifies a valid link. Link entries that do not have <u>either</u> this bit set <u>or the Associate RCRB Header bit set (or both)</u> are ignored by software.	HwInit
1	<b>Link Type</b> – This field indicates the target type of the link and defines the format of the link address field. Defined encodings are: 0 – Link points to memory-mapped space (for RCRB). The link address specifies the 64-bit base address of the target RCRB. 1 – Link points to configuration space (for a Root Port or Root Complex Integrated <b>DeviceEndpoint</b> ). The link address specifies the configuration address (segment, bus, device, function) of the target element.	HwInit
<u>2</u>	<u><b>Associate RCRB Header</b>– This field when set to 1 indicates that the Link Entry associates the declaring element with an RCRB Header Capability in the target RCRB. Link entries that do not have either this bit set or the Link Valid bit set (or both)</u>	<u>HwInit</u>

	<p><u>are ignored by software.</u></p> <p><u>The Link Type bit must be 0b when this bit is set.</u></p>	
23:16	<p><b>Target Component ID</b> – This field identifies the Root Complex Component that is targeted by this link entry. A value of 0 is reserved. Component IDs start at 1.</p>	HwInit
31:24	<p><b>Target Port Number</b> – This field specifies the port number associated with the element targeted by the link entry; the target port number is with respect to the Root Complex Component (identified by the Target Component ID) that contains the target element.</p>	HwInit

<End of ECN>